ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Önder ÇOBAN

ATTRIBUTE INFERENCE OVER REAL-WORLD ONLINE SOCIAL NETWORKS: A COMPREHENSIVE PRIVACY ANALYSIS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2021

ABSTRACT

PhD THESIS

ATTRIBUTE INFERENCE OVER REAL-WORLD ONLINE SOCIAL NETWORKS: A COMPREHENSIVE PRIVACY ANALYSIS

Önder ÇOBAN

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Prof. Dr. Selma Ayşe ÖZEL
II. Supervisor : Asst. Prof. Dr. Ali İNAN
Year: 2021, Pages: 295
Jury : Prof. Dr. Selma Ayşe ÖZEL
: Assoc. Prof. Dr. Mustafa ORAL
: Asst. Prof. Dr. Mümine KAYA KELEŞ
: Prof. Dr. Yücel SAYGIN
: Assoc. Prof. Dr. Osman ABUL

Today, people can make interactions through online social networks (OSNs) that allow users to reveal and see their personal information, make connections with other users, and view the public information of other users through the connections. As such, OSNs are extremely popular and inevitable parts of people's daily lives today. As a result, OSNs have become attractive data sources for researchers from many disciplines including computer science to investigate privacy, security, and user behavior issues. However, studies that aim to make privacy analysis of Turkish OSN users are quite limited. In this thesis, a comprehensive privacy analysis of Turkish Facebook users was carried out due to the popularity of Facebook OSN. Different methods were used to infer users' private attributes including gender, kinship, and so on. Then the inferred attributes were used in the privacy risk analysis to show that users are at higher risk than they believe. Additionally, two natural language processing (NLP) tasks including sentiment analysis and named entity recognition have been performed as they can be utilized in privacy risk analysis. NLP and feature inference tasks were handled as sub-problems. In these tasks, various contributions were made to the literature, and some of them provided state-of-the-art results.

Keywords: Attribute inference, privacy, security, online social networks, Facebook.

ÖZ

DOKTORA TEZİ

GERÇEK ÇEVRİMİÇİ SOSYAL AĞLAR ÜZERİNDE NİTELİK ÇIKARSAMA: KAPSAMLI BİR GİZLİLİK ANALİZİ

Önder ÇOBAN

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Prof. Dr. Selma Ayşe ÖZEL
II. Danışman : Dr. Öğr. Üyesi Ali İNAN
Yıl: 2021, Sayfa: 295
Jüri : Prof. Dr. Selma Ayşe ÖZEL
: Doç. Dr. Mustafa ORAL
: Dr. Öğr. Üyesi Mümine KAYA KELEŞ
: Prof. Dr. Yücel SAYGIN
: Doç. Dr. Osman ABUL

Günümüzde insanlar, kullanıcılarına kişisel bilgilerini paylaşma ve görüntüleme, diğer kullanıcılar ile bağlantı kurma ve bağlantılarını kullanarak diğer kullanıcıların paylaştığı herkese açık bilgileri görüntüleme imkânı sunan çevrimiçi sosyal ağlar (ÇSA) vasıtasıyla etkileşim kurabilmektedir. Bu nedenle, ÇSA'lar oldukça popülerdir ve günümüzde insanların günlük hayatının vazgeçilmez bir parçasıdır. Sonuç olarak, ÇSA'lar bilgisayar bilimini de içeren birçok alandan araştırmacılar için mahremiyet, güvenlik ve kullanıcı davranışı gibi konuları incelemek için çekici bir veri kaynağı olmuştur. Ancak, Türkiye'deki ÇSA kullanıcılarının gizlilik analizini amaçlayan çalışmalar oldukça sınırlıdır. Bu tezde, popüler olmasından dolayı Türkiye'deki Facebook kullanıcıları için kapsamlı bir gizlilik analizi gerçekleştirilmiştir. Kullanıcıların cinsiyet, akrabalık vb. gibi gizli bilgilerini çıkarsamak için farklı yöntemler kullanılmıştır. Ortaya çıkarılan bilgiler daha sonra kullanıcıların inandıklarından daha yüksek risk taşıdığını göstermek için gizlilik riski analizinde kullanılmıştır. Ek olarak, gizlilik riski analizinde kullanılabileceği için duygu analizi ve adlandırılmış varlık ismi tanıma olmak üzere iki doğal dil işleme (DDİ) çalışması gerçekleştirilmiştir. DDİ ve nitelik çıkarsama görevleri alt problemler olarak ele alınmıştır. Bu görevlerde literatüre çeşitli katkılar sağlanmış ve bazılarında en güncel sonuçlar elde edilmiştir.

Anahtar kelimeler: Nitelik çıkarsama, mahremiyet, güvenlik, çevrimiçi sosyal ağlar, Facebook.

EXTENDED ABSTRACT

Online Social Networks (OSNs) are extremely popular services that allow users to connect, express themselves, and share information such as activities, photographs, favorites, posts, and so on (Viswanath et al., 2009; Abdesslem et al., 2012). Popular OSNs like Facebook, Instagram, and Twitter serve hundreds of millions of users which are connected with over a billion links. Some OSNs are even said to reflect real-life (Wani et al., 2018). For instance, over the first quarter of 2020, Facebook had over 2.6 billion monthly active users, which is indicating an impressive 10% yearly increase (Anonymous., 2020a).

As a result of this popularity and content-rich activity of OSN users, the data collected through OSNs have attracted extensive attention from commercial, governmental, and academic organizations. Researchers of various fields ranging from sociology to computer science have shown interest in OSN data (Xiao et al., 2012; Wani et al., 2018). Researchers from computer science as being popular one among these fields, aim at extracting useful information (Kridalukmana., 2015; Siwag et al., 2014), analyzing user behavior and social activity (Terrana et al., 2014; Terrana et al., 2015), finding similar users (Fire et al., 2014), identifying fake profiles (Gupta and Kaushal., 2017), analyzing sentiment and opinion trends (Ahkter and Soria., 2010), and analyzing content-sharing behavior (Mfenyana et al., 2014) among many others.

Unfortunately, OSN data is not always used for such good purposes as research. Consider the recent Cambridge Analytica data scandal (Cadwalladr and Graham-Harrison., 2018) that Facebook had to face. According to Facebook's own statement, the profile information of more than 87 million users may have been used inappropriately. This has not been and will not be the only case as publicly sharing private information on their profiles make OSN users an easier target for cyber-criminals (Illmer, 2016). For instance, the Guardian has covered the news that trillions of tweets would be put up for sale (Garside, 2015). In a much recent

case, a collection of popular Facebook quiz apps exposed the private data of up to 120 million users (Nick, 2018).

All of these cases show that alongside the sunny bright world where OSN users vividly socialize with their friends digitally, there is a much darker facet of OSN services where what we share might do us harm. Fortunately, there also has been considerable research effect put into understanding the potential extent of this harm that may cause privacy threats caused by misuse of OSNs such as identity disclosure, attribute disclosure, link disclosure, and affiliation disclosure (Zheleva and Getoor., 2011). In addition, there are also studies on OSN security that try to either expose the inadequacies of existing OSN frameworks or suggest new ones (Gross and Acquisti., 2005; Cutillo et al., 2009; Krishnamurthy et al., 2009).

In literature, many other studies provide significant insight into the privacy and security implications of OSN usage, user behavior analysis, sentiment analysis, opinion trend analysis, and so on. However, the number of studies with a focus on privacy analysis and attribute inference for Turkish OSN users, especially for Facebook, is very limited. Motivated by this shortcoming, in this thesis, we perform a comprehensive analysis which mainly focuses on private attribute inference and privacy analysis of Turkish OSN users.

We selected Facebook as our target OSN due to its popularity and perform our analysis of real-world data. To perform our comprehensive privacy analysis, we performed inference tasks to infer user's different attributes including gender, kinship, other OSN account (via user matching), and so on. Additionally, we performed NLP tasks including sentiment analysis and named entity recognition since they can be employed in the privacy risk analysis process. We handled each of the inference and NLP tasks as a separate problem. We would like to note that most of these sub-tasks have not been performed before for Turkish OSN users and we obtained the state-of-the-art results by providing novel aspects for some of these tasks.

The first step required for OSN research is to access OSN data. However, despite the abundance of studies on OSN research, publicly available OSN data is very scarce. Therefore, we first focused on the problem of efficient collection of real, large-scale, content-rich, and accurate Facebook data. We presented an algorithmic data collection methodology that is applicable to any OSN (Coban et al., 2020a). To summarize key differences, our methodology supports multi-threaded data collection, is resilient against access restrictions imposed by Facebook, collects each and every bit of publicly shared data, and respects BFS order from the seed account. Upon completion of crawling Facebook data, we analyzed the crawled data to summarize the behavior of Turkish Facebook users. To the best of our knowledge, this is the first effort that only regards the users from Turkey and provides their behavioral analysis.

Next, we performed user attribute disclosure (or inference) to disclose some private information of users that can be utilized to harm the privacy of users. Our attribute inference tasks focus on detecting the private gender and kinship attributes of users. In the gender detection task, we used different models that use various information obtained from users' profiles. We obtained an accuracy of **0.982** that is the state-of-the-art performance for Turkish users and outperforms the majority of the results of existing gender detection studies performed over OSNs.

In the kinship detection task, on the other hand, we performed fine-grained kinship detection of Facebook users based on their wall contents. We believe that this is the first research for fine-grained kinship detection with respect to both information considered and language in OSNs. We obtained a promising best F1 score of **0.41** that shows that content-based kinship detection can be a good starting point for future studies even it has some challenges.

Moreover, we performed user identification (i.e., user matching) to infer other OSN accounts of Facebook users. This task aimed to match different accounts of the same user across different OSNs. However, it can be considered as an attribute inference task and useful for other OSN related problems such as link

prediction, friendship recommendation, and so on. Therefore, we performed a user identification task where we build a learning model that is based on the usernames of users which are fundamental elements of all websites and can reflect users' characteristics and habits. The primary focus in this task was to build a learning model to say that two account matches if they belong to the same user. We performed this task just based on usernames as our crawled Facebook data is incomplete. We conducted our experiments on our real-world username dataset and obtained the best F1 score of **0.921**. To the best of our knowledge, this task is its first kind as it considers Turkish Facebook users.

One of the most interesting properties of OSNs is that communication takes place via text messages. Extracted information from users' textual contents can be used for many different purposes. This information can also be employed in the privacy risk analysis of users. For instance, sentiment analysis can be used in measuring social tie strengths between two OSN users. Extracted NEs, on the other hand, can be used to detect whether two OSN users graduated from the same school. Deriving many other examples is also possible. Therefore, in this thesis, we additionally performed SA and NER tasks where we investigate the sentimental orientation of Turkish Facebook users and performed named entity recognition over Facebook textual data respectively. Notice that OSN data often includes very dirty texts and compared to NLP studies for other languages, studies focusing on Turkish OSN data are scarce, and the majority of existing studies focus on Twitter OSN. As such, we contributed to the literature by performing these natural language-dependent tasks in the context of Facebook OSN and Turkish users.

For the NER task, we developed a CRF-based NER system and employed it to discover NEs from Facebook posts. In order to do, we created a dataset, namely, FBNER, which contains Turkish posts that automatically crawled from Facebook OSN. To the best of our knowledge, this is the first study to extract NEs from Turkish Facebook posts. Our CRF-based NER system achieves an F1 score of **0.713** when training and test sets are from the same domain.

For the SA task, on the other hand, a manually labeled dataset that is including Turkish texts collected from public Facebook activities is created. This task differs from existing studies in terms of the dataset scale, the natural language of texts in the dataset and the extend of experimental analyses which include both ML and DL techniques. We extensively reported not only the results of different learning models involving SA, but also the statistical distribution of meta-data of user activities across various attributes such as gender and age. Our experimental results indicated that RNNs achieve the best accuracy of **0.916** with word embeddings. To the best of our knowledge, this is the best result for SA on Facebook data in the context of the Turkish language.

Any information shared on OSNs may be sensitive and any kind of disclosure can significantly harm user's privacy (Srivastava and Geethakumari, 2013) by giving way to some threats such as identity theft, digital stalking, building consumer models for advertising, identity cloning, and blackmailing among many others. A measure of how much sensitive information is shared with others on an OSN would help the users to understand whether they individually share too much (Sramka, 2015). Therefore, we performed a preliminary privacy risk scoring of Turkish Facebook users by using state-of-the-art techniques in this thesis.

Finally, we performed a privacy risk analysis of Turkish Facebook users by employing attribute inference that depends on users' textual contents and social connections. We would like to note that in the attribute inference tasks described above (i.e., gender, kinship, and user matching), we used machine learning models. However, our attribute inference task in this final task focused on inferring values of many other additional attributes including e-mail, date of birth, political view, and so on. We used inferred values of attributes in our privacy analysis task that is the main focus of this thesis.

Our comprehensive analysis which includes sub-tasks mentioned above differs from existing work in the broadness of the features considered, machine learning methods applied, and the size of the OSN dataset used in the experimental

evaluation. Although we have performed different sub-tasks, our results, in general, show that OSN data is very dirty and often incomplete. The majority of Turkish users are aware of privacy risks, but they still unaware that their private information can be inferred even they keep related information private. Performing OSN related studies often provide high accuracy due to the structure of OSNs in their nature which includes social connections among users. Content-based tasks, on the other hand, often challenging and requires language-dependent effective steps to achieve high accuracy.

GENİŞLETİLMİŞ ÖZET

Çevrimiçi Sosyal Ağlar (ÇSA) kullanıcılara birbirlerine bağlanmayı, kendilerini ifade etmeyi ve aktivite, fotoğraf, favori, durum güncellemeleri ve benzeri bilgileri paylaşmalarını mümkün kılan oldukça popüler olmuş servislerdir (Viswanath et al., 2009; Abdesslem et al., 2012). Facebook, Instagram ve Twitter gibi popüler ÇSA'lar bir milyardan fazla bağlantı ile bağlanmış yüz milyonlarca kullanıcıya hizmet vermektedir. Hatta bazı ÇSA'ların gerçek hayatı yansıttığı bile söylenmektedir (Wani et al., 2018). Örneğin, Facebook 2020 yılının ilk çeyreğinde 2,6 milyar aylık aktif kullanıcı sayısına erişmiştir ve bu değer yıllık %10 oranında etkileyici bir artış göstermiştir (Anonymous., 2020a).

Bu popülaritenin ve ÇSA kullanıcılarının içerik bakımından zengin aktivitelerinin bir sonucu olarak, ÇSA'lardan toplanan veri ticari, kamusal ve akademik organizasyonların yoğun ilgisini çekmiştir. Sosyolojiden bilgisayar bilimine kadar çeşitli alanlardan araştırmacılar ÇSA verisine ilgi göstermiştir (Xiao et al., 2012; Wani et al., 2018). Bu alanlar arasında popüler bir alan olan bilgisayar bilimi araştırmacıları; faydalı bilgi çıkarımı (Kridalukmana., 2015; Siwag et al., 2014), kullanıcı davranış ve sosyal aktivite analizi (Terrana et al., 2014; Terrana et al., 2015), benzer kullanıcıları bulma (Fire et al., 2014), sahte hesapların tespiti (Gupta and Kaushal., 2017), duygusal yönelim ve fikir eğilimi (Ahkter and Soria., 2010) ve içerik paylaşım davranışı analizi (Mfenyana et al., 2014) gibi görevleri amaçlamaktadır.

ÇSA verisi ne yazık ki her zaman böyle iyi amaçlar için kullanılmamaktadır. Örneğin, Facebook yakın zamanda gerçekleşen Cambridge Anlytica veri skandalı (Cadwalladr and Graham-Harrison., 2018) ile yüzleşmek zorunda kalmıştır. Facebook 87 milyondan fazla kullanıcının profil bilgisinin uygun olmayan şekilde kullanılmış olabileceğini ifade etmiştir. Gizli bilgilerin kullanıcılar tarafından ÇSA hesaplarında paylaşılması onları siber suçlular için kolay bir hedef hâline getirdiğinden (Illmer., 2016) bu karşılaşılan ilk olay değildir

ve olmayacaktır. Örneğin, Guardian trilyonlarca tweet'in satılmış olabileceğini ortaya çıkarmıştır (Garside., 2015). Çok daha yakın bir olayda, popüler Facebook quiz uygulamalarından bazıları 120 milyon kullanıcının gizli bilgilerini ortaya çıkarmıştır (Nick., 2018).

Tüm bu vakalar, ÇSA kullanıcılarının arkadaşlarıyla dijital olarak canlı bir şekilde sosyalleştiği güneşli parlak dünyanın yanı sıra, ÇSA servislerinin çok daha karanlık bir yönü olduğunu ve paylaştıklarımızın bize zarar verebileceğini göstermektedir. Neyse ki, kimlik ifşası, öznitelik ifşası, bağlantı ifşası ve üyelik ifşası gibi ÇSA'ların kötüye kullanımından kaynaklanan gizlilik tehditlerine neden olabilecek bu zararın potansiyel kapsamını anlamak için önemli araştırma çabaları da bulunmaktadır (Zheleva and Getoor., 2011). Ek olarak, ÇSA güvenliği konusunda da mevcut ÇSA çerçevelerinin yetersizliklerini deneyen ya da yenilerini öneren çalışmalar bulunmaktadır (Gross and Acquisti., 2005; Cutillo et al., 2009; Krishnamurthy et al., 2009).

Literatürde, ÇSA kullanımının gizlilik ve güvenlik etkileri, kullanıcı davranış analizi, ÇSA kullanıcılarının duygu ve fikir eğilim analizi ve benzeri konular üzerinde önemli bilgiler sunan diğer birçok çalışma bulunmaktadır. Ancak özellikle Facebook olmak üzere Türk ÇSA kullanıcıları için gizlilik analizi ve nitelik çıkarsama odaklı çalışmaların sayısı oldukça sınırlıdır. Bu eksiklikten yola çıkarak bu tezde, Türk ÇSA kullanıcıları için nitelik çıkarsama ve gizlilik analizi odaklı kapsamlı bir analiz gerçekleştirilmiştir.

Hedef ÇSA olarak popülerliği nedeniyle Facebook seçilmiş ve analizler gerçek veri üzerinde gerçekleştirilmiştir. Kapsamlı gizlilik analizi gerçekleştirmek için kullanıcıların cinsiyet, akrabalık vb. gibi farklı bilgilerini çıkarsama amaçlı görevler gerçekleştirilmiştir. Ek olarak, gizlilik riski analizinde kullanılabileceği için duygu analizi ve varlık ismi tanıma gibi doğal dil işleme (DDİ) görevleri gerçekleştirilmiştir. Nitelik çıkarsama ve DDİ görevlerinin her biri ayrı bir problem olarak ele alınmıştır. Bu alt görevlerin çoğunluğu Türk ÇSA kullanıcıları

bağlamında daha önce ele alınmamıştır ve bu problemlerin bazıları için yeni bakış açıları sunularak en güncel sonuçlar elde edilmiştir.

ÇSA odaklı bir araştırma için gereken ilk adım, ÇSA verisine erişmektir. Ancak, ÇSA odaklı çok sayıda çalışmaya rağmen, herkesin erişimine açık ÇSA verisi çok azdır. Bu nedenle, bu tez öncelikle gerçek, büyük ölçekli, zengin içerikli ve doğru Facebook verisinin verimli bir şekilde toplanmasına odaklanmıştır. Bu amaçla, herhangi bir ÇSA'ya uygulanabilen bir algoritmik veri toplama metodolojisi sunulmuştur (Coban et al., 2020a). Temel farklılıkları özetlemek gerekirse; metodolojimiz çok iş parçacıklı veri toplamayı desteklemektedir, Facebook tarafından uygulanan erişim kısıtlamalarına karşı dayanıklıdır, herkese açık olarak paylaşılan verilerin her bir parçasını toplamaktadır ve tohum hesaptan başlayarak BFS sırasına göre ağı dolaşmaktadır. Facebook verilerinin taranması tamamlandıktan sonra, Türk Facebook kullanıcılarının davranışlarını özetlemek için taranan veriler analiz edilmiştir. Bildiğimiz kadarıyla, bu sadece Türkiye'den kullanıcıları ilgilendiren ve davranış analizlerini sağlayan ilk çalışmadır.

Daha sonra, kullanıcıların mahremiyetine zarar vermek için kullanılabilecek bazı özel bilgileri ifşa etmek için kullanıcı nitelik ifşası (veya çıkarsama) gerçekleştirilmiştir. Nitelik çıkarsama görevleri, kullanıcıların gizli cinsiyet ve akrabalık özniteliklerini tespit etmeye odaklanmaktadır. Cinsiyet tespiti görevinde, kullanıcıların profillerinden elde edilen çeşitli bilgileri kullanan farklı modeller kullanılmıştır. ÇSA'lar üzerinde yapılan mevcut cinsiyet tespiti çalışmalarında elde edilen sonuçların çoğundan daha iyi olan ve Türk kullanıcılar için en güncel performans olan **0,982** doğruluk elde edilmiştir.

Akrabalık tespiti görevinde ise Facebook kullanıcılarının duvar içeriklerine göre ince taneli akrabalık tespiti yapılmıştır. Bu görev hem dikkate alınan bilgi hem de dil açısından ÇSA'lar üzerinde ince taneli akrabalık tespiti amaçlı ilk araştırmadır. İçeriğe dayalı akrabalık tespitinin bazı zorlukları olsa da gelecekteki çalışmalar için iyi bir başlangıç noktası olabileceğini gösteren **0,41** F1 skoru elde edilmiştir.

Ayrıca, Facebook kullanıcılarının diğer ÇSA hesaplarını çıkarsamak için kullanıcı tespiti (yani kullanıcı eşleştirme) gerçekleştirilmiştir. Bu görev aslında aynı kullanıcının farklı ÇSA'lardaki hesaplarını eşleştirmeyi amaçlamaktadır. Bununla birlikte, bir nitelik çıkarsama görevi olarak düşünülebilir ve bağlantı tahmini, arkadaşlık önerisi ve benzeri gibi ÇSA'lar ile ilgili diğer sorunlar için faydalı olabilir. Bu nedenle, bu görevde tüm web sitelerinin temel unsurları olan ve kullanıcıların özelliklerini ve alışkanlıklarını yansıtabilen kullanıcı adlarına dayalı bir öğrenme modeli oluşturulmuştur. Bu görevdeki temel hedef, aynı kullanıcıya ait iki hesabın eşleşeceğini söyleyecek bir öğrenme modeli oluşturmaktır. Taranan Facebook verisi tüm bilgileri içermediğinden, bu görev yalnızca kullanıcı adlarını kullanmaktadır. Deneyler gerçek dünya kullanıcı adı veri seti üzerinde gerçekleştirmiş ve en iyi **0,921** F1 skoru elde edilmiştir. Bildiğimiz kadarıyla bu görev, Türk Facebook kullanıcılarını farklı ÇSA hesaplarını eşleştirmeyi amaçlayan ilk çalışmadır.

ÇSA'ların en ilginç özelliklerinden birisi, iletişimin metin mesajları yoluyla gerçekleşmesidir. Kullanıcıların metin içeriklerinden çıkarılan bilgiler birçok farklı amaç için kullanılabilir. Bu bilgiler, kullanıcıların gizlilik riski analizinde de kullanılabilir. Örneğin, duygu analizi, iki ÇSA kullanıcısı arasındaki sosyal bağ kuvvetini ölçmek için kullanılabilir. Çıkarılan varlık isimleri ise iki ÇSA kullanıcısının aynı okuldan mezun olup olmadığını tespit etmek için kullanılabilir. Diğer birçok örneği türetmek de mümkündür. Bu nedenle, bu tezde ayrıca, Türk Facebook kullanıcılarının duygusal yönelimlerini araştırmak ve varlık ismi tanıma amacıyla Facebook metin verileri üzerinde sırasıyla duygu analizi (DA) ve adlandırılmış varlık ismi tanıma (VİT) görevleri gerçekleştirilmiştir. ÇSA verilerinin genellikle çok kirli metinler içermektedir ve diğer diller ile karşılaştırıldığında, Türkçe ÇSA verilerine odaklanan DDİ çalışmaları azdır ve mevcut çalışmaların çoğu Twitter üzerinde odaklanmıştır. Bu nedenle, doğal dile bağımlı bu görevler Facebook ve Türk kullanıcıları bağlamında gerçekleştirilerek literatüre katkıda bulunulmuştur.

VİT görevi için, KRA (Koşullu Rassal Alanlar) tabanlı bir sistem geliştirilmiş ve Facebook gönderilerinden varlık isimlerini tespit etmek için kullanılmıştır. Bu amaçla Facebook'tan otomatik olarak toplanan Türkçe gönderileri içeren FBNER adlı bir veri kümesi oluşturulmuştur. Bildiğimiz kadarıyla, bu görev Türk Facebook paylaşımlarından varlık isimlerini tespit etmek amacıyla yapılan ilk çalışmadır. KRA tabanlı VİT sistemimiz, eğitim ve test setleri aynı alandan olduğunda **0,713** F1 skoru elde etmiştir.

DA görevi için ise, herkese açık Facebook etkinliklerinden toplanan Türkçe metinleri içeren manuel olarak etiketlenmiş bir veri kümesi oluşturulmuştur. Bu görev, veri seti ölçeği, veri setindeki metinlerin doğal dili ve makine öğrenmesi ve derin öğrenme tekniklerini içeren deneysel analizlerin kapsamı bakımından mevcut çalışmalardan farklıdır. Sadece DA'yı içeren farklı öğrenme modellerinin kapsamlı sonuçları değil, kullanıcı etkinliklerinin meta verilerinin cinsiyet ve yaş gibi çeşitli nitelikler bakımından istatistiksel dağılımı da kapsamlı bir şekilde raporlanmıştır. Deneysel sonuçlarımız, TSA (Tekrarlayan Yapay Sinir Ağları)'nın kelime gömmeleri ile **0.916** gibi en iyi doğruluğa ulaştığını göstermektedir. Bildiğimiz kadarıyla, bu, Türkçe bağlamında Facebook verileri üzerinde gerçekleştirilmiş DA için en güncel sonuçtur.

ÇSA'larda paylaşılan herhangi bir bilgi hassas olabilir ve her türlü ifşa kimlik hırsızlığı, dijital takip, reklam için tüketici modelleri oluşturma, kimlik klonlama ve şantaj gibi bazı tehditlere yol açarak kullanıcının gizliliğine önemli ölçüde zarar verebilir (Srivastava ve Geethakumari, 2013). Bir ÇSA'da başkalarıyla paylaşılan bilgilerin ne kadarının hassas olduğunun ölçüsü, kullanıcıların bireysel olarak çok şey paylaşıp paylaşmadıklarını anlamalarına yardımcı olacaktır (Sramka, 2015). Bu nedenle, bu tezde güncel teknikler kullanarak Türk Facebook kullanıcılarının gizlilik risk puanlaması için bir ön analiz yapılmıştır.

Son olarak, kullanıcıların metin içeriklerine ve sosyal bağlantılarına dayanan nitelik çıkarsamayı kullanarak Türk Facebook kullanıcılarının gizlilik riski

analizi gerçekleştirilmiştir. Yukarıda açıklanan nitelik çıkarsama görevlerinde (yani cinsiyet, akrabalık ve kullanıcı eşleştirme) makine öğrenimi modelleri kullanılmıştır. Bununla birlikte, bu son ve kapsamlı analizdeki nitelik çıkarsama görevi e-posta, doğum tarihi, politik görüş vb. gibi diğer birçok niteliğin değerlerinin de öğrenilmesine odaklanmaktadır. Bu niteliklerin değerleri bu tezin ana odak noktası olan kapsamlı gizlilik analizi görevinde kullanılmıştır.

Bütün bu alt problemleri barındıran kapsamlı analizimiz dikkate alınan niteliklerin genişliği, uygulanan makine öğrenimi yöntemleri ve deneysel değerlendirmede kullanılan ÇSA verisinin boyutu bakımından mevcut çalışmalardan farklıdır. Her bir alt problem ÇSA verisi ile ilgili farklı bir sorunu ele alsa da elde edilen deneysel sonuçlar genel olarak ÇSA verisinin çok kirli ve eksik olduğunu göstermektedir. Ek olarak sonuçlar, Türk kullanıcıların genellikle gizlilik risklerinin farkında olduğunu, ancak bilgiler gizli tutulsa dahi ifşa edilebileceğinin farkında olmadıklarını göstermektedir. ÇSA odaklı çalışmalarda, kullanıcılar arasındaki sosyal bağlantıları içeren doğası gereği genellikle yüksek doğruluk elde edilmektedir. Ancak, içerik tabanlı problemler zordur ve yüksek doğruluk elde etmek için dil bağımlı etkili adımlar gerektirmektedir.

ACKNOWLEDGEMENT

I would like to thank:

My advisors, Prof. Dr. Selma Ayşe ÖZEL and Asst. Prof. Dr. Ali İNAN for their supervision, patience, motivations, beneficial advice, and precious times for this thesis.

My PhD thesis juries, Assoc. Prof. Dr. Mustafa ORAL, Asst. Prof. Dr. Mümine KAYA KELEŞ, Prof. Dr. Yücel SAYGIN, and Assoc. Prof. Dr. Osman ABUL for their comments, suggestions, and valuable help.

My dear family, who always be supportive and love me, whatever happens.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

API	: Application Programming Interface
AUC	: Area Under The ROC Curve
BERT	: Bidirectional Encoder Representations from Transformers
BFS	: Breadth-First Search
BOW	: Bag-of-Words
CBOW	: Continuous Bag-of-Words
CPU	: Central Processing Unit
CNN	: Convolutional Neural Network
CoNLL	: Conference on Computational Natural Language Learning
CRF	: Conditional Random Fields
DBMS	: Database Management System
DBOW	: Distributed Bag-of-Words
DFS	: Depth-First Search
DL	: Deep Learning
DOM	: Document Object Mapping
DPP	: Democratic Party of the Peoples
DT	: Decision Tree
DTW	: Dynamic Time Warping
ERD	: Entity-Relationship Diagram
FCG	: First Name Centric
FIFO	: First-In-First-Out
FNM	: The number of male users having the same first name with WO
FNF	: The number of female users having the same first name with WO
GDPR	: General Data Protection Regulation
GPU	: Graphics Processing Unit
GRU	: Gated Recurrent Unit
GP	: Good Party

HP : Happiness Party
HTML : Hyper Text Markup Language
HTTP : Hyper Text Transfer Protocol
IDF : Inverse Document Frequency
IPS : Intrinsic Privacy Risk Score
IRT : Item Response Theory
JDBC : Java Database Connectivity
JDP : Justice and Development Party
KNN : k-Nearest Neighbor
KVKK : Personal Data Protection Law (Kişisel Verileri Koruma Kanunu)
LCS : Longest Common Substring
LCSeq : Longest Common Subsequence
LR : Logistic Regression
LSCC : Largest Strongly Connected Component
LSTM : Long-Short Term Memory
MBW : Modified Balanced Winnow
ML : Machine Learning
MLE : Maximum Likelihood Estimation
MUC : Message Understanding Conferences
NA : Not Available
NB : Naïve Bayes
NBM : Naïve Bayes Multinomial
NE : Named Entity
NER : Named Entity Recognition
NLP : Natural Language Processing
NLTK : Natural Language Toolkit
NMP : Nationalist Movement Party
NN : Neural Network(s)
NPS : Network-aware Privacy Risk Score

OSN	: Online Social Network
PRE	: Privacy Risk Estimation
ReLU	: Rectified Linear Unit
RF	: Random Forest
RNN	: Recurrent Neural Network
RPP	: Republic People's Party
ROC	: Receiver Operating Characteristic Curve
SA	: Sentiment Analysis
SCC	: Strongly Connected Component
SNA	: Social Network Analysis
SMO	: Sequential Minimal Optimization
SQL	: Structured Query Language
SVM	: Support Vector Machine
TC	: Text Categorization (or Classification)
TF	: Term Frequency
TF*IDF	: Term Frequency*Inverse Document Frequency
TPU	: Tensor Processing Unit
UI	: User Interface
URL	: Uniform Resource Locator
WCC	: Weakly Connected Component
WCE	: Weakly Connected Expansion
WO	: Wall Owner
WSD	: Word Sense Disambiguation
XML	: Extensible Markup Language

1. INTRODUCTION

1.1. Online Social Networks

OSNs are usually run by individual corporations (e.g., Facebook, Twitter), and are accessible via the web (Mislov et al., 2007). They are online platforms that are used by people to stay in touch with their connections and establish new connections with other OSN users based on shared features such as communities, hobbies, interests, and common friendships (Zhang et al., 2010). OSNs vary a lot and there exist a large number of social networking sites of different categories (e.g., online sharing, publishing, networking, etc.) and each type of these OSNs provides specific features of services for its users (Zhang and Philip, 2019).

For instance, users use LinkedIn for business and professional networking and give details about their professional evolution. Facebook is often preferred by users who want to connect with family members and friends. It allows users to socialize with each other by making friends, posting text, sharing photos and videos, while Twitter serves as a micro-blogging tool for users who want to read or write the latest news (Li et al., 2017). Popular OSNs like Facebook, Instagram, and Twitter serve hundreds of millions of users that are connected with over a billion links.

Figure 1.1. An example of a social network with various users and shared attributes.

Figure 1.1 depicts an example of a social network graph with nodes (i.e., users) and their connections (Siddula et al., 2018). As a result of the content-rich activity of OSN users, data collocated through OSNs have attracted extensive attention from both commercial, governmental, and academic organizations. Governments are considering OSNs as a means for informing, engaging with, and serving their citizens (Mfenyana et al., 2014). Researchers of various fields ranging from sociology to computer science have shown interest in OSN data (Xiao et al., 2012; Wani et al., 2018). Sociologists rely on OSN data to study human behavior (Wong et al., 2014).

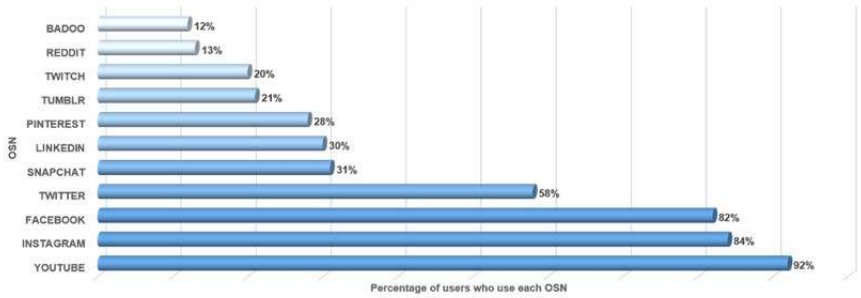


Figure 1.2. Percentage of users who use each OSN in Turkey.

Within computer science, on the other hand, OSN research has focused on various problems such as extracting useful information (Siwag et al., 2014; Kridalukmana, 2015), analyzing user behavior and social activity (Terrana et al., 2014; Terrana et al., 2015), finding similar users (Fire et al., 2014), identifying fake profiles (Gupta and Kaushal., 2017), analyzing opinion trends, and content-sharing behavior (Mfenyana et al., 2014) among many others.

Facebook is one of the most popular OSNs in the world. Figure 1.2 depicts OSN usage statistics in Turkey, which also shows that Facebook is one of the most popular OSNs in Turkey as well (Wearesocial, 2019). The explosive growth of Facebook has caught the eyes of researchers from different disciplines. Much of the existing studies on Facebook have focused on different aspects of its usage,

continuance, identity presentation, and privacy concerns (Ellison et al., 2007; Shi et al., 2010). In this thesis, therefore, our selection is Facebook to study a comprehensive attribute inference over real-world Facebook data collected from accounts of users from Turkey. We aim to show that OSN users are at higher risk of privacy than they believe, as their private attributes can be inferred/learned in an automated way by using different techniques like ML.

1.1.1. An Overview of Facebook

Initially, Facebook was founded in 2004 as a communication tool for Harvard students, but today it is the largest OSN in the world (Blachnio et al., 2013) with an average of 1.56 billion daily active users (indicating an impressive 8% yearly increase) for March 2019 (Facebook 2019). Today, Facebook is not just a communication tool and it plays different roles in different fields such as gaming, news sharing, advertising, learning, marketing, and so on (Comer et al., 2012). For instance, it is widely used by many companies that endeavor to create customer communities around pages, namely, fan pages that are presenting their products or services (Blachnio et al., 2013). In the next two subsections, we present the general properties of Facebook and its privacy protection mechanisms respectively.

(1) General Properties of Facebook: Accessing Facebook is free of charge and Facebook allows users to create personal profiles that include their basic information such as name, gender, birthday, and personal interest (Wilson et al., 2012). Users can connect with other users by friending which means that each connection is an undirected link between two befriended users (Blachnio et al., 2013). This is because friendship in Facebook is bidirectional unlike other popular OSNs such as Twitter and Instagram (Wani and Jabin, 2018). Therefore, it is possible to know that there exists an implicit friendship link between two befriended Facebook users, even though one of them keeps this friendship relation

private. This is one of the most important properties of Facebook in its nature and it is mostly used to form the basis of many inference-oriented studies.

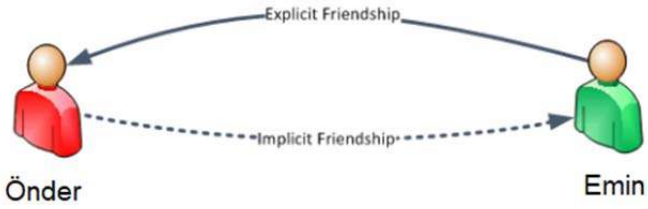


Figure 1.3. Example of implicit and explicit friendships on Facebook.

Friendship relation on Facebook is exemplified in Figure 1.3 (Jernigan and Mistree, 2009) which shows that although Önder's profile is private, his friendship with Emin can be discovered simply by viewing Emin's public Facebook profile.

Facebook users can also create and join virtual groups, share content, and learn about other users' information through their online profiles (Quan-Haase and Young, 2010). Each profile also includes a message board called *"Wall"* or *"Timeline"* that serves as the primary messaging mechanism between friends (Wilson et al., 2012). Facebook users can also chat with each other and send private messages to each other.

Figure 1.4. Screenshot of a Facebook account with respect to its pages.

Figure 1.4 depicts a screenshot of a Facebook account as of April 2020. As seen from this figure, a Facebook user has five sections in his/her account

including *Wall (zaman tüneli)*, *About (hakkında)*, *Friends (arkadaşlar)*, *Photos (fotoğraflar)*, and *More (diğer)* pages. Wall contains all posts of account owner along with shared posts of both his/her friends and other sources such as liked pages. About page includes basic attributes (i.e., gender, date of birth, etc.) and some other information about the user such as important events, family members, private relationships, and so on. For instance, if a user discloses his/her relationship information, it is worth noting that anyone can access this information in the *Family and Relationship (Aile ve İlişkiler)* section.

An example case of this scenario is depicted in Figure 1.5 which shows that revealed user accounts may be in a passive state in some cases due to any reason. Friends page, on the other hand, provides a detailed list of friends of the account owner.

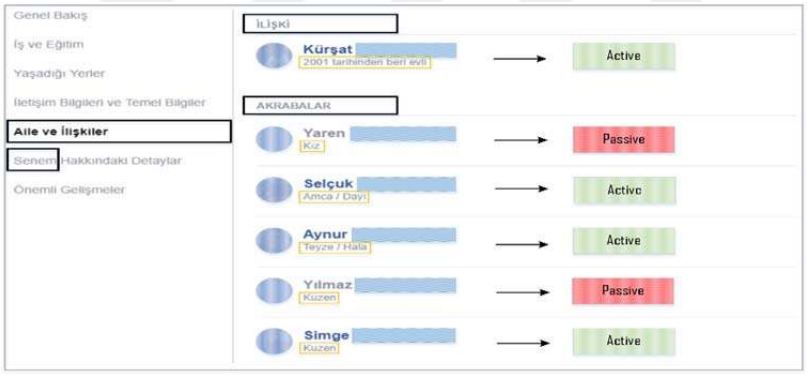


Figure 1.5. Kinship and private relations of an arbitrary Facebook user.

We would like to note that as of mid-2018, anyone can only view display names of passive users and he/she cannot access the profile owner's account as Facebook removed passive hyperlinks (includes profile owner's Facebook ID) for such users. In such a case, it is not possible to access a passive user's Facebook account from the profile owner's friend list, relationships, and so on. Photos page include account owner's photographs which may be uploaded by either account owner or his/her friends (Coban et al., 2020a). On the other hand, there is the main

page, namely *Home (Ana Sayfa)* that includes the *"News Feed"* component that is a stream of posts shared by the account owner and other sources liked/followed by the account owner.

Facebook users are also able to find their friends and create many other things such as pages, ads, groups, and events. Groups can include anything from graduation to hobbies and interests. Similarly, the event feature allows Facebook users to organize parties, concerts, and others to get together in the real world. Users can also become fans of anything such as people, organizations, television shows, movies, and musicians by liking their pages (Al-Lozi, 2011). Moreover, there are countless third-party applications available for Facebook users. These applications are developed by individuals who are out of Facebook employment.

Facebook users often interact with each other by posting a status update (i.e., post) that enables them to alert their network about what happened in their lives. This type of communication often takes place via text messages. Furthermore, a Facebook post may include a photo, video, live video, post of other users or pages, event or other things happen in the account owner's life. Users can also mention or *"tag"* and @reply their friends that they want to include communication. On Facebook, any post has a recursive structure in which comments (i.e., a message posted under a post) and replies (i.e., a message posted under a comment) are placed at bottom of each other like a stair. If exists, Facebook also stores several meta-data information for each wall activity such as post, comment, and reply. This meta-data includes the time of posting, owner of related activity, via, place, and other users tagged if exists.

An important feature of Facebook is the ability to differentiate posts based on their HTML structures. This is because Facebook shows usual, event, direct, and shared posts in different ways as depicted in Figure 1.6.

(a) Usual post (b) Direct post

(c) Event post (d) Shared post

Figure 1.6. Facebook post types without relative comments and replies

Details of these post types and reasons of preference by Facebook users to use them are described as follows:

- Usual post: Facebook automatically creates this classical structure if a user shares a status update about anything.
- Direct post: This type of post is mostly preferred by users to share a post on one of their friends' walls. If a user posted on another user's wall, Facebook automatically creates this structure which is different from the usual post structure.
- Event post: This type of post is automatically generated by Facebook when users add an important event (e.g., marriage, giving birth, buying a car, etc.) into their profiles.
- Shared post: Facebook users can share a post of another account which can be one of their friends, liked Facebook page of any organization, and so on. In that case, Facebook creates a structure that covers the shared post as well.

Facebook reactions are also provided to enable its users to react with the contents of the network. These reactions defined by Facebook are *"Like"*, *"Love"*, *"Haha"*, *"Wow"*, *"Sad"*, and *"Angry"* as depicted in Figure 1.7 (Giuntini et al., 2019).

Figure 1.7. Facebook reactions

(2) Privacy in Facebook: Privacy remains an ongoing problem for Facebook as users share their personal data consciously or unconsciously on this platform. Like other OSNs, Facebook also provides a couple of ways for its users to adjust their privacy settings. Users can configure these settings that enable them to make privacy decisions such as who is allowed to see information, send friend requests, and find related user through external search engines. Users can adjust their privacy settings and limit profile access at any time by choosing one of the following privacy levels:

- Everyone: Anyone on the internet
- Friends of Friends: Friends of profile owner as well as their friends
- Friends: Only friends
- Friends Except: All friends except for a specific group or individual user
- Only Me: Only profile owner

Even though we select Facebook to study in this thesis, the OSN research field focuses on different aspects of usage, continuance, identity presentation, and privacy concerns (Ellison et al., 2007; Shi et al., 2010) for other well-known OSNs as well. In the next section, therefore, we present a brief description of the main research fields in the OSN research field.

1.2. Main Research Fields in The Field

Analysis and mining of OSNs introduce basic concepts and principal algorithms that are suitable for exploring massive OSN data. It covers the tools for representing, measuring, modeling, and mining meaningful patterns from large OSN data (Zafarani et al., 2014). From 2006 on, researchers' attention turned on OSNs and many studies have been done to explore different aspects of OSNs.

1.2.1. Data Collection

To study in the OSN research field, collecting data is a necessary first step. However, despite such an abundance of studies in OSN research, publicly available OSN data is very scarce (Coban et al., 2020a). SNAP datasets provided by Stanford University is the only example (Leskovec and Krevl., 2014). The reasons behind the scarcity of OSN data are privacy concerns, complexity and volume of the OSN data, and the value of OSN data (Coban et al., 2020a).

(1) Privacy Concerns: OSN service providers are obliged by laws and regulations such as the GDPR in Europe (Voigt and Von dem Bussche., 2017) and KVKK in Turkey (Turan, 2012) to protect the individual privacy of their respective users. As a result, direct access to OSN data is limited only to public shares by the users, or permission-based API calls.

(2) High complexity and a large volume of unstructured data in OSNs: OSN data by its nature needs to be modeled as a graph. OSN data includes various forms of interactions (friendship, posts, likes, etc.) between users and various data types (e.g., text, emoticons, etc.). As such, building an efficient data model to store and analyze OSN data is a challenging and time-consuming task (Wani et al., 2018). For instance, the estimated size of network traffic required to disclose the entire Facebook OSN is approximately 750 Gigabytes (Wong et al., 2014).

(3) Monetary value of the OSN data: The value of many OSN service providers is rooted in the customer base. Facebook has always been a well-known

example of this fact (Gamboa and Gonçalves, 2014). The rich interactions of users within the OSNs create value and service providers prevent data collection efforts especially due to this value.

Depending on the reasons given above, obtaining high-quality OSN data is a challenging task that has been faced by many researchers before. Even some studies use various techniques (i.e., collecting manually, making online or offline surveys, etc.) to collect OSN data, existing studies within computer science obtain OSN data mainly depending on two approaches, namely, API approach and HTTP approach. The former is the most common way and means using APIs which are developed by OSN service providers (Motamedi et al., 2013; Terrana et al., 2014). The latter, on the other hand, uses HTTP requests and responses to retrieve public OSN data by imitating a human user's interactions with the OSN servers (Abdesslem et al., 2012; Terrana et al., 2015).

Each approach has its own strong and weak sides. With the API approach, the data collector acts as an OSN application developer and has to abide by the service provider's strict access restrictions and limitations such as limited view and queries (Motamedi et al., 2013). For instance, it is not possible to obtain the data of more than 25 likes or comments for a post on the Facebook OSN (Kastrati et al., 2015; Terrana et al., 2015). On the contrary, the HTTP approach requires implementing own crawler that does not have permission requirements and be able to access far more accounts (Xiao et al., 2012). Crawlers also do not have to tackle the challenges imposed by the APIs developed by OSN service providers.

However, with the HTTP approach, the data collector has to impersonate a human user. Sending too many HTTP requests within a time frame raises suspicion by the service provider and the corresponding OSN account gets blocked temporarily or permanently due to malicious activity. Additional difficulties can be listed easily: masking data requests behind click streams, extracting OSN data from HTML responses, and dealing with web browsers instead of a programming interface (Coban et al., 2020a). Researchers from computer science often prefer to

use OSN APIs or design their own crawlers to collect data depending on the amount and their purpose of use.

In this thesis, we selected Facebook as our target OSN and collected public Facebook data of users from Turkey with the help of our crawler which uses the HTTP approach (see Section 4.2 for further details).

1.2.2. Privacy and Security Over OSNs

Another key research thread deals with issues of privacy over OSNs. It is well known that OSNs ask their users for personal information and that many of those users happily provide it (Gayo Avello, 2011). Even OSNs allow users to control and customize which of their personal information is public to other users or applications, privacy emerges as an important concern (Lindamood et al., 2009). The main reasons for this concern stem from the nature of OSNs and possible malicious use of individual data. An OSN service has full access to any information of its members including the data they want to keep private. Access to the user's data often can be used by OSN service providers to enhancing (e.g., to provide personalized service and to understand user characteristics and behaviors) their service and providing useful applications.

Such services, however, may violate the user's privacy themselves even they have strict privacy policies. For instance, according to Facebook's own statement in the Cambridge Analytica scandal (Cadwalladr and Graham-Harrison, 2018), the profile information of more than 87 million users may have been used inappropriately for political purposes. According to *Time* magazine, the Russian Federation performed a phishing attack by sending expertly tailored malware messages to more than 10 thousand Twitter users employed by the U.S. Department of Defense for influencing the 2016 presidential election (Calabresi, 2017). The *Guardian* has covered the news that trillions of tweets would be put up

for sale (Garside, 2015). In a much recent case, a collection of popular Facebook quiz apps exposed the private data of up to 120 million people (Nick, 2018).

Moreover, third parties can have direct or indirect access to the OSN data by different ways (e.g., crawling) and they do not only could use the resulting information for online stalking and targeted advertising, but they could also use or sell it to others with unknown nefarious intentions (Tang et al., 2011) alongside the threats by OSN service providers.

All of these cases show that alongside the sunny bright world where OSN users vividly socialize with their friends digitally, there is a much darker facet of OSN services where what we share might do us harm. Fortunately, there also has been considerable research effort put into understanding the potential extent of this harm. Zheleva and Getoor examine the potential privacy threats caused by misuse of OSNs under the following four categories (2011):

- Identity disclosure: Detecting a user's account on anonymized OSN data or across data of multiple OSNs (Backstrom et al., 2007; Narayanan and Shmatikov, 2009; Wondracek et al., 2010).
- Attribute Disclosure: Disclosing information (e.g., language, gender, religion, age, kinship, etc.) that a user would prefer to keep private (Lindamood et al., 2009; Tang et al., 2011; Dey et al., 2012; Kokkos and Tzouramanis, 2014).
- Link Disclosure: Disclosing a social relationship (Backstrom et al., 2007; Korolova et al., 2008).
- Affiliation Disclosure: Disclosing whether a user is included in a particular user group. This type of disclosure may also lead to identity, attribute, or social link disclosures in some cases (Zheleva and Getoor, 2011).

An adversary who has access to the OSN data can utilize statistical and ML techniques to perform privacy attacks given above (Tang et al., 2011). Nevertheless, identity disclosure and attribute disclosure have been more studied traditionally (Zhaleva and Getoor, 2011).

(1) Attribute Disclosure: The widespread use of OSNs has brought forward the issue of privacy as users' sensitive information needs to remain private. Privacy is an important concern as while the distribution of information in the real world is almost local, publicly shared information in OSNs can be retrieved on the internet anytime, anywhere, and by anyone (Aghasian et al., 2017). Besides, private information of OSN users can also be effectively inferred by using different techniques including graph analysis and probabilistic classification (Talukder et al., 2010, Akcora et al., 2012). For instance, it is possible to estimate some private traits of a user's personality by leveraging his/her Facebook activities (Kosinski et al., 2013). Another study states that it is even possible to infer user characteristics from the attributes of users who are part of the same community (Mislove et al., 2010). On the other hand, third-party OSN applications and befriended users may also put the information owner at risk in OSNs (Vidyalakshmi et al., 2014).

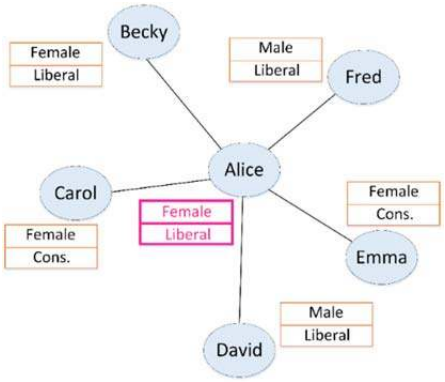


Figure 1.8. Private attribute inference using friendship relations.

Despite such risky matters, attribute inference can also be used for good purposes such as personalized recommendation and targeted advertising (Tuna et al., 2016). Thus, many studies have been performed to automatically identify user attributes from OSN data which is so rich data source that it is possible to disclose private attribute without a learning model especially when the data is complete. For instance, an adversary can disclose various confidential information of users just by using social connections between (e.g., friendship, following, etc.) users. Figure 1.8 shows an example inference by looking at the most frequent attribute values in friends' profiles (Talukder et al., 2010). As seen from the figure, *Alice* keeps her political view and gender private, but considering her friendship relations it is possible to infer these attributes.

Even studies often focus on detecting a single attribute of users, other studies focus on multi-attribute detection over OSNs. Multi-attribute detection helps to have wider insight into the interests of users. For instance, the recommendation of a product may depend on both gender and age as women at different ages may be interested in different subjects (Tuna et al., 2016). In the literature, existing studies use profile information, social connections, or user-generated contents which include various attributes of users to perform attribute inference over OSNs.

In this thesis, we performed attribute inference tasks to infer the gender and kinship attributes of users. In these tasks, we employed traditional ML techniques over real-world datasets that are created from the crawled Facebook data (see Section 5.2.2 for further details).

(2) Identity Disclosure: Another traditional privacy-related task is identity disclosure which aims to map an OSN account to a specific real-world entity, find whether a particular individual has an account in the OSN, or find whether two OSN accounts belong to the same user (Zhaleve and Getoor., 2011). One of the popular reflections of this problem over OSNs is to match users across different OSNs. As depicted in Figure 1.9 (Raad et al., 2010), this is because today people

tend to create multiple OSN accounts for different purposes such as finding new friends, discussing opinions, playing online games, and so on (Peled et al., 2013).

Based on this reality, each OSN uses its own way to store and display the information about its users (Vosecky et al., 2009) and users disclose different aspects of their information which is often incomplete (Zafarani and Liu., 2013; Li et al., 2017). Matching users, therefore, can benefit inter-social network operations and functionalities such as aggregating user information, friend recommendation, searching a user, and so on (Vosecky et al., 2009; Zafarani and Liu., 2013; Goga et al., 2015).

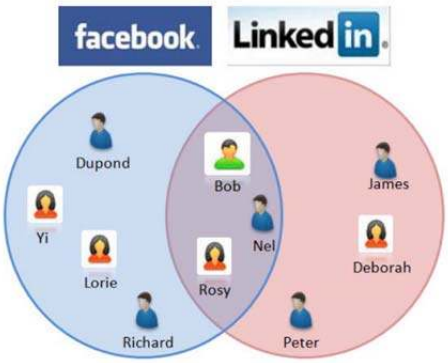


Figure 1.9. Matching users across OSNs. Bob, Nel, and Rosy have accounts in both Facebook and LinkedIn

User profile matching, however, is not a straightforward task as the connectivities of user identities across different OSNs are often unavailable. This is because users are free to select usernames they want instead of their real identities and OSNs rarely link their user accounts with other sites and services (Zafarani and Liu., 2013). As such, identifying users across different OSNs is an attractive topic for researchers who studied this problem based on profile information (Vosecky et al., 2009; Raad et al., 2010; Bennacer et al., 2014; Goga et al., 2015; Wang et al., 2016; Li et al., 2017; Li et al., 2019), user-generated contents (Peled et al., 2013;

Sha et al., 2016; Hazimeh et al., 2017; Bhat et al., 2018), and network structure (Bennacer et al., 2014; Goga et al., 2015; Zhou et al., 2015).

Unfortunately, studies that use especially content and network attributes tend to fail as publicly shared data is often unavailable, incomplete, or unreliable (Li et al., 2019). On the other hand, display names and usernames are the most discriminative features for user identification (Vosecky et al., 2009) as they are fundamental elements of all websites and can reflect a user's characteristics and habits.

In this thesis, we performed an inference task to identify/learn other OSN accounts of Facebook users with the help of the user matching (see Section 5.2.1 for further details).

(3) Privacy Scoring: Researchers have also been trying to measure the privacy risk of OSN users. Popular OSNs demand users to share more data even by providing their sensitive information to make other users can find them more easily (Liu and Terzi, 2010; Caramujo and Da Silva, 2015). As a result, OSN users often willingly disclose their personal information (Vidyalakshmi et al., 2015b) and they are not often aware of who will or will not have access to what they have just published (Sramka, 2015; Alemany et al., 2018). Any information shared over OSNs may be sensitive and any kind of disclosure (including inference with automatic ways) can significantly harm the privacy of users (Srivastava and Geethakumari, 2013) by giving way to some threats.

These threads may occur as identity theft, digital stalking, building consumer models for advertising, identity cloning, and blackmailing among many others (Liu and Terzi, 2010; Vidyalakshmi et al., 2014; Srivastava and Geethakumari, 2015; Veiga and Eickhoff, 2016; Oukemeni et al., 2018). As such, both users and OSN service providers recognize the need to ensure that user privacy is well preserved in OSNs (Sramka, 2015). OSN service providers have considerably improved their privacy protection tools by introducing more controls

like groups, lists, and circles (Caramujo and Da Silva, 2015; Aghasian et al., 2017; Vidyalakshmi et al., 2015a; Vidyalakshmi et al., 2015b).

However, the most powerful data protectors are the users themselves (Bioglio and Pensa, 2017), as a user's level of privacy attitude determines its sharing activities on any OSN. A privacy-aware user tends not to share his/her own and his/her friends' sensitive or private information, whereas an unaware user does not carry such a concern. On the other hand, access control policies enforced by OSN providers are based on privacy settings which are confusing, time-consuming to manage, and make users face with too many options, controls, and lack of understanding privacy risks (Gundecha et al., 2011; Minkus and Memon, 2014; Srivastava and Geethakumari, 2014; Alemany et al., 2018; Oukemeni et al., 2019). Therefore, determining privacy risks when publishing information on OSNs often presents a challenge for users.

A measure of how much sensitive information is shared by users with others on an OSN would help the users to understand whether they individually share too much (Sramka, 2015). However, privacy measurement is a challenging task because privacy does not have a specific definition as its degree differs from individual to individual. In other words, there are different perceptions and concerns of privacy among different cultures, notions, societies, and religions (Srivastava and Geethakumari., 2015; Al-Asmari and Saleh., 2019). Even so, there has been a large number of studies with a different point of views which consider privacy in OSNs.

Conventional privacy-related studies mainly focus on protecting or measuring identities or private attributes of users (Wang et al., 2019). Unfortunately, content-based measuring has not been strongly addressed before and existing privacy models for structured data are inadequate to capture privacy risks from user textual contents (Biega et al., 2016). However, a lot of users' personal aspects can be extracted from user-generated contents (Song et al., 2018), as a text message or post may directly or indirectly disclose personal information

like user's gender, age, political view, interest, and hobbies that many users do not prefer to reveal (Caliskan Islam et al., 2014; Biega et al., 2016; Al-Asmari and Saleh., 2019).

In literature, only a few studies (Biega et al., 2016; Song et al., 2018; Wang et al., 2019) consider content in privacy scoring, whereas most of them does not include methods for obtaining/driving/inferring sensitive information from the contents (Srivastava and Geethakumari., 2013, Cetto et al., 2014; Vidyalakshmi et al., 2015a; Li et al., 2018). Content-based sensitivity detection is also studied in some studies, but these studies do not address the problem in terms of privacy measurement (Thomas et al., 2010; Mao et al., 2011). Instead, these studies often focus on detecting sensitive contents or classifying contents in more general private information (i.e., vacation, location, health, etc.) and personal attributes (Thomas et al., 2010; Mao et al., 2011).

There are also studies on OSN security. These studies try to either expose the inadequacies of existing OSN frameworks or suggest new ones. Some of the previously published studies on this topic are as follows: Cutillo et al. (2009) propose a distributed OSN framework called the Safebook (obviously, the name was inspired by Facebook) that caters to individual privacy and security by design. Krishnamurthy and Wills (2009) show that cookies placed by OSN websites can be obtained by listening to network traffic, thereby violating personally identifiable data such as gender, mailing address code, and e-mail address. Similarly, Gross and Acquisti (2005) examine the effects of OSNs on individual privacy and security over a sample of Carnegie Mellon University students as of 2005.

In this thesis, we first performed a preliminary privacy analysis of Turkish Facebook users using state-of-the-art techniques. Then, we also performed a comprehensive privacy risk analysis of users using many different attribute inference mechanisms. In this extended analysis, we inferred attribute values based on the contents and social connections of users. Differently from the existing

studies, we also suggested using social tie strengths of users when computing network-aware risk scores (see Section 5.2.3 for further details).

1.2.3. NLP Over OSNs

OSNs with all their features have become an important focus in social communication and their data provides a valuable insight into behavior, experiences, opinions, and interests (Villota and Yoo., 2018). As such, OSNs have also got the attention of the research community for different text mining tasks such as attribute inference (Peersman et al., 2011), author profiling (Fatima et al., 2017), SA (Ahkter and Soria, 2010), and NER (Okur et al., 2018) as the most important and well-known characteristic of OSNs is that users communicate each other via short text messages (Peersman et al., 2011).

(1) Named Entity Recognition: Even state-of-the-art NER systems give highly accurate results in the domain of formal texts, the informal domain has become another new trend for NER studies with the help of the expansion of OSNs (Eken and Tantug, 2015). NER can be defined as identifying and categorizing a certain type of data and aims to classify and locate atomic elements in a given text into predefined entity types (Okur et al., 2018). Traditional NER consists of three subtasks that aim to group entities into seven categories. These coarse-grained NE types and subtasks are: (i) ENAMEX which aims to locate NEs including PERSON, LOCATION, and ORGANIZATION (PLOs), (ii) TIMEX which aims to locate absolute temporal expressions only including DATE and TIME, and (iii) NUMEX aims to locate numeric expressions by covering the type of MONEY and PERCENT expressions (Sundheim, 1995; Tur et al., 2003).

Among these categories, ENAMEX is the most used category in the NER field as PLOs are more informative than the other types. Other studies address the fine-grained NER in which there are more than four entity types to be captured (Sahin et al., 2017). NER systems are important modules in the systems of

information extraction, question answering, media monitoring, summary extraction, machine translation, SA, and TC (Küçük et al., 2016a; Okur et al., 2018; Ertopçu et al., 2017). There are different approaches in the NER field including probabilistic (supervised ML), symbolic (rule-based), and hybrid approaches which use tagged corpora, dictionaries, or gazetteers (Bayraktar and Temizel, 2008).

Even the NER studies are performed for English at first, the satisfaction on English NER tasks led the field to new research areas (e.g., multilingual NER systems) and the state-of-the-art NER systems have been produced for several languages (Özkaya and Diri, 2011; Şeker and Eryiğit, 2012). However, compared to the other languages, studies in Turkish are limited both in terms of scope and number (Küçük et al., 2016a; Küçük and Arıcı, 2016b; Sahin et al., 2017).

Studies in the context of OSNs often focus on Twitter and studies focusing on Facebook data is very scarce especially for Turkish. NER on this informal domain remains to be a challenging task as user-generated contents are generally containing noisy texts which have grammatical errors.

In this thesis, therefore, we additionally performed NER by using our CRF-based NER system. To achieve this, we created a manually labeled NER dataset, namely, FBNER, and extracted coarse-grained NEs from users' textual contents (see Section 5.1.2 for further details)

(2) Sentiment Analysis: In the early stages, SA was oriented mostly towards the classification of reviews, but with the increasing popularity and widespread use of OSNs, nowadays SA has also been used for retrieving, extracting, and classifying sentiments from user-generated OSN content (Balahur and Jacquet, 2015). This is because OSNs provide an opportunity to collect vast amounts of data as they have millions of interacting users who express and share their views on a wide variety of subjects such as items, products, movies, political parties, and events (Akaichi et al., 2013; Süerdem and Kaya, 2015).

SA is a language-dependent task to classify texts according to their polarity which can be positive, negative, or neutral (Ortigosa et al., 2014, Balahur and Jackuet, 2015). Automatic extraction of the sentiment from a text can help to analyze what people think about specific issues (e.g., political issues), brands, items, and so on (Gezici and Yanıkoğlu, 2018). SA techniques can also be employed for commercial purposes such as giving recommendations, advertising, and capturing public opinion about products.

The majority of SA studies performed on OSNs focus on Twitter (Ahkter and Soria, 2010), whereas Facebook has attracted less attention from the research community. The possible reasons for the scarcity of SA work over other OSNs are many folds: collecting (even public) data from other OSNs is not easy compared to Twitter and the primary motive of the use of these OSN is rather to connect with family members and friends (Wilson et al., 2009), search job or use location-based applications. In contrast, Twitter is considered an effective tool for rapid dissemination of information (e.g., for digital diplomacy) and for connecting with communities rather than individuals (Li et al., 2017).

The majority of SA studies focus on Twitter OSN as stated before. However, Facebook activities (i.e., posts, comments, or replies to these) are more succinct than item reviews and are easier to classify than tweets as they may contain more characters that Facebook allows its users to write longer texts, therefore leading to a more accurate portrayal of emotions (Ahkter and Soria, 2010). In terms of the language context of existing works, most of the research in SA over OSN data has focused on English and Chinese (Habernal et al., 2013). For the specific case of SA over Turkish text posted on Facebook, there is only one study (Süerdem and Kaya, 2015) where the analyses are performed on a very small and manually labeled dataset. This shows that automatic SA over Facebook has not yet been thoroughly targeted by the research community in the context of the Turkish language. The primary reason behind such scarcity could be the lack of

resources and techniques in this field for the Turkish language (Dehkharghani et al., 2016).

In this thesis, we performed SA over Facebook data to explore the sentimental orientation of Turkish users. We employed different methods from both ML and DL fields over our manually labeled SA dataset (see Section 5.1.3 for further details).

1.3. The Aims and Objectives of this Thesis

The explosive growth of OSNs has caught the eyes of researchers from different disciplines including computer science which has focused on different aspects of OSNs such as usage, opinion, and trend analysis, privacy and security analysis, and so on. From the literature, we understand that majority of studies often focus on Twitter, and Facebook has attracted less attention from researchers. Another gap is that the number of studies focusing on Turkish OSN users is very restricted. Motivating by these gaps, this thesis covers a comprehensive privacy analysis of Turkish Facebook users.

We aim to reveal how much private personal data can actually be learned based on the publicly shared information of users. We make this investigation by performing different attribute inference tasks to learn/infer gender, kinship, and OSN account attributes. Additionally, we perform another extended attribute inference task that covers many other attributes including political view, e-mail, date of birth, and so on. In this inference task, we use different rule and lexicon-based methods to infer attribute values. We then use values of the inferred attributes to show that users are at higher privacy risk than they believe.

To perform all of these tasks, we first focus on the problem of efficient collection of large-scale, content-rich, and accurate Facebook data while respecting the individual privacy of the respective OSN users. Accordingly, we crawled three different snapshots of Facebook data of Turkish users and performed sharing behavior analysis.

OSNs have also got the attention of the research community for different text mining tasks as users often communicate with each other with text messages (Peersman et al., 2011). Facebook data includes user interactions, wall activities (i.e., posts, comments, and replies), and basic profile information of users. Moreover, extracted information from the textual contents of users can be used for different purposes including privacy risk analysis. Taking this truth into consideration, we also choose NER and SA tasks to study over the crawled Facebook data. This is because NER and SA are seeming to be popular text mining tasks that have been studied for other languages and OSNs. Unfortunately, studies focusing on these tasks are very scarce in terms of Turkish Facebook users.

1.4. Contributions of this Thesis

Studies covered in this thesis aim to perform a comprehensive analysis of Turkish Facebook users with respect to different aspects such as sharing behavior, SA, NER, privacy measuring, and private attribute inference.

For the data collection and sharing analysis task, we proposed a system that has various advantages over existing OSN data collection efforts. To summarize the key differences, our system supports multi-threaded data collection, is resilient against access restrictions imposed by the Facebook OSN, collects every bit of publicly shared data, and respects BFS order from the seed account. After crawling, we analyze the crawled data to summarize the sharing behavior of Turkish Facebook users (Coban et al., 2020a). To the best of our knowledge, this is the first effort that only regards the users from Turkey and provides their behavioral analysis.

Our tasks which are related to private attribute inference present the-state-of-the-art results and they have not been considered before with respect to Turkish Facebook users. For the gender detection task, we performed the first research into the gender classification of Turkish Facebook users based on different information extracted from users' profiles. In this task, we employed different models including

profile information, wall interaction, and wall content models to explore which information of users is more effective in the gender detection task. We also introduced several new features in the wall interaction model which is easy to apply for other language-dependent gender detection tasks by providing a lexicon of gender-oriented words.

For the kinship detection task, we investigated whether it is possible to detect kinship between two Facebook users based on their wall contents. We believe that this is the first research for fine-grained kinship detection with respect to both information considered and language in OSNs. Our results are promising and show that content-based kinship detection is can be a good starting point for future studies even it has some challenges.

For the user matching task, we build a learning model to say that two Facebook accounts are belonging to the same user or not. We perform this task just based on usernames as our crawled Facebook data is incomplete. We conduct our experiments on our real-world username dataset and obtained quite reasonable results compared to the results of the existing studies.

Our text mining-related tasks are the first of their kind with respect to the targeted OSN, the language of content, and the used approach. For the NER task, we developed a CRF-based NER system and employed it to discover NEs from Facebook posts. In order to do so, we created a dataset, namely, FBNER, which contains Turkish posts that automatically crawled from the Facebook OSN. To the best of our knowledge, this is the first study to extract NEs from Facebook posts.

For the SA task, a manually SA dataset that includes Turkish texts collected from public Facebook activities were created. This task differs from existing studies in terms of the dataset scale, the natural language of texts in the dataset and the extend of experimental analyses which include both ML and DL techniques. We extensively reported not only the results of different learning models involving SA but also the statistical distribution of meta-data of user activities across various attributes such as gender and age. Our experimental results

indicate that RNNs achieve the best result with word embeddings. To the best of our knowledge, this is the best result for SA on Facebook data in the context of the Turkish language (Coban et al., 2021).

Finally, for the privacy analysis, we explored privacy risks of Turkish Facebook users with respect to their age and gender attributes using state-of-the-art techniques (Coban et al., 2020b). Besides, we extended our analysis by incorporating social tie strengths and attributes inferred from the contents and social connections of users into privacy risk measuring. To the best of our knowledge, this is the first research into privacy scoring of Turkish Facebook users based on attribute inference from contents.

1.5. Outline of the Thesis

This thesis is organized as follows: **Section 2** provides a task-dependent review of literature for each task performed in this thesis. **Section 3** covers materials (i.e., datasets, lexicons, and tools) that are used in building our models, while **Section 4** explains methods used to preprocess data, extract features, and build models. **Section 5** presents experimental results and discussions for each task, and finally, **Section 6** gives conclusions of this thesis.

2. PREVIOUS WORKS

OSNs have become popular platforms for users to connect, express themselves, and share other information such as activities, photographs, favorites, and posts (Viswanath et al., 2009; Abdesslem et al., 2012). Popular OSNs like Facebook, MySpace, and Twitter serve hundreds of millions of users that are connected through more than a billion links. These OSNs are even said to be able to reflect real-life (Wani et al., 2018). Over the fourth quarter of 2019, Facebook had over 2.50 billion monthly active users (indicating an impressive 8% yearly increase (Mfenyana et al., 2014). It should not be surprising that OSNs are among the most visited websites rivaling many popular search engines (Modi and Gandhi, 2014).

As a result of this popularity, the data collocated with OSNs have attracted extensive attention from both commercial, governmental, and academic organizations. Governments, for example, are considering OSNs as a means for informing, engaging with, and serving their citizens (Mfenyana et al., 2014). In the academy, researchers of various fields, ranging from sociology (Modi et al., 2013) to computer science (Xiao et al., 2012; Wani et al., 2018) have shown interest in OSN data.

Within computer science, OSN research has focused on various problems such as extracting useful information (Siwag et al., 2014), analyzing user behavior and social activity (Terrana et al., 2014), finding similar users (Terrana et al., 2015), identifying fake profiles (Conti et al., 2012), opinion trend analysis (Mfenyana et al., 2014), sentiment analysis (Troussas et al., 2013; Vashisht and Thakur, 2014; Trinh et al., 2016; Coban et al., 2021), named entity recognition (Özkaya and Diri, 2011; Okur et al., 2018), content-sharing behavior (Jansen et al., 2011; Coban et al., 2020a), and OSN usage (Wong et al., 2014) to list a few examples.

In this section, we review previous studies which are related to data collection, NLP tasks including SA and NER, privacy scoring, and attribute inference including kinship and gender detection over OSN data.

2.1. Data Collection from OSNs

In OSN research, data collection is a necessary first step, and collected data does not only provide users' information but also describes their social interactions. However, OSN datasets are not publicly available, due to privacy concerns and it makes the data collection process complex and time consuming (Wani et al., 2018). This task is generally performed by using two different ways: the first one is using API which is provided by OSN provider (Motamedi et al., 2013; Terrana et al., 2015). The other alternative way is to design a crawler (i.e., a program which starts with a list of user's page to visit) that uses HTTP requests and responses to retrieve public data shared on social network sites (Abdesslem et al., 2012).

Obtaining high-quality OSN data is a challenging task that has been faced by many researchers before. Due to its high world-wide popularity, Facebook has been one of the most heavily studied OSNs (Wong et al., 2014). In this section, we present our review of the literature with a focus on studies collecting data by using the API approach and HTTP approach. We would like to note that these studies were compiled through careful analysis of prominent journals with the keywords *"online social networks"*, *"crawler"*, and *"data collection"*.

Creating the desired dataset is a challenge faced by researchers who aim to analyze OSNs. However, Facebook has a great potential of the data and it is one of the most studied OSNs (Wong et al., 2014). Therefore, there are many research works related to OSNs with different purposes. Viswanath et al., (2009) crawled 90,269 Facebook users' information to evaluate the activity between users from the same regional network. Their two-step crawler starts from a single user and visits all friends of the user by using the BFS method. Catanese et al., (2011) implemented a Facebook crawler by using two different sampling methods,

namely, BFS sampling and uniform sampling. They faced a limitation that a crawler can obtain at most 400 friends of a user and analyzed the collected OSN data in terms of different graph metrics. Abdesslem et al., (2012) introduced a methodology to collect more reliable data on the sharing behaviors of OSN users.

Xiao et al., (2012) designed and implemented a Facebook crawler that can retrieve the complete friend list of a user. Their crawler is based on interaction simulation and overcomes the drawback faced by Catanese et al. (2011). They have created a dataset that contains a total of 262,526 users. Flores et al., (2013) implemented a Facebook crawler by using Selenium API and they crawled public data from 500 friends and their second level friends of two politicians. They have analyzed the collected data by using text mining techniques to investigate how information and influence are spread through an OSN for election purposes. Motamedi et al., (2013) implemented three independent crawlers for most popular OSNs (i.e., Facebook, Twitter, and Google+) to analyze and compare users' popularity and activity which can provide insight to their users to decide which OSN to use.

Wong et al., (2014) designed a Facebook crawler by using a headless browser and they modeled data as a graph in which they visualized and analyzed the crawled data. Mfenyana et al., (2014) implemented a focused Facebook crawler by using RestFB (i.e., Facebook Graph API client written in Java) and Jackson libraries that are written in Java. They proposed a system prototype to perform opinion monitoring and trend analysis on Facebook. Terrana et al., (2014) used a Facebook crawler to analyze user relationships based on sentiment classification of user-generated content such as posts, comments, and likes. Siwag et al., (2014) proposed a crawler architecture for Facebook and clustered crawled data by several criteria such as gender, location, and hometown.

Terrana et al., (2015) crawled different OSNs for the comparison of user-profiles by using textual data (e.g., posts, comments) shared by users. Their crawler has a separate module for each OSN and Facebook crawler module uses Facebook

Graph API. Kastrati et al., (2015) used Facebook Graph API to crawl only posts, feeds, and comments of a user. They use their collected data for social criminal activity analysis. Kridalukmana (2015) crawled Facebook by using Facebook Graph API and transformed data into an attributed graph to illustrate the object (i.e., user, page, or photograph) interrelations.

Passaro et al., (2016) crawled Facebook pages of the most popular newspapers in Italy to create a corpus that can be used for sentiment analysis and emoticon detection. Chen et al., (2016) collected posts and interactions of 859 users with the help of Facebook Graph API. Mittal and Sahu (2017), collected 70K tweets of their selected famous accounts from Twitter. They performed language detection experiments on collected data. Abid et al., (2018) implemented a Facebook crawler with the help of the Selenium API to explore the social network around a user by considering a distance parameter. They proposed a tool, namely, SONSAI that predicts user sensitive attributes. Wani et al., (2018) designed a Facebook crawler, namely, IMcrawler, and they crawled information of around 10K user profiles from four metropolitan cities of India. In their work, they performed a behavioral analysis of users based on both profile features and wall activities.

As seen from the literature, researchers prefer to use OSN API(s) or design their own crawlers to collect data depending on the amount, and their purpose of use. In this thesis, we preferred to implement a crawler (see Section 4.2) without using Facebook Graph API to analyze content-sharing behaviors of Turkish users (see Section 5.1.1).

2.2. NLP Over OSN Data

This section provides an overview of the literature with a focus on NLP studies over OSNs. In this section, we particularly present our review of the literature with respect to the NER and SA studies.

2.2.1. Sentiment Analysis

SA focuses on retrieving, extracting, and classifying sentiments from OSNs due to their rise in usage and interest among users (Balahur and Jacquet, 2015).

The published studies with a focus on SA on Facebook OSN are as follows: Ahkter and Soria (2010) labeled messages based on observed emoticons within the texts and performed binary SA on Facebook status updates. Akaichi et al. (2013) performed SA on a small dataset which is comprised of Facebook status updates posted by Tunisian users. Troussas et al. (2013) performed SA on Facebook messages for language learning and they obtained an accuracy of 0.74 with the help of the Rocchio classifier on randomly collected messages. Habernal et al. (2013) performed SA of Facebook brand pages and they performed experiments on two datasets that have binary (i.e., positive, negative) and multiple (i.e., positive, negative, and neutral) classes.

Zamani et al. (2014) developed a tool that can perform SA by using a lexicon-based approach in which polarity is detected as percentages of the matched positive, negative, and objective words in a Facebook page. Vashisht and Thakur (2014) categorized mostly used emoticons based on their expression of sentiment as positive and negative. They employ a Finite State Machine (FSM) to detect both emoticon(s) within the text and polarity of the related text. Ortigosa et al. (2014) studied SA for Facebook messages and developed a tool, namely, SentBuk which uses lexicon-based and ML approaches. Süerdem and Kaya (2015) performed SA of two retail brands based on Turkish comments about them. The authors used IDF weighted bow features and selected the most informative ones by using the information gain algorithm. They obtained the best accuracies as values of 0.91 and 0.88 on datasets related to the first and second brands by feeding the NB classifier with selected features.

Pool and Nissim (2016) performed SA on three different benchmark datasets which include posts of different Facebook pages. Islam et al. (2016)

performed SA on Facebook comments that are written in the Bengali language. They used the NB classifier and obtained the highest F1 score as a value of 0.72 with the help of bigram features. Akter and Aziz (2016) performed SA on a dataset that includes randomly selected Facebook statuses on a group page. The authors employed both ML and lexicon-based approaches and conclude that the latter is more efficient than the former.

Trinh et al. (2016) performed lexicon-based subjectivity and sentiment classifications of Facebook comments which are chosen from education, sport, and movie topics. The authors created a Vietnamese emotional dictionary (VED) which includes five sub-dictionaries. These dictionaries contain nouns, verbs, adjectives, adverbs, and intensification words paired with hand ranked integers which describe the corresponding emotional values from the most negative (-5) to the most positive (+5). Meire et al. (2016) performed binary sentiment classification of Facebook posts which are written in Dutch language and labeled by using a distant supervision approach. The authors obtained an AUC value of 0.83 with an RF classifier on a manually labeled test dataset by using posts' variables, leading variables, and lagging variables.

Tian et al. (2017) calculated sentiment scores for each Facebook post by using emoji-based sentiment scores of all comments under the related post. They calculated each comment's score by using a sentiment-emoji score lexicon (Novak et al., 2015). Krebs et al. (2017) performed reaction prediction for Facebook posts based on two stages which are based on emotion miner and neural networks respectively. At the final stage, the LR algorithm is employed on the outputs of these two stages for reaction detection. Iram (2018) developed a system that uses tf*idf weights and Sentiwordnet scores of the chunks extracted from the related post. This system uses overall positive, negative, and objective scores of a post to detect its polarity.

Existing work on SA has concentrated on English texts obtained from Twitter feeds and user reviews on hotels, movies, products, and so on. Among

these, SA studies on Turkish especially for Facebook data are extremely scarce. Therefore, in this thesis, we perform SA on public Facebook data collected from Turkish user accounts (see Section 5.1.3 for further details).

2.2.2. Named Entity Recognition

To build successful social media analysis, it is necessary to employ successful processing tools for NLP tasks such as NER (Okur et al., 2018). In the literature, there are many studies in the NER field for different languages (Eken and Tantug, 2015) and the state-of-the-art performance has reached nearly human annotation performance on formal texts. However, the NER tools give low accuracy when they are employed on informal texts such as e-mails and microblog or social media contents (Eken and Tantug, 2015). This is because such user-generated content may be ungrammatical and can have misspelling errors, unlike formal texts.

Morphologically rich languages pose interesting challenges for NLP tasks as is the case for NER (Hasan et al., 2009). As Turkish is a morphologically rich and highly agglutinative language, it attracts the attention of the NLP community. Nevertheless, the results for Turkish NER remain behind the reported accuracies for English (Özkaya and Diri, 2011; Şeker and Eryiğit, 2012). The published works on this topic for Turkish are as follows: Tur et al. (2003) proposed a probabilistic model for NER and evaluated their model on a dataset including Turkish news articles. Bayraktar and Temizel (2008) utilized the local grammar-based (LG) NER approach to extract person names from Turkish financial texts. Dalkılıç et al. (2010) performed rule-based NER on a small dataset including texts in economic, sport, and health categories.

Yeniterzi (2011) investigated the effect of morphology in a CRF-based NER system for Turkish. In this work, the author used two-level tokenization methods to represent states, namely, word-level and morpheme-level. Özkaya and

Diri (2011) performed NER experiments on informal e-mail texts using CRF. They obtained the best result as 97% on corporate e-mail texts for person NE type. Küçük and Yazici (2011) presented a hybrid NER model based on the previously proposed rule-based recognizer. Tatar and Cicekli (2011) proposed an automatic rule learning method to extract NEs from news articles. Their method is based on the concept of the generalization of strings by processing similarities and differences between them.

Şeker and Eryiğit (2012) propose a CRF-based model for NER which is the state-of-the-art system with an F-measure of 0.919 for Turkish. They use base and generator gazetteers (also known as entity dictionaries) and morphological processing to obtain feature vectors for training and test instances. Metin et al. (2012) evaluated previously proposed association measures to extract two-worded NEs and they also proposed a new association measure. Çelikkaya et al. (2013) employed a CRF-based model, that is proposed by Şeker and Eryiğit (2012), on three different datasets from different domains including Twitter, Speech, and Forum. They evaluated their NER system by using three different feature models and they report that the system achieves the best results as 50.84%, 5.62%, and 19.28% for Speech, Forum, and Twitter data respectively.

Küçük and Steinberger (2014) performed experiments on Turkish tweets to investigate facilitating and impeding factors in the NER process. They have obtained a 53.23% best achieving rate and concluded that NER systems for a well-formed text face considerable performance decreases when they are evaluated on tweets. Demir and Özgür (2014) employed a semi-supervised learning approach based on neural networks to build a NER system for morphologically rich languages. They used a continuous representation of words in the unsupervised stage and utilized the language-independent features in addition to these word embeddings in the supervised stage. Küçük et al. (2014) proposed a new annotated tweet dataset for NER and performed a detailed analysis by using a rule-based approach.

Eken and Tantuğ (2015) performed NER experiments on tweets without using normalization and morphological processing. They conclude that the performance of the NER on tweets reached the success rate of 64.93% with the help of some basic features and the first and last four characters of the word. Küçük (2015) proposed an automatic method to compile resources for Turkish NER by using titles of the Wikipedia articles. In this work, the proposed method achieved a 91.25% success rate and improved success when used as a training set on other test sets from different domains. Kuru et al. (2016) proposed a new approach that employs character-level NER. In this approach, each sequence is represented as a sequence of character tokens. The authors state that they could obtain slightly different performance than the state-of-the-art results for seven languages without using language-dependent features and gazetteer features. Emekligil et al. (2016) developed a CRF-based NER system to extract NEs (e.g., IBAN, currency, organization name, etc.) from customer order texts. Küçük et al. (2016a) proposed an annotated dataset that only includes ENAMEX types. Küçük and Arıcı (2016b) proposed a Turkish NER system based on Wikipedia. This system is composed of sub-modules (e.g., person name recognition system) and uses previously created gazetteers in this field.

Kalender and Kormaz (2017) proposed an entity-discovery system that detects entity mentions in a text without a Wikipedia entry by using a semi-supervised learning approach. They have employed the word2vec algorithm to represent their feature set by word embeddings. This work demonstrates the potential of word vectors for fine-grained NER systems. Sahin et al. (2017) proposed a dataset that is composed of automatically categorized and annotated sentences from Wikipedia. They have extracted relevant entity and domain information from a semantic knowledge base, Freebase, and constructed large-scale gazetteers. This dataset has six different versions and it can be used in both NER and TC studies. Ertopçu et al. (2017) proposed a new approach in which they consider the problem of NER as a classification problem by using two different

evaluation models, namely, discrete, and continuous. Şeker and Eryiğit (2017) extended their CRF based NER model which they previously proposed in (Şeker and Eryiğit, 2012). They extended the covered NE types and adapted the model for user-generated content. In their work, they also introduced three datasets by re-annotating the two existing datasets from (Tur et al., 2003; Çelikkaya et al., 2013), and ITU Web Treebank (IWT) (Pamay et al., 2015) respectively. Küçük et al. (2017) and Yeniterzi et al. (2018) provide a literature review on NER studies for the Turkish language. Okur et al. (2018) utilized a semi-supervised learning approach based on neural networks. In this work, they employed language-independent features and word embeddings to build a Turkish NER system for short texts.

It can be understood from the literature that the majority of existing NER studies focus on formal natural texts, but the number of studies focusing on informal OSN texts is rather limited. In the context of Turkish, studies over OSN data often focus on Twitter feeds, but we have not detected any previous study focusing on Facebook data. Therefore, in this thesis, we performed NER over Facebook data to fill this gap. To the best of our knowledge, this is the first try at extracting NEs from Turkish Facebook posts (see Section 5.1.2 for further details).

We would like to note that the primary reason to study SA and NER in this thesis is that different aspects can be extracted from the textual contents of users. As such, these tasks can be employed in many OSN related problems including privacy risk scoring of users.

2.3. Privacy and Attribute Inference Over OSNs

Conventional privacy-related studies mainly focus on protecting or measuring identities or private attributes of users (Wang et al., 2019). Besides, many privacy studies are focusing on the private attribute inference of OSN users. As such, in this section, we present our review of the literature with respect to both

attribute inference and privacy risk measuring (or scoring) in the following subsections respectively.

2.3.1. Privacy Risk Scoring

With the growing popularity of OSNs, users disclose their sensitive information consciously or unconsciously that puts them at privacy risks. Hence, several techniques and methods have been proposed for privacy risk evaluation. Most of the existing studies propose or use a PRE method, while the rest suggest a novel framework or approach for privacy evaluation of OSN users.

In the first group, one of the earlier and most popular works that depend on profile items is performed by Liu and Terzi (2010). This study proposes an IRT-based model to assess the privacy risk of OSN users. It also has been a guide for later studies (Domingo-Ferrer, 2010; Srivastava and Geethakumari, 2013; Minkus and Memon, 2014; Sramka, 2015; Symeonidis et al., 2015; Pensa and Di Blasi, 2016a; Pensa and Di Blasi, 2016b; Aghasian et al., 2017; Pensa and Di Blasi, 2017; Li et al., 2018; Pensa and Di Blasi, 2019) which propose or use a PRE method based on this work. Privacy functionality score (Domingo-Ferrer, 2010) is one of these studies and it measures privacy by dividing the amount of information a user can see about other users by the amount of information he/she shows about him/herself.

There are other methods or metrics (Braunstein et al., 2011; Gundecha et al., 2011; Nepali and Wang, 2013; Petkos et al., 2015; Vidyalakshmi et al., 2015b; Veiga and Eickhoff, 2016; Bioglio and Pensa, 2017; Alemany et al., 2018; Djoudi and Pujolle, 2019; Wang et al., 2019) as well, which have a focus on privacy evaluation in OSNs. For instance, Bioglio and Pensa adopted the susceptible-infectious-recovered model for modeling the spread of information in an OSN (2017).

In the second group, on the other hand, studies focus on building a new and complete framework or model for PRE and increasing privacy awareness in OSNs (Becker and Chen, 2009; Talukder et al., 2010; Caliskan Islam et al., 2014; Srivastava and Geethakumari, 2014; Vidyalakshmi et al., 2014; Caramujo and Da Silva, 2015; Vidyalakshmi et al., 2015a; Zeng et al., 2015; Biega et al., 2016; Oukemeni et al., 2018; Song et al., 2018; Al-Asmari and Saleh, 2019; Oukemeni et al., 2019). The majority of these studies suggest to use three types of information, but they often do not concern with techniques to be employed. For instance, SONET defines an attribute to an actor, attribute to attribute, and actor to actor relationships and contains a privacy index, namely PIDX, based on visibility and sensitivity of user attributes. Except those, there are also developed OSN applications (Akcora et al., 2012; Cetto et al., 2014; Bioglio et al., 2018) which are devoted to increasing privacy awareness of users. For instance, Friend Inspector is a serious game that allows its users to playfully increase privacy awareness on Facebook (Cetto et al., 2014).

In terms of the focus of the previous studies, it is clear that the majority of them consider privacy risk scoring of OSN users, while some of them take privacy in different aspects such as measuring information diffusion (Bioglio and Pensa, 2017; Alemany et al., 2018), privacy settings configuration (Srivastava and Geethakumari, 2014), modeling privacy attacks (Yang et al., 2012), and privacy-preserving friending (Akcora et al., 2012; Aghasian et al., 2018). Besides, most of the existing works consider users' points of view when measuring privacy risk, while a few are service-oriented (Vidyalakshmi et al., 2014) and consider third-party applications (Talukder et al., 2010, Symeonidis et al., 2015).

In terms of the type of information, most of the existing studies rely on structured data (i.e., profile items) and some other studies suggest using actions and content in scoring methods or models. However, content-based research just suggests using content but does not concern with technologies that can extract or infer attribute values from the unstructured data (Braunstein et al., 2011; Srivastava

and Geethakumari, 2013; Petkos et al., 2015; Al-Asmari and Saleh, 2019). Even there are some other content-based studies, they aim to detect sensitive information without privacy evaluation and scoring. As of writing this thesis, we only detected a few studies (Caliskan Islam et al., 2014; Biega et al., 2016; Song et al., 2018; Wang et al., 2019) that use contents in privacy evaluation of OSN users.

In this thesis, however, we use real-world OSN data for privacy scoring of users unlike the majority of existing studies in the literature. We perform privacy scoring of Turkish Facebook users using state-of-the-art methods in two different tasks. In our preliminary task, we perform a basic privacy scoring analysis over our smallest snapshot of crawled data. In our extended task, on the other hand, we perform privacy scoring of users on our largest snapshot of the crawled data. In this task, we inferred the private attributes of users and performed privacy risk scoring to show that users are at higher risk than they believe. We also showed that social tie strengths among users have to be taken into consideration when computing their network-aware privacy risk scores (see Section 5.2.3 for further details).

2.3.2. Attribute Inference

An adversary (i.e., user, third-party, or a boot, etc.) can access the OSN data and infer private information (e.g., gender, age, political view, etc.) with the help of statistical and ML techniques. As such, researchers have been performed many studies to infer the private single or multiple attributes of OSN users. However, in this thesis, we performed kinship and gender inference tasks. We also included the user identification task in this section as it aims to infer private OSN accounts. Therefore, in this section, we present our literature review that is just focused on gender detection, kinship detection, and user matching studies. The published studies on these research topics are given in the following sub-headings.

(1) Gender Detection: In literature, this type of research has been done many times before to detect the gender of the writer/author of the textual content.

Here, however, we focus gender detection studies over OSNs from which some studies use text messages of users, while some of them use profile information (Tuna et al., 2019), and images (Corriga et al., 2018; Han et al., 2018). The published works on this topic are as follows: Keeshin et al. (2010) performed gender detection on Facebook status updates using different features based on words, counts, and structures. The authors report that they achieved an accuracy of 0.677 with the help of the NB classifier on a dataset that includes 170K status updates. Peersman et al. (2011) tried to infer age and gender in OSNs. They use a TC approach to detect the age and gender of users based on their chat texts which are collected from a Belgian OSN. In their work, the authors perform age and gender categorization together and report that they were able to obtain an accuracy of 0.66 for age classification including gender by using 50K most informative unigrams (Peersman et al., 2011).

Burger et al. (2011) performed gender detection on a large-scale tweet corpus (i.e., 4.1M tweets of 184K of users) which is crawled from Twitter. They have extracted 15.7M character and word level n-gram features from their corpus. It is worth noting that they use both profile information and tweets in their ML model. The authors report that they achieved an accuracy of 0.92 when using tweets together with profile information. Tang et al. (2011) use a display name centric approach to detect the gender of Facebook users. The authors created a dataset by collecting information of 1.2M users from the New York network with the help of a focused crawler. They achieved an accuracy of 0.952 by using an ML-based inference model which mainly uses the first name and other profile information of users.

Deitrick et al. (2012) extracted 9,170 features from 3,031 tweets to perform gender categorization of Twitter users. Using the MBW algorithm, they predicted gender with an accuracy of 0.985 by using the most informative 53 features. Ciot et al. (2013) performed gender inference of Twitter users in a non-English context. They crawled messages posted in four different languages to

create their datasets. The authors report that they achieved the best accuracy as 0.87 for Turkish on a dataset that includes Turkish tweets of 3.6K users. Despite the huge number of features that make gender classification computationally expensive, Alowibdi et al. (2013a) use only 5 features to detect the gender of Twitter users. Using these color-based and language-independent features, the authors obtained the best accuracy of 0.718. In their other study, Alowibdi et al. (2013b) also explored profile characteristics for gender classification on Twitter. The authors proposed a technique that uses phonemes to extract and reduce features. They state that the highest accuracy was obtained with an accuracy of 0.82 by using 16K phoneme-based trigram features. Schwartz et al. (2013) use a lexicon-based learning approach for gender inference of Facebook users and the authors report that they obtained an accuracy of 0.919 using their approach.

Sap et al. (2014) created a lexicon from a corpus of 300M words written by 75K users to predict the age and gender of Facebook users. Rangel and Rosso (2013) used some style features to predict the gender of Facebook users. The authors obtained an accuracy of 0.59 on a dataset that includes 1.2K comments written in Spanish. Fatima et al. (2017) performed gender detection for author profiling on Facebook. The authors create their dataset by asking selected Facebook users to provide their demographic information along with their comments and posts. Based on their results, they achieved the best accuracy as 0.875 on a multilingual corpus that is comprised of posts and comments of users.

Tellez et al. (2018) performed gender classification based on both texts and images posted by Twitter users. The authors report that they achieved the best F1 score for English as 0.826 when they solely use the TC approach. Giannakopoulos et al. (2018) performed gender inference over a Twitter dataset and obtained an accuracy of 0.872 as the best result by using color, image, and display name fields from user profiles. Nicvist et al. (2018) performed gender detection using a simple classifier which obtained an accuracy of 0.61 on a dataset of 456 tweets. Bassem and Zrigui (2018) performed a comparative evaluation by using DL models (i.e.,

GRU, LSTM, and CNN) with word embeddings for the gender detection of Arabic Twitter users. They obtained the best result with an accuracy of 0.796 with the help of the GRU model. Han et al. (2018) performed age and gender classification of Instagram users based on activities, image objects, and tags. They obtained an accuracy of 0.82 for gender detection by using a multi-layer perceptron algorithm. Corriga et al. (2018) performed gender and emotion detection tasks for Facebook users from the University of Cagliari. They used the BoW of textual tags which are extracted from 3K images of the users. The authors obtained an accuracy of 0.63 as their best value for the gender prediction task. Khandelwal (2019) identified both humor and gender using tweets, first name, and profile picture profile owner. The author obtained the best result with an accuracy of 0.805 with the help of the SVM classifier.

To provide an overview, the majority of gender detection studies have focused on Twitter users and little has been done on non-English context. One of the earliest gender detection studies in Turkish is performed by Amasyalı and Diri (2006) that focuses on author profiling over news texts. There also exist only a few other studies with the aim of gender prediction on chat messages (Kucukyilmaz et al., 2006), tweets (Sezer et al., 2019a; Sezer et al., 2019b), and Facebook comments (Talebi and Köse, 2013; Çelik and Aslan, 2019).

In this thesis, we perform gender detection for Turkish Facebook users. To the best of our knowledge, our gender prediction task is the first to try to predict the gender of Turkish Facebook users by using both language dependent and independent features. What is more, our study provides the state-of-the-art result in the context of gender prediction of Turkish Facebook users (see Section 5.2.2.2 for further details).

(2) Kinship Detection: Kinship detection which is also known as kinship verification is an attractive task for the research community. This task is mostly taken as a computer vision task and handled via facial image analysis of people (Liang et al., 2019). As a result of the recent explosion in usage of OSNs, this well-

studied problem is becoming increasingly important and it is also explored in the context of OSNs to detect social relationships among users (Wang et al., 2010; Shao et al., 2014). In the context of OSNs, this problem can be considered as a user attribute inference task, and the published works which are related to this topic as follows: Fang et al. (2015) performed a relational user attribute inference task by using lexical and visual features that are extracted from user-generated contents. Unlike existing studies, the authors do not ignore the dependency between attributes and investigate six types of user attributes including relationship, gender, age, and so on. They collected a dataset from Google+ by using its public API and their preprocessing has resulted in 2,543 users and 846K posts. The authors manually labeled users with respect to their attributes and used different models to extract both visual and textual features. The authors conducted extensive experiments and obtained an accuracy of 0.62 with respect to the relationship attribute. Notice that this study only considers the private relationship status of users and aims to infer whether they are married or single.

Liang et al. performed kinship determination over China's mobile social networks and used SMS text data, phone contacts, and call data records (Liang et al., 2018). The authors selected the most discriminative 75 features from their dataset which contains 1.37M kinship relations and obtained an F1-score of 0.78 by using the XGBoost algorithm. They only consider whether there exists a kinship relation between users and do not care about the specific type of kinship. Umair et al. obtained forensic artifacts of Facebook messenger users with the help of both an Android and an IOS device (Umair et al., 2018). The authors created a dataset including 199 instances to train three different classifiers that are built-in Weka software. This is done by using message exchange patterns and Facebook contact information of the profile owner to classify the device owner's contact classification into weak, medium, and strong. The authors do not consider the message contents and they conclude that their results may be biased or over-fitted, but a Random Tree classifier achieves an F1-score of 0.99 on the supplied test set.

There also exist other studies that aim to predict the strength of relationship/tie/link between users in context of OSNs (Gilbert and Karahalios, 2009; Xiang et al., 2010; Zhao et al., 2012; Xu et al. 2019). Such studies are mostly interested in detecting the strength of an interaction between users or detecting popular friends (Jiang et al., 2012) by using profile similarity, interaction activity, and so on.

However, there is not any previous study exploring fine-grained kinship relations between OSN users based on user-generated contents. As such, in this thesis, we perform a kinship detection task that aims to fill this gap and it does not only consider the existence of kinship, but it is also interested in the type of the kinship relation. It seems that our kinship detection task is the first study for fine-grained kinship relation detection with respect to the OSNs.

(3) User Matching: User matching (i.e., user identification) across different sites can be useful for many purposes. As such, it has attracted the interest of the research community. The published studies on this topic are as follows: Vosecky et al. (2009) developed a profile comparison tool to identify users across OSNs. This tool decides whether two accounts match by calculating the similarity of their vectorized profile attributes. The authors evaluated this tool on a dataset that contains a small number of positive pairs and measured the performance (i.e., an accuracy of 0.83) as the number of correct predictions. Raad et al. (2010) proposed a user matching framework that employs profile attributes of users. This framework makes it possible to assign different similarity measures for attributes. The authors evaluated this framework on randomly generated OSN profiles and detected that this approach outperforms classical methods.

Malhotra et al. (2012) extracted online digital footprints (i.e., aggregated information from different OSN accounts that belong to the same user) of users across OSNs by using public profile attributes. The authors discovered that username and display name are the most discriminative elements for user identification. Peled et al. (2013) used both profile and network structure-based

features to match user profiles with the help of binary classification. The authors obtained a 0.98 AUC value on a dataset that includes 158 positive and 306 negative pairs from two different OSNs. Zafarani and Liu (2013) extracted different features from usernames that are employed for matching users. The authors achieved an accuracy of 0.983 by using their methodology namely, MOBIUS, that incorporates binary classification.

An extended stable matching method based on profile and content attributes is proposed by Liu et al. (2014). The authors detected that this method identifies up to 70% of user accounts. An iterative user identification algorithm is proposed by Bennacer et al. (2014). This method uses both public profile and network structure information to iteratively match profiles. The authors report that this algorithm obtains a high precision of 0.94 on a quite large dataset. Zhou et al. (2015) proposed an algorithm, namely, FRUI, to calculate the degree for all candidate matched pairs. The authors consider top-ranked pairs as identical and demonstrate that their algorithm outperforms other network structure-based methods. Goga et al. (2015) defined a set of properties, namely, ACID, to evaluate matching tasks depending on profile items in a reliable way. The authors claimed that accuracies reported in previous studies are lower than in practice.

Wang et al. (2016) performed user matching just based on the usernames of users. The authors use a similarity score which is obtained on self-information vectors to decide whether two usernames belong to the same user. Another ML-based framework, namely, MUSIC, is proposed by Sha et al. (2016). The authors use both word2vec and doc2vec models to represent user's contents such as posts, messages, and so on. This framework achieves an F1-score of 0.893 just by using feature vectors extracted from user-generated contents. Li et al. (2017) considered the user matching task as a binary classification and obtained the best F1-score of 0.962 by using attributes extracted from only display names of users. Fang et al. (2017) proposed a co-training approach to speed-up the user matching process. In this approach, profile and network structure information are used and it is detected

that co-training performs better than the profile and relation information models. Hazimeh et al. (2017) proposed an approach, namely, SocialMatching++ that uses user events and biographies to enhance attribute approaches in user matching.

Li et al. (2018) performed user identification across three OSNs by using solely display names of users. The authors obtained a 0.9 AUC value on their datasets and found that more than 45% of users prefer to use the same display name on different OSNs. Li et al. (2019) proposed a user identification framework that achieves an F1 score reaching 0.9+ by using features extracted from display names and usernames. In this thesis, we also perform user matching based on several features that are extracted from usernames.

According to literature, privacy, security, attribute inference, and NLP-based tasks are well studied especially in the context of the English language and users from other nations. However, in these fields, the number of studies considering Turkish users and their textual contents are very scarce. The majority of NLP based tasks over OSN for Turkish also mainly focus on Twitter OSN. Therefore, in this thesis, we perform a comprehensive analysis of OSN data with respect to Turkish Facebook users by considering different aspects such as SA, NER, attribute inference, and privacy scoring. Some of these tasks are the first of their kind and obtained state-of-the-art results which are presented in Section 5.

3. MATERIALS

This section gives details about materials used in this thesis including used software tools, crawled public Facebook data, and other task-dependent datasets.

3.1. Used Software Tools and Services

In this thesis, various software tools are used to build, customize, and design our methods which are described in the next section. These tools are as follows:

- Keras: It is a high-level NN API that is written in Python programming language and able to run on top of well-known ML frameworks such as Tensorflow and Theano (Chollet, 2015; Abadi et al., 2016; Team et al., 2016). Keras allows for easy prototyping of NN algorithms and runs seamlessly on both CPU and GPU.
- Tensorflow: It is an open-source and free tool developed by the Google Brain team for numerical computation and large-scale ML and DL research. Tensorflow can run on multiple CPUs and GPUs and has several wrappers in different programming languages like Python, C++, and Java. At a high level, Tensorflow is a Python library that allows making computations as a graph of data flows where nodes represent mathematical operations and edges represent carried data or tensors which are multi-dimensional arrays (Abadi et al., 2016).
- Weka: Waikato Environment for Knowledge Analysis (Weka in short) is written in Java programming language and provides collections of various algorithms in the context of preprocessing, classification, clustering, filtering, and so on. Researchers and practitioners can employ it both by including it in their own code and by using its user interfaces (Witten et al., 2005; Hall et al., 2009).

- Gephi: Gephi is open-source software for graph and network analysis. It is written in Java programming language and has many applications including exploratory data analysis, link analysis, social network analysis, and biological network analysis (Batian et al., 2009).
- Mallet: It is an open-source and Java-based package for statistical natural language processing, text categorization, clustering, topic modeling, and other ML applications to text. Mallet also includes various algorithms like CRF for sequence tagging for applications such as NER (McCallum, 2002).
- Selenium: It is an automated testing tool that is used to test web applications across different platforms (Patil and Temkar, 2017). It makes it possible for developers to emulate a user's interaction with the browser such as entering text into fields, selecting drop-down values, clicking links, and so on. It also provides other various controls such as mouse movement, and JavaScript execution (Anonymous, 2019a). As such, Selenium may be used for any task which requires automated interaction with a web browser and can help to replicate and store human actions on a web page (Flores et al., 2013). In this thesis, we used the Java-based Firefox Driver to employ the Selenium.
- Scikit-learn: It is an open-source ML library that is written in Python programming language and supports supervised and unsupervised learning. Scikit-learn provides dozens of built-in ML algorithms in the context of classification, regression, clustering, dimensionality reduction, and so on (Pedregosa et al., 2011).
- NetworkX: It is an open-source Python package for creating, manipulating, and studying the structure, dynamics, and functions of complex networks (Hagberg et al., 2008). It supports various data structures of graphs and provides different graph algorithms and analysis measures.

- MySQL Workbench: MYSQL is one of the relational DBMSs that allows to store data in multiple tables with relationships and supports multiple storage engines like InnoDB. MySQL Workbench is a unified visual database designing and modeling tool for MYSQL that provides data modeling, SQL development, and server administration.
- Zemberek: It is an open-source Java-based NLP tool for Turkish (Akın and Akın, 2007). The current version of the Zemberek provides various modules to perform NLP related tasks such as language identification, morphological analysis, language modeling, and so on.
- Gensim: It is a Python-based, platform-independent, and open-source tool that provides various algorithms for building and analyzing models that take text data as input (Rehurek and Sojka, 2011).
- DL4J: DJ4J (aka. Deeplearning4j) is a Java-based, open-source, and distributed DL framework which works with Spark and Hadoop on top of distributed CPUs or GPUs. It hosts its own C++ based scientific computing library that claims at least 2x more speed-up over Python's Numpy package (Fox et. Al, 2016). The algorithmic core of the DL4J enables scientists and developers to build deep neural networks on a high-level using concepts such as layer.
- Google Colab: It is a research project for prototyping ML and DL models on powerful hardware options like GPUs and TPUs (Bisong, 2019). Google Colab (i.e., Colaboratory) is free to use and provides a free Tesla K80 GPU of about 12GB. Anyone can run a session for 12 hours due to there might be chances of people using it for the wrong purposes (Ponnappa, 2019). When the 12 hours have passed, the session must be restarted.
- NLTK: It is a suite of Python libraries and algorithms for symbolic and statistical NLP tasks (Loper and Bird, 2002). It allows any programmer,

even a beginner, to get acquainted with NLP tasks easily without spending too much time to study or gather resources (Lobur et al., 2011). NLTK is widely used in linguistics, artificial intelligence, and ML projects.

- Skopt: Scikit-Optimize (a.k.a skopt) is a simple and efficient software tool intended to minimize expensive and noisy black-box functions. It implements several methods for sequential model-based optimization and aims to be accessible and easy to use in many contexts (Louppe 2017).
- Imbalanced-learn: It is an open-source python toolbox that is aiming at providing a wide range of methods to cope with the problem of learning over an imbalanced dataset which is frequently encountered in ML tasks (Lemaitre et al., 2017).

3.2. Crawled Facebook Data

Using our crawler (see Section 4.2), public data of Facebook users from Turkey is crawled. During the crawling process, three different snapshots are taken which include data of 5K, 10K, and 20K users respectively. Crawled data only includes textual contents available on users' profiles and walls (i.e., timeline). Thus, our crawled public Facebook data contains profile items and wall activities (i.e., posts, comments, and replies) of each user in our database. Table A.1 and Table A.2 present network metrics and properties and user interactions in three different snapshots of the crawled data. Notice that we use the Gephi package to perform network analyses of the crawled data and give its detailed statistical description with respect to the snapshot size in Table A.1.

3.3. Datasets

As part of the materials used in this thesis, different datasets are created from the crawled data. Following sub-headings give details of these datasets which are called with the domain names in which they are used.

3.3.1. Super Set of Activities

This is a set of all activities within the largest snapshot (i.e., 20K) of the crawled data. This superset of activities (SSA) contains user-generated texts along with their meta-data information. As seen from Table A.2, this set contains 2.76M posts, 2.74M comments, and 466K replies of users. We named this data as SSA due to some of our datasets are created by taking its subsets.

3.3.2. Raw Text Datasets

This section describes the raw text datasets used in this thesis. These datasets are task-dependent and some of them are taken from previous studies, while a few of them are created by taking subsets of the SSA data.

3.3.2.1. Word Embedding Data

We use the mentioned data to learn distributed representations of words. As such, we used all of the textual activities from the SSA data to learn distributed representation of the data with different layer sizes. Notice that only activities including user-generated texts are used to form this data which includes 3.95M activities (i.e., 868,603 posts, 2,646,186 comments, and 442,788 replies) in total. We use learned word embeddings to feed ML and DL models in our tasks.

3.3.2.2. Hybrid Dataset

As its name implies this is a hybrid dataset that includes unannotated 85K sentences (or sequences) with 2M tokens from METU Turkish Corpus (Say et al., 2002) and randomly selected 100K Facebook posts with 2M tokens from the SSA dataset. We use this dataset to train the word2vec model and obtain distributed representations of tokens in our NER task (see Section 5.1.2).

Figure 3.1. An example categorization of a Facebook post and its relative comments and replies.

3.3.2.3. Wall Datasets

Wall datasets, namely, WD1, WD2, and WD3 are created by taking activities of users who make their gender attributes public in the largest snapshot of the crawled data. Considering publicity of gender and wall page of users and

applying a filter that eliminates users whose wall pages are empty or private; a final user set is created for gender inference task (see Section 5.2.2.2). The wall datasets include activities of 8,422 users from which 4,211 are males and 4,211 are females. As depicted in Figure 3.1, wall datasets are created by clustering activities found in WO's wall depending on the owner and target user of activities. The description of the three wall datasets are as follows:

- WD1: This set contains all activities that targeting the WO for each user in our final set. For instance, C1 and R4 are included in the WD1 set as they are directly targeting the WO (i.e., *Müge*). Notice that any activity is included in this set only if it is a direct post or the first relative comment or reply to any activity of WO.
- WD2: This set contains all activity contents posted by WO. For instance, P1, R1, R2, and R3 are posted by *Müge* who is WO in the depicted case in Figure 3.1.
- WD3: This set does not require any criteria and contains all activities found in the WD1 and WD2 datasets. In other words, it is the union of the WD1 and WD2 datasets.

Table 3.1. Quantitative description of the wall datasets

Attribute/Dataset	**WD1**		**WD2**		**WD3**	
	Male	Female	Male	Female	Male	Female
Posts	22,956	15,236	276,176	156,846	299,132	172,082
Comments	189,574	126,675	161,061	124,663	350,635	251,338
Replies	4,541	2,658	118,221	81,351	122,762	125,420
All Activities (AA)	217K	144K	555K	362K	772K	506K
Total tokens in AA	1.37M	813K	6.27M	3.05M	7.64M	3.81M
Average token	6.35	5.64	11.3	8.3	9.9	6.08

Notice that in these datasets each document sample represents all concatenated activities in related type which are found in WO's wall. Table 3.1 presents a quantitative description of the wall datasets and shows that the size of

the datasets varies in the range of 361K-1.27M activities depending on the type of the dataset.

As seen from Table 3.1, male users have more activities than females and their activities include more words on average than those of females.

3.3.2.4. SA Datasets

These datasets are created by taking two snapshots of the SSA data which only includes user-generated texts of Facebook users. The first one is model selection data which is used to explore which learning method and feature set produce better results. The second one is used to evaluate the sentiment of Facebook activities by using the detected best learning model.

(1) Model Selection Data: To create this type of data, we queried our database to randomly fetch activities that include user-generated contents. We then applied the following criteria to eliminate unrelated messages:

- The message must be written in Turkish. Notice that in some cases users may post a message in other languages especially in English.
- The message must include at least one term except for emoticons and other meaningless information. This criterion ensures that the message does not consist of only punctuation, emoticon, or anything else.

Applying these criteria in the querying phase has resulted in 146.4K messages in total. However, we randomly selected and manually annotated a subset of these messages which is an evenly distributed corpus in positive and negative categories and has 12,660 samples in total.

(2) Evaluation Data: This data set is used to mine relationships between attributes (i.e., age and gender) of users and meta-data (e.g., reactions and time of posting, etc.) of their activities. Table 3.2. presents a quantitative description of our evaluation data. As seen from Table 3.2, this dataset includes 72,274 posts, 46,930

comments, and 25,193 replies of 754 users who disclose their both gender and age attributes in 20K.

Table 3.2. The number of users and their activities with respect to the age-range and gender attributes in SA evaluation data

Gender	Age-range					Total
	11-20	21-30	31-40	41-50	51-60	
			# of users			
Male	10	98	221	94	41	464
Female	4	100	136	36	14	290
Total	14	198	357	130	55	754
			# of posts			
Male	2,049	13,484	22,231	12,480	2,945	53,189
Female	28	3,378	11,300	3,648	731	19,085
Total	2,077	16,862	33,531	16,128	3,676	72,274
			# of comments			
Male	384	7,453	17,717	6,189	1,879	33,622
Female	8	2,094	8,671	2,184	351	13,308
Total	392	9,547	26,388	8,373	2,230	46,930
			# of replies			
Male	187	2,279	8,999	4,540	2,337	18,342
Female	8	933	3,781	1,989	140	6,851
Total	195	3,212	12,780	6,529	2,477	25,193

3.3.3. NER Datasets

(1) Datasets: In this thesis, different datasets from previous studies are used to measure the efficiency of the developed NER system. A new dataset, namely, FBNER (Facebook Named Entity Recognition) dataset is also introduced which is comprised of posts from the crawled Facebook data. In this thesis, our NER datasets are as follows:

- TWT1: It is comprised of manually annotated Tweets (Eken and Tantug, 2015) and consists of all seven entity types. This dataset is not well structured (e.g., some of the sequences spread over multiple lines) and not suitable to automatically process. As such, some manual corrections are made on this dataset without changing the number of named entities and sequences.
- TWT2: It is a re-annotated version (i.e., TDS1_v4) of the dataset introduced by Çelikkaya et al. (2013) and includes manually labeled Tweets in seven entity types (Şeker and Eryiğit, 2017).
- Train7 and WFS7 are comprised of news articles texts and seven entity types (Şeker and Eryiğit, 2017).
- IWT: It is a corpus, namely, ITU Web Treebank, that is compiled from Turkish texts found on the web (Pamay et al., 2015; Şeker and Eryiğit, 2017).
- FBNER: It is our manually annotated NER dataset which is comprised of automatically collected 5K Facebook posts (i.e., sequences) with 77K tokens in total. This dataset has seven entity types and its two subsets are also created by taking 70% of sequences as FBTRN and the rest as FBTST.

Table 3.3. A quantitative description of the NER datasets.

Dataset	Total # of								
	Seq.	Token	ENAMEX			TIMEX		NUMEX	
			PER	LOC	ORG	DATE	TIME	MONEY	PERCENT
TWT1	9,355	108,712	1,837	1,112	1,541	220	77	82	60
TWT2	5,040	46,382	4,258	215	399	56	20	12	3
Train7	24,819	444,843	14,693	9,767	9,160	1,366	148	648	597
WFS7	1,572	24,755	658	637	410	119	19	41	62
IWT	5,011	43,336	380	260	401	59	9	45	8
FBNER	5,000	77,580	1,711	1,522	1,183	355	42	35	24
FBTRN	3,500	59,844	1,323	1,245	1,000	278	34	20	17
FBTST	1,500	17,736	388	277	183	77	8	15	7

As some of the previous datasets have less NEs than the FBNER, NWSA and TWTA are created by combining news (Train7, WFS7, and IWT) and tweets (TWT1 and TWT2) datasets respectively to use in our experiments. Table 3.3. summarizes the quantitative description of the datasets described above concerning the number of sequences, tokens, and NEs. Note that we use exact matching while counting the number of NEs. For instance, if a named entity with type PERSON in a sequence is *"Mustafa Kemal"* then we increase the related counter by 1, not 2.

(2) Data Format: NER systems often use CoNLL data format (Sang and De Meulder, 2003) in which data file contains each token (word) per line with empty lines representing sequence boundaries. The last token in each line represents the NE type for the related token. As not all NEs are composed of a single token, there are different models to represent multi-word NEs. Following list contains commonly used prefixes in these models (Konkol and Konopik, 2015):

- B (Beginning) represents the first word of NE,
- I (Inside) represents the part (inside word) of NE,
- L (Last or End) represents the last word of NE,
- O (Outside) represents a word that is not a part of NE,
- U (Unit) represents the single word NEs.

We would like to note that there is also a raw tagging model in which tags are used without any prefix or position information.

Table 3.4. Data format for three different NE tagging models

Token	**Tagging Model**		
	Raw	**IOB2**	**BILOU**
Mustafa	PERSON	B-PERSON	B-PERSON
Kemal	PERSON	I-PERSON	I-PERSON
Atatürk	PERSON	I-PERSON	L-PERSON
,	O	O	O
Mersin	LOCATION	B-LOCATION	U-LOCATION
gezisindeyken	O	O	O
…	O	O	O

An example data format for an arbitrary sequence *"Mustafa Kemal Atatürk, Mersin gezisindeyken… (while Mustafa Kemal Atatürk is on a trip to Mersin…)"* is given in Table 3.4. In this thesis, we use the IOB2 (or BIO2) model that just uses the B (only for each first word of an entity), I, and O prefixes to distinguish NE types.

3.3.4. Username Dataset

As seen from Table A.15, 3.7K of 20K users reveal their connections in their Facebook profiles. Distribution of revealed links/accounts by these users are as follows; 2,836 are Facebook accounts, 105 are Twitter accounts, 295 are Instagram accounts, 9 are LinkedIn accounts, 9 are Myspace accounts, and 509 are other links that are not related to any OSN. Notice that other links do not refer to any OSN, while Facebook links refer to already discovered accounts. In other words, the wall owner shares his/her own Facebook account link in the connections section.

Excluding such Facebook links and other links, we build our positive and negative instances that include ground-truth pairs of usernames that refer to different OSNs including Facebook-Twitter, Facebook-Instagram, Facebook-LinkedIn, and Facebook-Myspace. As stated above, among the 20K users in our Facebook data, only 3.7K of users disclose at least one OSN account (or connection). The distribution of these connections shows that, quite interestingly, a large majority (~75%) of users share their Facebook account that they already reside. Only 418 of all 3.7K users share connections to other OSN services. Additionally, we detected that 94 of these 418 connections are completely comprised of a numeric string username (e.g., *"12******9"*) that is possibly assigned by the social site automatically. In this situation, username-based matching methods fail to work properly, due to such usernames contain almost no information redundancies (Li et al., 2017; Li et al., 2018).

As such, we excluded such type of numeric usernames from that of the 418 usernames to make our methods more suitable. This elimination process has resulted in a total of 324 connections that form the base of our training/test dataset. Consequently, the training/test dataset is comprised of pairs of OSN usernames alongside the binary class label. If two usernames belong to the same user, we say that the corresponding instance is positive.

We sampled positive instances using the connection shares in our Facebook dataset. Table 3.5 provides some positive samples for reference purposes.

Table 3.5. A sample list of positive username pairs.

User No	**Usernames on** OSN 1	OSN 2
1	murat.d****n.19**	mrt_d****n
2	lutfu.b****z	lutfub****z
3	kemal.t****n.1*	t****n.kemal
4	mye.t****n	t****necrin
5	e****.sahin.7*2	e****.sahin.7*2

As seen from Table 3.5, some users prefer to use the same usernames across distinct OSNs, where others (possibly due to unavailability of the same username in the other OSN) make very simple changes in their original usernames such as adding prefixes or suffixes, making abbreviations, and so on. We would like to note that some letters of the usernames listed in Table 3.5 have been masked with asterisks to respect the privacy of corresponding users.

Negative instances were fabricated using a fail-safe strategy: we randomly picked two users of our dataset, such that at least one of which shared at least one connection. Let these users be denoted with u and v respectively. We assumed that u and v belong to different individuals. Based on this assumption, we randomly paired a username from u's known OSN accounts (except for Facebook) with a username from v's known OSN account(s). We consider this strategy to be fail-safe because at least one of the u and v uses the "connections" feature of Facebook

and neither account lists the other as belonging to the same user if both use the "connections" feature.

This process yielded a total of 324 real-world positive instances. To ensure a balanced distribution of positive and negative instances, we also created 324 negative instances, which means that our username dataset is comprised of 648 real-world instances in total.

3.3.5. Kinship Datasets

These datasets are created from our largest snapshot of crawled Facebook data (i.e., 20K) to perform fine-grained kinship detection based on wall activity contents (see Section 5.2.2.1). In our largest snapshot (i.e., 20K), 3,378 of users disclose their 7,581 family memberships, whereas 4,660 users share their 4,660 private relationships (see Table A.2). Notice that Facebook users can reveal one or more family members, whereas they can disclose only one private relationship in their profiles (see Figure 1.5). We take these 12,241 relationships in total to create our candidate user set. However, we had to exclude some of the relationships from this set of users due to the following reasons:

- As seen from Figure 1.5, revealed relational accounts may be in a passive state which means that related account is inaccessible for any reason. In that case, it is not possible to view/access the related user's activities. Therefore, if one of two users in a kinship pair is passive it is excluded from the base set.
- We excluded some private relationships (e.g., in a relationship, single, widowed, and divorced) which does not provide information about the relational user. In such a disclosure, the related user only gives information about his/her relationship status, he/she does not share with whom one is in this relation.

- We removed kinships with the type of family member *("akraba")*, as we are unable to know the degree and type of the kinship in that case.
- We only keep private (i.e., special) relationships which are disclosed as marriage and determined who is being a wife and who is being a husband in related kinship pair based on the users' genders. Private relationships with the type of marriage included in the general type of spouse.

After the elimination process, we grouped kinships with the same meaning into more general types and obtained our candidate set of users with their relationships. For instance, we grouped relationships with the type of husband, wife, partner, and married into spouse type which is a general type covering all of these types. Besides, we expanded the candidate set by performing relationship inference which is exemplified in Figure 5.24. Note that this simple inference task relies on implicit friendship relations (see Figure 1.3) of Facebook's nature. Table 3.6 presents the distribution of revealed (R), inferred (I), and combined (RI) relations in our candidate set.

Table 3.6. The number of generalized kinship types for revealed and inferred relations in the candidate set of users.

Kinship	Generalized Kinship Types								
	Parent	Child	Sibling	Spouse	Cousin	Grand-child	Sibling of Parent	Grandparent	Child of Sibling
R	291	430	1,682	1,502	1,300	27	376	21	603
I	9	4	1,104	9	1,141	39	434	3	260
RI	300	434	2,786	1,511	2,441	66	810	24	863

We would like to note that even it is possible to perform inference over inferred relations as well, we perform inference just one time to avoid having biased pair of kinship users. In the largest snapshot of the crawled data, only 261 users' first neighborhoods are visited (see Table A.2) by our crawler (Coban et al., 2020a). This means that majority of discovered users including users in the candidate set are still not visited. Therefore, we performed an additional focused

crawling with the help of our crawler to collect the wall activities of the users in the candidate set. After completion of focused crawling, we again, filtered out unfeasible pairs from the candidate set due to the following reasons:

- Both users in a kinship pair may keep their walls private or there may not any activity in their walls.
- Even both users in a pair keep their walls public, it may not be possible to observe any kinship words in their activity contents.

After applying this second filtering, we obtained our final set of users with their kinships. Afterward, we created different datasets based on two different approaches, namely, single and mutual. In the single approach, we take each relationship in one direction and add its symmetric relation into the dataset. In the mutual approach, on the other hand, we include relation as mutual instead of its two sides. For instance, when we consider Figure 1.5 again, we include two samples in child and parent categories for the relation between *Yaren* and *Senem*. However, in the mutual model, we only include one sample/instance in the child-parent category for this relation.

Table 3.7. Class distribution for kinship datasets created with the single approach

Data Set	**Category (Code)**								
	Parent (S1)	Child (S2)	Sibling (S3)	Spouse (S4)	Cousin (S5)	Grandchild (S6)	Sibling of Parent (S7)	Grand parent (S8)	Child of Sibling (S9)
S_R	545	545	2,596	138	2,116	41	777	41	777
S_{RI}	732	732	4,174	152	3,776	67	1,245	67	1,245

From this point of view, we created four different data sets and named each of them with a pattern such that a single letter S or M indicates whether the dataset contains single of mutual relations, whereas a subscript text R or RI next to the letter shows that the dataset includes revealed or sum of revealed and inferred relations. For instance, S_R shows that the dataset includes two sides of revealed relations.

Table 3.8. Class distribution for kinship datasets created with the mutual approach

Data Set	Category (Code)					
	Cousin (M1)	Spouse (M2)	Sibling (M3)	Sibling of Parent-Child of Sibling (M4)	Parent-Child (M5)	Grandparent-Grandchild (M6)
M_R	1,058	69	1,298	777	545	41
M_{RI}	1,058	76	2,087	1,245	732	67

Table 3.7 and Table 3.8 present a detailed description of our datasets created with single and mutual approaches respectively. As seen from the tables, the number of classes varies depending on the used approach and our kinship datasets are imbalanced and have highly skewed class distribution.

3.4. Embedding Models

An embedding model is created to obtain a distributed representation of data by training a paragraph or word embedding model on an unlabeled and large corpus. The word embedding model is proposed to encode each word/term as a low dimensional, continuous, and dense vector. It is typically induced using unlabeled data in an unsupervised way (Zhang et al., 2016). This method uses a three-layers neural network with two different structures to learn distributed representation of words (Mikolov et al., 2013a; Mikolov et al., 2013b).

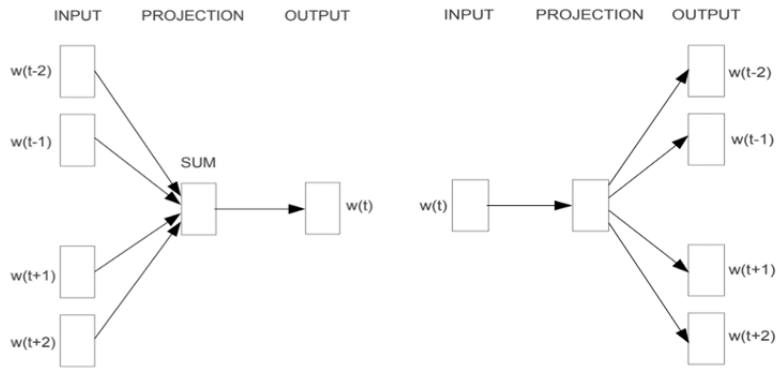


Figure 3.2. CBOW and skip-gram architectures for word2vec model

Figure 3.2 depicts these two architectures which are named as continuous bag-of-words (cbow) and skip-gram (Mikolov et al., 2013a). As seen from Figure 3.2, cbow predicts the current word given its context, while skip-gram predicts the surrounded context given the current word. In this thesis, we use word embeddings in different tasks with different manners in both ML and DL models. As such, we use different word embedding models that are trained on Turkish texts. These models which we call with *"WE"* prefix to imply word embeddings are as follows:

- WE_{TR}: This is a pre-trained model for the Turkish language from Wikipedia dump and publicly available on Github (Köksal, 2018). It was trained using cbow architecture with layer size 400, window size 5, and the minimum number of counts 5. It is created by using the gensim (Rehurek and Sojka, 2011) package with a basic preprocessing that does not include any semantic analysis.
- WE_{Fast}: It is created to distribute pre-trained word vector models for 157 languages by *fastText* software (Grave et al., 2018). In this thesis, we select and use the one related to the Turkish language among these models which were trained on Common Crawl and Wikipedia data using cbow architecture with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5, and 10 negatives. An important detail to note here is that each word vector is obtained by taking the sum of the vectors of the character n-grams appearing in the word. The full word is also included in character n-grams.
- WE_{FB}: It implies that this model is trained on our word embedding data (see Section 3.3.2.1). In this thesis, we create different WE_{FB} models with different layer sizes using the gensim package with cbow, the window of size 5, and the minimum number of counts 1. Notice that these models are different from previous models described above as they are trained on Facebook data.

In some of our tasks, in this thesis, we also use paragraph vectors (or document vectors) which is an extension of word embeddings (Le and Mikolov, 2014). Unlike the word embeddings model, it learns to correlate labels with words, rather than words with other words. We train our paragraph vector (PV) models on the word embedding data to represent texts with document vectors. As in the WE_{FB} models, this training task is done for different layer sizes again by using the gensim package with dbow, window size 5, and the minimum number of counts 1. Notice that we call our PV models with a prefix *"PV"* and subscript word FB to imply that it is trained on Facebook data.

3.5. Lexicons and Gazetteers

This section gives details about lexicons and gazetteers used in this thesis.

3.5.1. Lexicons

In this thesis, we use the following task-dependent lexicons which are employed for different tasks and some of them are created by considering the related task, while some others are created by compiling or combining existing lexicons from the literature.

(1) Gender-Oriented Words: A lexicon that is comprised of lexically and semantically gender-oriented Turkish words. We take gender-oriented words characterized by Doğan (2011) along with their frequently used abbreviations and different surface forms on Facebook. This is because OSN users often use informal language and mostly write their messages without adhering to the grammatical rules. We also include mostly observed different variations (e.g., *"kardeş (brother)" → {"gardaş", "birader", "bilader"}*) of these formal words. Figure 3.3 depicts the word cloud of these formal words (different surface forms and variations are not included to reduce complexity) used in this thesis.

Figure 3.3. Word cloud for simple surface forms of gender-oriented Turkish words

Notice that blue and pink ones represent male and female-oriented words respectively.

(2) Non-Gender-Oriented Words: It is possible to use some words in a different sense in Turkish. This case is often observed for gender-oriented words as well. As such, we also created a small exceptional words lexicon that includes frequently used but non-gender-oriented words and phrases. We use this small lexicon to avoid extracting non-informative features. Notice that this small lexicon is created by only considering our observations and includes 30 words which are given in Table 3.9. Differentiating word sense is requiring an efficient WSD process, but we are trying to bring a simple solution by using this small lexicon as WSD is out of scope for this thesis.

Table 3.9. List of words in our non-gender-oriented words lexicon.

Single words	*analı, babalı, kocaman, kocayın, kocadı, çocuklar, çocuklu, çocukluğum, çocukluğun, çucukluk, evlatlar, kızlar, analar, babalar, çocuğa, babacan, annesiz, babasız, dedeler*
Multi-words	*hanım köylü, hanım efendi, kız kulesi, kız çocuğu, anneler günü, babalar günü, aşk hayatı, iş hayatı, okul hayatı, koca koca, analı babalı*

As can be seen from Table 3.9, this dictionary includes multi-word phrases including at least one gender-oriented word along with different surface forms of gender-oriented words. Note that these words are searched by using exact matching

criteria due to they can be used in a gender-oriented sense with some additional suffixes.

(3) Extended SentiTurkNet (eSTN): This lexicon is comprised of Turkish words with their sentiment polarity values. It is created by extending STN (SentiTurkNet) lexicon which is the first comprehensive polarity lexicon for Turkish (Dehkharghani et al., 2016). In the expansion process, Erşahin et al. (2019) searched all words in an automated synonym dictionary generation tool (Aktas et al., 2013) and if there is a match, the synonyms are added to the STN along with polarity values which are already present in the STN. They called this extended lexicon as eSTN that does not include objective and multiword terms (Erşahin et al., 2019).

(4) Seed Words: Our seed words lexicon is a combination of two lexicons created by Gezici and Yanıkoğlu (2018) and Kemik NLP group and it is comprised of 599 positive and 895 negative words. Notice that this lexicon also contains multi-words unlike the eSTN lexicon.

Table 3.10. An incomplete list of seed words

Positive Seed Words	**Negative Seed Words**
muhteşem *(magnificent)*	fiyasko *(failure)*
güzel *(beautiful)*	berbat *(awful)*
eğlenceli *(enjoyable)*	hayalkırıklığı *(disappointment)*
şahane *(fantastic)*	sıkıcı *(boring)*
kusursuz *(perfect)*	vasat *(mediocre)*

Seed words have been commonly used for SA studies. These types of words have some advantages such that they are not often used in the negated form and their sentiment polarity does not change much in different domains. Table 3.10 presents an incomplete list of seed words compiled by Gezici and Yanıkoğlu (2018).

(5) Informal Words: This small lexicon is created just based on our observations and includes 45 informal-formal word pairs such as *"gı → kız (girl)"*,

"dewrem → devrem (my friend)", *"bilader → birader (brother)"*, and *"gardaş → kardeş (brother)"*. We use this lexicon to correct informal words as possible as by replacing a matched informal word with its formal surface within the related pair.

(6) Kinship Words: As depicted in Figure 3.4, this lexicon includes kinship words used in Turkish. It is created by only considering some generalized kinship types such as cousin *("kuzen")*, spouse *("eş")*, sibling *("kardeş")*, child *("çocuk")*, and so on. To create this lexicon, we include formal and informal use of desired kinship words as Turkish Facebook users often write activities with grammatical errors. We would like to note that this lexicon does not just include words, but it also keeps the gender orientation of words and whether referred kinship type requires the same surname as well.

Figure 3.4. Word cloud of kinship lexicon.

As Turkish is an agglutinative language, it is possible to produce words from the same root with different surface forms. As such, the majority of the words in our lexicon were included without any inflections to increase the matching rate when searching by using regular expressions.

(7) Cities: This lexicon includes city and district names in Turkey. Each record in this lexicon stores information in which district belongs to which city.

(8) Schools and Universities: This lexicon contains the names of primary, secondary and high schools (Anonymous, 2016; Canaydın, 2017) in Turkey. It also includes university names in Turkey along with the names of cities they have been located (Sarhan, 2017).

(9) Political Pages: This lexicon includes a list of Facebook pages along with their supported political party. To create this lexicon, we first taken a subset of 459,335 Facebook pages liked by 13,023 of 20K users in the largest snapshot of the crawled data. We would like to note that our crawler does not crawl liked pages of users, but this additional data is crawled by Kılıç and Inan (2021) in their separate study that is aiming to propose a quantitative framework for privacy risk score evaluation.

Hereby, we could create a subset of liked pages that are created for political purposes. After creating our subset, we selected pages that their titles include any term related to six main political parties (i.e., JDP, NMP, RPP, HP, DPP, and GP) in Turkey. For instance, if the title of a page includes *"Devlet Bahçeli"*, we assigned the related page to the pages supporting the NMP party. Similarly, if the title of a page contains *"Kemal Kılıçdaroğlu"*, we assigned the related page to the pages supporting the RRP party. This process produced a list including 1,763 pages that each of which is associated with one of the six parties.

Table 3.11. An incomplete list of Facebook pages with respect to their associated political parties.

Party	Page Id	Page Title
JDP	*4***1*	*T C Devlet Başkanı Erdoğan*
	*5***7*	*Efsane Lider R T Erdoğan*
	*7***6*	*AkParti*
NMP	*7***3*	*Lider Devlet Bahçeli*
	*2****1*	*Ülkücü Hareket Engellenemez*
	*4***0*	*Milliyetçi Hareket Partisi MHP*
RPP	*4***9*	*Kemal Kılıçdaroğlu*
	*1*****7*	*Cumhuriyet Halk Partisi*
	*1*****4*	*Ne Mutlu CHP liyiz*
GP	*1**4*	*Meral Akşener*
	*1***6*	*Cumhurbaşkanı Adayım Meral Akşener*
	*2****5*	*İYİ Parti Gönüllüleri*
DPP	*5***3*	*Selahattin Demirtaş*
	*1*****4*	*Sırrı Süreyya ÖNDER HDP*
	*1*****2*	*HDP Lideri Selahattin Demirtaş*
HP	*2****8*	*Temel Karamollaoğlu*
	*1*****4*	*Saadet Partisi Milli Görüş*
	*9****4*	*Saadet Partisi Pendik Gençlik Kolları Resmi Sayfası*

Table 3.11 gives an example and an incomplete list of pages with their associated parties. We used this lexicon to infer the political views of users based on their liked pages (see Section 5.2.3.2.)

3.5.2. Gazetteers

The term *"gazetteer"* is used to represent a set of lists containing names of entities such as cities, organizations, days of the week, and so on. In this thesis, we used the following gazetteers:

- Organization gazetteer includes generator words (e.g., *"müdürlüğü"*), the names of football clubs in Turkey, and abbreviated names (e.g., *"tbmm"*, *"yök"*, *"tmmob"* etc.) of organizations.
- Location gazetteer contains country names together with province and district names in Turkey.

- Person gazetteer includes person names introduced by Eken and Tantug (2015) and revised in this thesis.
- Month and currency gazetteers which are introduced by Seker and Eryigit (2017).
- Days of week gazetteer as used by Bontcheva et al. (2013).
- Honorific words (e.g., *"bey"*, *"hanım"*, *"hocam"*, *"sayın"*, etc.) gazetteer as used by Ertopçu et al. (2017). This gazetteer also includes some of the person titles in Turkish such as *"dr"*, *"şht"*, *"başk"*, and *"md"*.

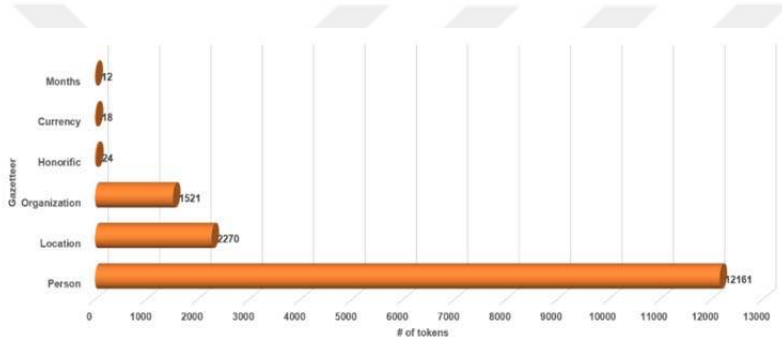


Figure 3.5. The number of tokens in our gazetteers.

Moreover, our gazetteers include PLOs that we extracted from a dataset introduced by Tur et al. (2003). Figure 3.5 depicts the number of tokens in our gazetteers.

3.6. The Problem of Privacy

In this section, we present a brief formal definition of the privacy problem with respect to OSNs. Let $U = \{U_1, U_2, \dots, U_n\}$ contains data for n OSN users. The data $U_i \in U$ for the user i is a tuple of the form $U_i = \langle I_i, D_i \rangle$, where I_i and D_i are user i's identifier and crawled data respectively. The crawled data includes profile information, liked pages, wall contents, relationships, friendships, and wall interactions. Some of our mining (i.e., SA, NER) and attribute inference tasks (i.e., gender detection, kinship detection) are performed over a different subset $U' =$

$\{U'_1, U'_2, \dots U'_m\}$ such that $U' \subset U$ and $m < n$. These tasks, in this thesis, exploit ML and/or DL techniques to develop highly accurate prediction models on the subset U'. So, given a user dataset U', identifiers are removed and each D'_i is converted to a fixed number of features. Then, these features in tabular form are used to build a learning function $f(D'_i) = c$, where c is one of the predefined set of k classes represented by $C = \{c_1, c_2, \dots c_k\}$. These definitions store the following information/values for the following tasks:

- SA: U' includes data of 12,660 users, each D'_i is a randomly selected activity content, and C is a set of two categories, namely, positive and negative.
- NER: Employed different datasets some of which are not created from the crawled OSN data. For the FBNER dataset, U' includes data of 5,000 users, each D'_i is a randomly selected activity content, and C is a set of seven coarse-grained NE types which are Person, Location, Organization, Date, Time, Money, and Percent.
- Kinship Detection: U' contains disclosed kinship relations in the largest snapshot of the crawled data, while each D'_i includes activities targeting wall owner. Finally, C is a set of nine single (see Table 3.7) or six mutual (see Table 3.8) kinship types.
- Gender Detection: U' includes data of 8,422 users, while each D'_i in the U' contains profile information, wall contents, and wall activities of user i. Finally, C is a set of two categories, namely, male and female.
- User Identification: U' contains data of 648 users and each D'_i in the U' contains only the username of user i. Finally, C is a set of two categories, namely, positive and negative to imply that whether two usernames match or not.

We would like to note that U' contains a different set of users along with their different types of data to be used as an inference channel. Depending on the task at hand, U' may include the same user but not his/her same data stored in the D'_i. This is because some of our tasks create their basis training data by randomly selecting the data of users. For instance, in the SA task, two or more activities of the same user may have been selected randomly. Similar cases are also true for NER and user matching tasks as well. For instance, in the user matching task, different usernames of the same user are included into the U' to make it possible to create positive instances. More formally, the U' may contain $U'_i = \langle I'_i, D'_i \rangle$ and $U'_j = \langle I'_j, D'_j \rangle$ such that $I'_i = I'_j$ when the data of D'_i and D'_j are different parts of the data of the same user.

In the formal definition given above, D' stores m OSN users' data to use as an inference channel. From the database perspective, an inference channel in a database is a means by which one can infer data classified at a high level from data classified at a low level (Jajodia and Meadows, 1995). In this thesis, we consider users' data as the inference channel which is used to learn/infer attributes or other aspects of users. To exemplify, let $\langle a, f(a), g \rangle$ triplet denotes the first name data where a is the first name, $f(a)$ is its usage frequency in the OSN, and g is the gender of users having the first name a respectively. Using this data, an adversary can infer any user's gender by querying his/her first name c, that will have resulted in two triplets $\langle a_i, f(a_i), g_i \rangle$ and $\langle b_j, f(b_j), g_j \rangle$ where $f(a_i) \neq f(b_j)$ and $a_i = b_j = c$. In such a case, we say that Ch is a set of first names along with their frequencies and assigned gender attributes such that $\forall \langle x, f(x), g \rangle \in Ch, f(x) > 0$ and it is an inference channel for a set of the first names $F = \{(y, f(y)) | f(y) > 0, y \in I\}$, where I is the collection of all of the first names disclosed by users in the OSN. A similar and partial function, namely, FCG is used in this thesis to get a clue about users' gender (see Section 4.4.3). Deriving many other examples is also possible.

In this thesis, we first classify the OSN data at a low level under the purpose of the task at hand. We then use this data as our inference channel to build highly accurate inference models. In the majority of our sub-tasks, we use wall contents of users to create our inference channels via extracted features. However, in some others, we also use social connections, wall interactions, and network structure as well.

4. METHODS

This section describes all methods which are used in subtasks of this thesis. The following subsections give detailed information about used graph notation, algorithms to crawl Facebook data, graph analysis algorithms, and so on.

4.1. Graph Notation

This subsection presents the essentials of graph theory with a focus on notation, basic definition, types, and properties of graphs.

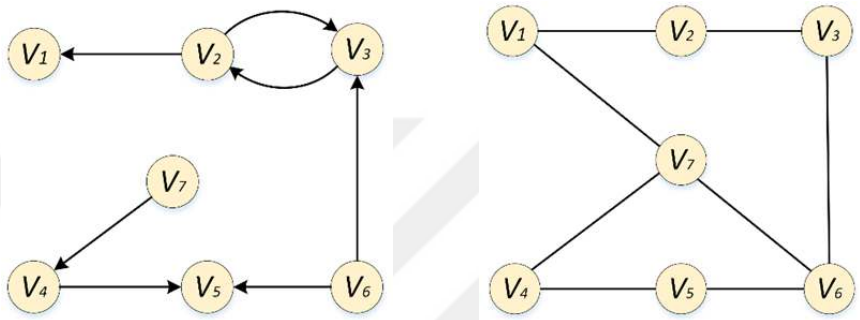
4.1.1. Graph Basics

A graph contains two sets of objects which are called nodes and edges where nodes represent fundamental elements of which graph is formed and edges correspond to the connection between these nodes. Nodes are major parts of any graph and they are called vertices or actors depending on the context. For example, in a web graph, nodes and edges represent websites and web-links between them, while in an OSN, graph nodes and edges correspond to people and their friendship connections respectively (Zafarani et al., 2014).

Graphs are used to model many applications and OSNs are often represented as graphs in computer science. Formally, a graph G can be represented as $G = (V, E)$ where $V = \{v_1, v_2, \dots, v_n\}$ denotes a set of nodes and $E = \{e_1, e_2, \dots, e_m\}$ represents a set of edges (Erciyes, 2018), where $|V| = n$ and $|E| = m$ correspond to node-set size (i.e., size of the graph) and edge set size respectively. Edges are also represented by their endpoint nodes by using $e(v_i, v_j)$ or $e(v_j, v_i)$ representations which mean that there is an edge between nodes v_i and v_j. Any graph $G(V, E)$ can also has a $G'(E', V')$, the subgraph with properties (Zafarani et al., 2014) such that $V' \subseteq V$ and $E' \subseteq (V' \times V') \cap E$.

Edges in any graph can have directions depending on the context. A directed edge (or arc) means that one node is connected to another, but not vice

versa (Zafarani et al., 2014). On the other hand, an undirected edge means that its endpoint nodes are connected in both ways. If a graph is composed of directed edges, it is called a directed graph, whereas a graph including undirected edges is called undirected. Notice that in a directed graph (or digraph), $e(v_i, v_j)$ is not the same as $e(v_j, v_i)$.



(a) Directed graph (b) Undirected graph

Figure 4.1. Two graphs with directed (a) and undirected (b) edges.

Based on the edge directions, directed graphs can be defined to include a set of ordered pairs of vertices, while undirected graphs include unordered pairs (i.e., a pair (a, b) which has a as its first element and b as its second element) of vertices (Zhang and Philip, 2014). Any graph which includes both directed and undirected edges is also called a partially directed graph. Figure 4.1 shows sample directed and undirected graphs that have an equal number of nodes (Zafarani et al., 2014).

To make computations over graphs, it is needed to convert them into suitable forms. Widely used ways to represent graphs are edge list, adjacency matrix, and adjacency list (Zhang and Philip, 2014; Zafarani et al., 2014). Edge list is a simple and common approach in which each element is an edge (v_i, v_j) indicating that node v_i is connected to node v_j. For the directed graph shown in Figure 4.1, we have the following edge list representation:

(v_2, v_1)

(v_2, v_3)

(v_3, v_2)

(v_4, v_5)

(v_6, v_3)

(v_6, v_5)

(v_7, v_4)

Adjacency list represents a graph with an array of linked lists from which each list includes a node and its neighbors connected to it. This list is often sorted by node order. Table 4.1 presents the adjacency list for the directed graph shown in Figure 4.1a.

Table 4.1. A sample adjacency list

Node	Neighbor(s)
$\boldsymbol{v_2}$	v_1, v_3
$\boldsymbol{v_3}$	v_2
$\boldsymbol{v_4}$	v_5
$\boldsymbol{v_6}$	v_3, v_5
$\boldsymbol{v_7}$	v_4

Adjacency matrix, on the other hand, represents a graph by a matrix $A[n, n]$ where each element $a_{ij} = 1$ if there is a connection from vertex i to j and $a_{ij} = 0$ otherwise. Table 4.2 presents the adjacency matrix of the directed graph that is shown in Figure 4.1b. When OSN graphs are represented by the adjacency matrix many cells remain zero that produces a large sparse matrix. Therefore, adjacency list and edge list ways are more frequently used in OSN research to reduce required space. On the other hand, the adjacency matrix is often used for dense graphs as searching an edge in this matrix requires constant time. Any graph has connectivity in its nature that defines how its nodes are connected via a set of

edges. A node v_i is connected to v_j (or v_j is reachable from v_i) if it is adjacent to it or there exists a path from v_i to v_j.

Table 4.2. A sample adjacency matrix

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	0	0	0	0	0	0
v_2	1	0	1	0	0	0	0
v_3	0	1	0	0	0	0	0
v_4	0	0	0	0	1	0	0
v_5	0	0	0	0	0	0	0
v_6	0	0	1	0	1	0	0
v_7	0	0	0	1	0	0	0

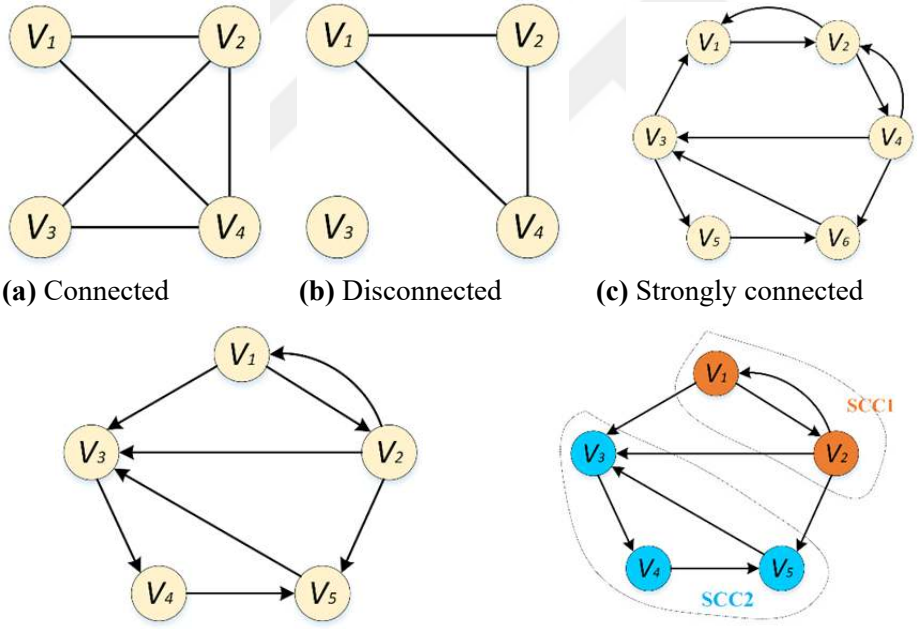
In this thesis, we consider a set of n users participating in an OSN as a directed graph $G(V, E)$ where V is a set of nodes (i.e., vertices) $V = \{v_1, v_2, \dots, v_n\}$ in which each node $v_i \in V$ stands for a user. On the other hand, E represents a set of directed edges $E = \{e_1, e_2, \dots, e_m\}$ such that each edge $e_i \in E$ corresponds to a friendship link between a pair of users $v_i, v_j \in V$. Each user in V may disclose information related to a set of topics $T = \{t_1, t_2, \dots, t_t\}$ which represents his/her profile items such as gender, age, job, religious view, and so on.

We would like to note that OSN data contains various complex types of interactions between users. Therefore, the best data structure to represent OSN data appears to be graphs (Leskovec and Krevl., 2014). However, choosing the exact type of graph is a difficult task. One has to decide between directed vs. undirected graphs and simple vs. multigraphs (Balakrishnan, 1997). In this thesis, we prefer representing an OSN with a simple graph with directed or undirected type depending on the task at hand. We also use edge list representation to make our computations on a graph representation of OSN data.

An undirected graph is connected if there is a path between any pair of its nodes. A digraph is weakly connected if there is a path between any pair of nodes without following the edge directions. The graph, on the other hand, is strongly

connected, if there is a directed path with following edge directions between any pair of its nodes (Zafarani et al., 2014).

Based on these connectivity definitions, it is said that an undirected graph is disconnected or connected, whereas a digraph is weakly or strongly connected. An undirected graph may also include one or more subgraph(s) (i.e., components) in which there is a path between every pair of nodes. In a digraph, a component is strongly connected if there are opposite paths between any pair of nodes, otherwise, it is a WCC which may also include SCCs as well. Notice that in a WCC, nodes cannot be visited by a single path. Figure 4.2 depicts disconnected, connected, weakly connected, strongly connected graphs, and SCCs in a weakly connected graph.



(a) Connected **(b)** Disconnected **(c)** Strongly connected

(d) Weakly connected **(e)** SCCs in weakly connected

Figure 4.2. Connectivity in graphs.

When a graph is constructed, a common practice is to identify its attributes and show measurements of these attributes to define its characteristics. Commonly used metrics include (Spiliotopoulos and Oakley, 2013; Zafarani et al., 2014):

- Graph Size: The number of nodes in the graph.
- Graph Density: Represents how many of a node's neighbors are connected. It is calculated as the ratio of the number of connections to the number of possible connections.
- Average Degree: Represents the number of nodes that have mutual connections.
- Average Path Length: Average distance between all pairs of nodes in the graph.
- Diameter: The longest distance within the graph. In other words, the maximum distance between any two given nodes.
- Graph Modularity: A scalar value between -1 and 1 which measures the density of connections inside communities as compared to connections between communities.
- The Number of Connected Components: The number of distinct clusters within a graph.
- Average Clustering Coefficient: A measure of the embeddedness of a node in its neighborhood.

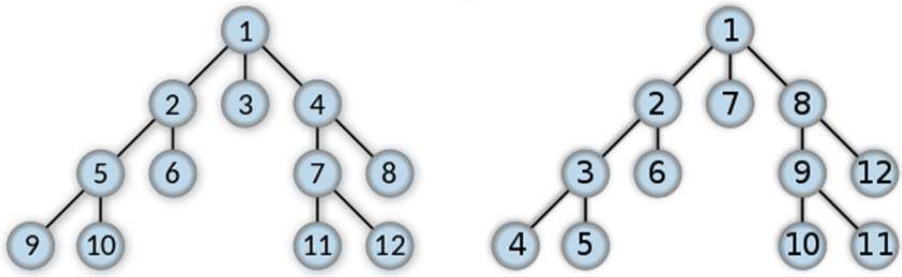
4.1.2. Graph Algorithms

In this thesis, we use several graph algorithms including BFS, DFS, Tarjan's SCC algorithm, and page rank for different purposes such as graph traversal and performing network analysis.

4.1.2.1. BFS

BFS is an algorithm that explores (or traverses) nodes by using a level-by-level approach to traverse or search graph data structures (Pathak et al., 2018). BFS starts at the root or any arbitrarily selected node and explores the edges of the graph to discover all vertices which are reachable from the root (Cormen et al., 2009). It takes linear time (i.e., complexity equals the sum of the number of edges and nodes) to search the overall graph. Figure 4.3a depicts an illustration of BFS on a simple graph where nodes are explored level by level (Anonymous 2019e).

Unlike the DFS which explores the node branch as far as possible before backtracking and expanding other nodes (Cormen et al., 2009), it discovers all vertices at level *n* from the root node before discovering any vertices at next level *n* + 1 (Rahim et al., 2018). BFS can be used to solve various problems in graph theory including testing bipartiteness of a graph, finding the shortest path and minimum spanning tree for unweighted graph, crawlers in search engines, and so on (Pathak et al., 2018, Anonymous 2019e).



(a) BFS (b) DFS

Figure 4.3. Traversing nodes in a BFS and DFS order

4.1.2.2. DFS

DFS is an algorithm for traversing a graph. There are several ways of searching a graph depending on the way that selects edges to search. DFS always chooses an edge which is emanating from the vertex which is visited most recently

and still has unexplored edges (Tarjan 1972). It stops searching when all vertices have been visited and it is very easy to program both in iterative and recursive fashions (Reif 1985).

DFS is used to go through or fetch items within data structures, graphs, or trees, whose basic characteristics is to go through all child nodes to the root node as deep as possible and then perform backtracking (Rodrigues et al., 2019). An illustration of a DFS algorithm on a graph is shown in Figure 4.3b (Anonymous 2019d). DFS is typically traversing the entire graph and takes time linear in the size of the graph (Anonymous 2019d). It is a key step in many problems such as finding SCCs, graph bridges, biconnected components, shortest path, and so on (Yang et al., 2019).

4.1.2.3. Strongly Connected Components

This is one of the earliest graph algorithms and proposed by Tarjan (1972) to find SCCs of a directed graph. It takes a directed graph as input and divides the graph into its SCCs that each of them is a subgraph where there is a path between all pairs of vertices. For instance, there are 3 SCCs in the graph (Anonymous 2019c) which is depicted in Figure 4.4.

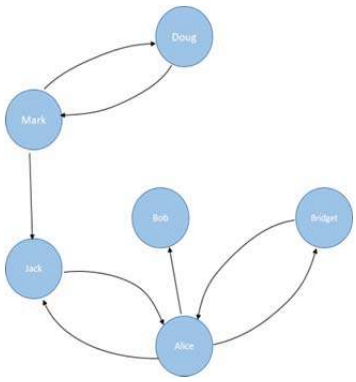


Figure 4.4. A sample directed graph of users.

The first and biggest component has members *Alice*, *Bridget*, and *Jack*, whereas the second component has *Doug* and *Mark*. *Bob*, on the other hand, is just a single member of the smallest component, as there is no outgoing edge from that node to any of the others. Tarjan's algorithm can be considered to contain two traversals of the graph.

A DFS traverses all edges and creates a depth-first spanning tree/forest at first. Secondly, once it finds the root of an SCC, all of its descendants which are not members of previously found components are included in this component (Nuutila and Soisalon-Soininen, 1994). This second traversal is implemented by using a stack, where each node is stored when visited by the DFS. If the root of a component is exited, all of its descendants are popped from the stack which forms the component. The Tarjan procedure is called once for each node and considers related node's each edge just once. Therefore, its running time is linear in the number of edges and nodes in the graph (Anonymous, 2019b).

4.1.2.4. PageRank

PageRank is proposed to measure the relative importance of web pages by computing a ranking for every web page based on the graph of the web (Brin and Page., 1998). In other words, it is a method for ranking the vertices in a graph according to their relative structural importance (Agirre and Soroa., 2009). The main idea of the PageRank is that whenever a link from v_i to v_j exists in a graph, a vote from node i to node j produced, and the rank of the node j increases. Besides, the strength of this vote also depends on the rank of the node i such that the more important node i is, the more strength its votes will have (Agirre and Soroa., 2009). For a directed graph $G(V, E)$, where $V = \{v_1, \dots, v_n\}$ is a set of n vertices and $E = (v_i, v_j)$ is a set of directed edges, the calculation of the PageRank vector Pr over G is equivalent to solve the following equation (Brinkmeier, 2006; Agirre and Soroa 2009):

$$Pr = dMPr + (1 - d)v \tag{4.1}$$

where v is a $n \times 1$ vector whose elements are $\frac{1}{n}$, d is a damping factor which is a scalar value between 0 and 1, and M is $n \times n$ transition probability matrix and defined as follows:

$$M_{ij} = \begin{cases} 1/d_i, & if\ a\ link\ from\ v_i\ to\ v_j\ exists \\ 0, & otherwise \end{cases} \tag{4.2}$$

where d_i is outdegree (i.e., number of outbound edges) of node v_i. In other words, $M = (K^{-1}A)^T$ where A is the adjacency matrix of the graph and K is the diagonal matrix with the outdegrees in the diagonal. In the traditional PageRank, the vector v is a stochastic and uniform constant vector which means that assigning equal probabilities to all nodes in the graph gives way to random jumps. However, the vector v can be non-uniform and assign stronger probabilities to certain kinds of nodes (Haveliwala, 2002), effectively biasing the resulting PageRank vector to preferred these nodes. Using a modified vector v is often called personalized PageRank which is used to create a personalized view of the relative importance of nodes (Agirre and Soroa., 2009; Pensa et al.,2019). PageRank can be computed using the well-known power iteration method (Golub and Van der Vorst., 2000) until convergence below a given threshold is achieved or typically until a fixed number of iterations are executed. The PageRank algorithm is reported to converge quickly even for graphs including millions of nodes thus, the effective complexity is linear in the number of edges (Brin and Page., 1998).

4.2. Crawling Public Facebook Data

In this thesis, it is preferred to design and implement a crawler to collect public data from Facebook OSN. Following sub-headings present our data model, data collection algorithms, and implementation details.

4.2.1. Data Model

A star schema is designed to model data, where a user relation is placed at its center and composed of 20 tables in total. In this model, each object in a Facebook account is considered as an entity. However, it is needed to use some extra tables to store the data in a meaningful and lossless way. On the other hand, decomposing is applied over tables to prevent data redundancy as possible, since too many join queries among multiple tables harm performance (Coban et al., 2020a). Username given by Facebook is used as a primary reference to distinguish users. Notice that this model is designed to store just the textual data of users. MySQL Workbench DBMS is used to design our data model and its star schema is shown in Figure D.1.

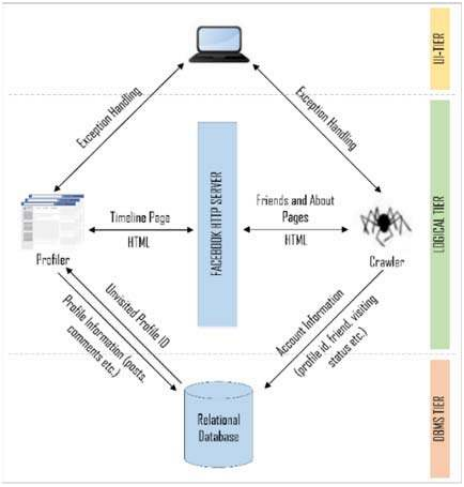


Figure 4.5. Design of data collection framework.

4.2.2. Data Collection

To collect data, a system is designed with three-tiers that are called as UI tier, DBMS tier, and LOGICAL tier. As seen in Figure 4.5, the UI tier enables us to follow the logical tier components that handle any exceptional cases manually. The DBMS tier implements the data model that is described in the previous heading. Finally, the LOGICAL tier contains our web spider which is composed of two components, namely, crawler and profiler. This multi-threaded crawler is developed by taking mainly the following things into consideration (Coban et al., 2020a):

- It must visit each account just one time.
- It must provide concurrent access control to the database since multi-threading enables it to run in a parallel fashion.

The crawler discovers accounts that are publicly reachable from the seed account u. The profiler, on the other hand, visits profile pages of discovered accounts and extract all publicly shared textual data. We prefer BFS (Cormen et al., 2009) of the Facebook OSN to ensure that each account is visited just once and give priority to discovered closer accounts to the u.

In detail, the crawler inserts the seed account u in the database at first. Then it continues discovering accounts by visiting its friends, friends of friends, and so on. For each discovered account v, the crawler checks whether the friend list of v is public. All discovered but unvisited friends of the v are stored in a FIFO queue. This BFS implementation stops only when all publicly reachable friends of u are visited. Crawler threads in BFS nature generate visit-order for the profiler threads. Notice that a profiler thread aims to extract and store all publicly shared content in related Facebook account.

4.2.2.1. Crawler Module

The crawler relies on a helper relation that is named as *Crwlr*. This relation contains four simple fields which are *Facebook ID* (i.e., *username* or *fid* in short) of an account as a primary key, a boolean flag *all_visited* that corresponds to whether this account's friend list visited and inserted into the table, a boolean flag *tl_visited* that indicates whether the account's profile page is visited and processed, and a field *depth* that shows how current account is far from the seed account. Our crawler supports multi-threaded execution through a thread pool.

As seen from Figure 4.6 (Coban et al., 2020a), the master thread creates the *Crwlr* relation, inserts the *fid* of the seed account, and then visits its friend list in Facebook OSN (line 3). Then it begins an infinite loop in which each iteration, the *fid* of an unvisited user from the current depth is retrieved.

Require: The seed user's fid (i.e., $username$)
Ensure: Traversal of the seed's in/direct friends in BFS order
1: **if** Relation $Crwlr$ does not exist **then**
2: Create an empty relation with schema $Crwlr(fid, all\_visited, tl\_visited, depth)$
3: Insert $(fid, false, false, 0)$ into $Crwlr$
4: **end if**
5: **loop**
6: Begin transaction
7: $vid \leftarrow fid$ of an unvisited user with $min(depth)$
8: $vid.frnds \leftarrow getFriends(vid)$
9: $depth = depth + 1$
10: **for all** $w \in vid.frnds$ **do**
11: Insert $(w, false, false, depth)$ into $Crwlr$
12: **end for**
13: Set $all\_visited$ for vid
14: End transaction
15: **end loop**

Figure 4.6. Pseudo-code of the crawler's master thread.

This user is identified by *vid* and all of its friends are retrieved (line 8) from the OSN and these friends are inserted with an incremented depth into the *Crwlr* relation at line 11. We assume that *vid.frnds* will be null if the friend list is not public. At line 13, *vid* will have been visited by the crawler, therefore the table is updated accordingly. Notice that the entire operation of visiting a user's friend list is enclosed packed into a database transaction (lines 6-14). This precaution ensures consistency under possible failure scenarios. The crawler can be stopped and restarted arbitrarily.

4.2.2.2. Profiler Module

Profiler module supports multi-threaded execution and the threads are controlled with a master-slave organization. The algorithmic design of the profiler's master thread is given in Figure 4.7a (Coban et al., 2020a). The master thread creates the data model (see Section 4.2.1) at first to store public profile information of the current account. Then, it invokes as many workers as necessary within an infinite loop (lines 4-6).

Require: Crawler populating the $Crwlr$ relation
Ensure: Traversal of discovered accounts in BFS order
1: **if** Data model is not implemented **then**
2: Create the schema for storing the public data of discovered profiles
3: **end if**
4: **loop**
5: $vid \leftarrow fid$ of a user whose $tl\_visited$ is $false$
6: Fork into $Profiler - WorkerThread(vid)$ {*a thread pool controls all live workers*}
7: **end loop**

(a)

Require: $\exists vid \leftarrow fid$ of a non-profiled user
Ensure: Inserts all public data within the profile page of vid
1: Begin transaction
2: Extract basic info: gender, phone, interest etc.
3: Extract places, work, education
4: Extract family members, relationships
5: Extract wall activities (posts, comments etc.)
6: Set $tl\_visited$ for vid
7: End transaction

(b)

Figure 4.7. Pseudo-code of the profiler's (a) master thread and (b) worker thread.

A worker thread of the profiler extracts basic attributes (e.g., gender, date of birth, political view, phone, etc.) and textual wall activities (e.g., post, comment, and reply) of a non-profiled user with the smallest *depth*. It then stores these

extracted data in the star schema. In the final step, it marks the related account as profiled (line 6) via the *tl_visited* flag. Note that if there is not any publicly available information in the related user's profile, it is assumed the data is to be *null*. Similar to the crawler module, the entire operation in the profiler's worker thread is enclosed within a database transaction (lines 1-7) to ensure concurrent access control. Figure 4.7b presents the pseudo-code for a worker thread of the profiler (Coban et al., 2020a).

4.2.3. Implementation Details

This section presents details of implementation, faced challenges, and methods used to tackle the challenges. Our crawler is implemented by using the Selenium API. It extracts public data from accounts by locating relevant elements in the DOM tree of each account's HTML source that is returned by the Facebook HTTP server. DOM provides the structured representation of a web page where nodes denote the HTML tags and tree hierarchy represents the organization of nested elements (Wani et al., 2018). Locating elements in the HTML source is performed by using selectors in XPath query language that can locate elements in an XML document. The reason for this is that HTML can be an implementation of the XML (i.e., XHTML). For instance, a selector defined as *"//title[@lang='eng']"* selects all elements which have a lang attribute with a value of *"en"*.

However, locating elements to extract information is not simple as Facebook uses dynamic content loading which complicates the data collection task. That is, elements to be located are not available in the HTML source codes of Facebook pages. This dynamic content needs to be triggered by user interactions (e.g., scrolling a page, clicking a link or button, etc.) with the related page. Connection pooling is also used to ensure concurrent access of multiple threads to our database.

Connection pooling is a mechanism to create and maintain a collection of JDBC connection objects. In this mechanism, each thread borrows a separate connection from the connection pool, use it, and then leave it back to the pool by closing it. The use of the connection pool improved the performance of the crawler, as it allows re-usability of the pool of connection objects. During the implementation process of the crawler, some challenges are faced to be tackled in this thesis. These challenges mainly stem from the structure and employed techniques of the Facebook OSN.

As stated before, the primary problem is the dynamic content loading structure that requires to simulate human interactions with the Firefox browser. There also other challenges caused by the Firefox browser and the restriction of the Facebook OSN for users. In more detail, the Firefox browser throws a binding exception when the crawler runs in a multi-threaded mode for more than 5 processes. Facebook restricts an account or IP address to have access to limited accounts in a time interval. As a sample scenario, Facebook thinks that a user may be a computer if the user frequently accesses accounts which are not friend list of neither itself nor its friends.

Figure 4.8. Screenshot of the seed user's page when it is blocked by Facebook.

As seen from Figure 4.8, when this restriction is exceeded, Facebook temporarily blocks the account that is used to login by the crawler. If a user is blocked, verification is required through a phone number or personal photograph. In this thesis, crawler logins to Facebook within short time intervals with fewer threads per process to avoid blocked. Another important point is that the crawler needs to establish a consistent connection with the Facebook HTTP server, which requires a large amount of network bandwidth. One of the more important of these is that Facebook has a rather complex structure and dynamic HTML template in the background (i.e., it updates attributes of HTML elements at regular intervals) as depicted in Figure C.1 and Figure C.2. These make the crawler's maintenance difficult. In this thesis, a crawler is developed with a control mechanism to detect and overcome such challenges.

4.3. Data Preprocessing

In this thesis, we use two different ways to preprocess textual datasets. The first way is basic preprocessing which includes the following steps:

- Removing all punctuations and digits, and
- Lowercasing the content

The second way is, on the other hand, linguistic preprocessing that includes the following steps:

- De-asciifying: OSN textual content is often dirty and created by using informal language. Facebook users may write words without Turkish characters and this prevents making linguistic analysis. Hence, this step replaces characters (e.g., *"aldik"* → *"aldık"*) with their Turkish equivalents.

- Stemming: In Turkish, it is possible to produce many surface forms of a single root word. As such, this step reduces (e.g., *"geldi"* → *"gel"*) words into their roots.
- Removing stopwords: This step removes stopwords from the content.
- Removing repetitive letters: This step reduces the number of repetitive letters (e.g., *"kardeşiiiimmmm"* → *"kardeşim"*) with more than two occurrences until the word is recognized as a Turkish word by Zemberek. Even the word is not Turkish after this reduction, we left it untouched. This is because Turkish includes words having duplicate letters such as *"saat"*, *"matbaa"* and so on.

Notice that we use the Zemberek tool to perform all linguistic analysis in this thesis. We also apply linguistic preprocessing based on our task which may require to exclude or include any step.

4.4. Feature Engineering

In this thesis, feature sets are mostly extracted by using text mining techniques as crawled data only includes users' textual data. Used feature sets vary depending on both the task they utilized and the type of source they are extracted. Our feature sets in this thesis include content, lexicon-based, profile, social interaction based, and other task-dependent features. The following sub-headings give details about our different feature sets.

4.4.1 Content Features

Some of our tasks in this thesis considered as a classical TC problem. Therefore, we borrow content features from this field for these tasks such as sentiment analysis and gender detection (see Section 5.1.3 and Section 5.2.2.2).

4.4.1.1. BOW and N-grams

BOW features are a set of unique words which are observed in related text or document. Notice that the number of unique features may be affected by applied preprocessing in the BOW model (Coban and Ozyer, 2018). On the other hand, n-gram feature generation is performed in two different ways which are character and word level, and n-grams are obtained by taking n consecutive characters or words in the text depending on the level. The character level n-gram model can tackle misspelling and abbreviations as it is language-independent (Kanaris et al., 2007). Table 4.3. presents obtained BOW and n-gram features for a sample piece of text *"social nets"*. Notice that in Table 4.3, '-' represents whitespace character, and '|' denotes n-gram boundary.

Table 4.3. BOW and n-gram features on a sample piece of text

Model	Level	Features
BOW	Word	|social|, |nets|
Uni-gram	Word	|social|, |nets|
Bi-gram	Word	|social nets|
Bi-gram	Character	|so|, |oc|, |ci|, |ia|, |al|, |l_|, |_n|, |ne|, |et|, |ts|
Tri-gram	Character	|soc|, |oci|, |cia|, |ial|, |al_|, |l_n|, |_ne|, |net|, |ets|
Quad-gram	Character	|soci|, |ocia|, |cial|, |ial_|, |al_n|, |l_ne|, |_net|, |nets|

4.4.1.2. Distributed Embeddings

Distributed embeddings imply to use of word or document vector as features. In this thesis, we use the avg-word2vec approach in which each dimension of the document vector is considered as a feature. We apply this simple approach as done by Kilimci and Akyokus (2018) and obtain a vector of related text by taking the average of the vectors of words that are observed in the related document. On the other hand, document vectors are also used to represent texts by considering each dimension of the inferred document vector by related PV model as a feature. Notice that this strategy is valid only for methods on the ML side as DL methods use a different strategy to feed a learning method (see Section 4.6).

4.4.2. Lexicon-based Features

The basic idea behind this approach is to extract features based on language-specific lexicons. One of the common practices in the ML field is to use task-dependent lexicons to extract features. For instance, lexicons are used in NER studies to catch NEs observed in sequences. Similarly, sentiment-words lexicons can be employed to calculate the overall polarity of any textual content in SA studies. It is also possible to extend such a scenario for the majority of other tasks in ML fields. As such, in this thesis, we use several lexicons to extract discriminative features depending on our task at hand.

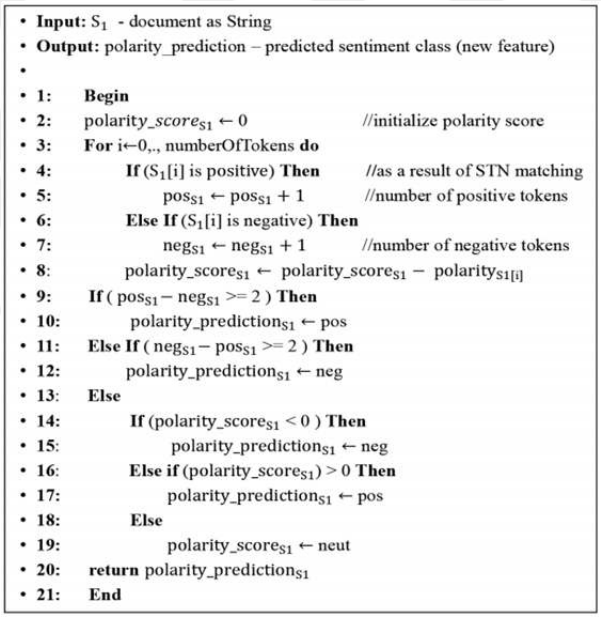


Figure 4.9. Unimproved lexicon-based feature generation

4.4.2.1. Polarity Features

These features are created and used in our SA task by using a lexicon-based approach proposed by Erşahin et al., (2019) and depicted in Figure 4.9. We use the eSTN lexicon (with a version including objective and multi-words) from the same study and extract a single feature, namely, F1 by adapting the following

techniques to this method: (i) support for searching multi-word terms and (ii) improving negation handling and adding polarity shifting by employing rules and booster words (i.e., *"en (the most)", "gerçekten (really)", "çok (very)", and "bayağı (too many/much)"*) from Gezici and Yanıkoğlu (2018).

Note that F1 takes a value of 0 for negative or 1 for positive based on polarity scores of the words which are observed in the related document (Erşahin et al., 2019). We also extract another feature, namely, F2, that includes both the number of positive and negative seed words observed in the current text as done by Gezici and Yanıkoğlu (2018) and Salur et al. (2019).

4.4.2.2. Gender-Oriented Features

We extracted gender-oriented features from the wall activities of users as well. These features are based on both activity content and interacted users (i.e., friends of WO) who make their gender attribute public. Content features are extracted by using our small lexicon of gender-oriented Turkish words (see Section 3.5.1). Interaction features, on the other hand, are extracted by considering whom the WO answers, who answers the WO, and whom the WO tagged in any post. These features are grouped as follows:

(1) Features based on what users say to each other: In this group, features are extracted with the help of a lexicon-based approach which quantifies the number of gender-oriented words observed in activities. The basic idea is that what users say to each other and what others say to the WO can give clues about his/her gender.

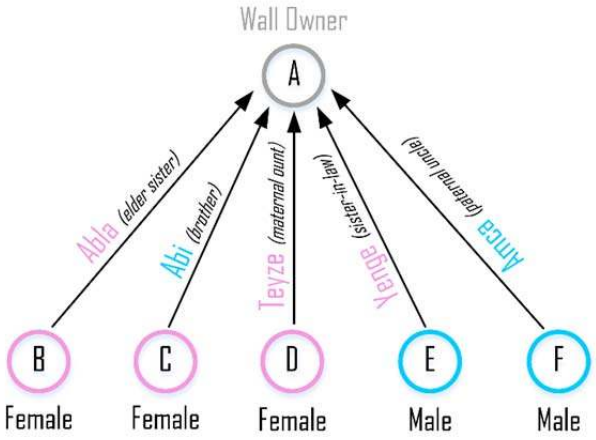


Figure 4.10. Users interacted with WO and gender-oriented words in their activities

As seen from Figure 4.10, WO is most likely male as other users mostly use male words in their activities which targeting WO. However, this approach has a few drawbacks to implement, due to it is too hard to detect which activity directly targets WO under any activity from his/her wall. For instance, there may be many comments and replies under any post from WO (see Figure 3.1). In this case, it is a challenging task to determine which one of the other users' activities directly targets WO (i.e., written to answer WO). This is because other users may answer each other under any activity from WO. Therefore, we assume that an activity targets WO only when it lays down one of the following conditions:

- It is a direct post (see Figure 1.6b) shared by another user on WO's wall.
- It is the first activity under any activity from WO.

However, there is still another challenge that it is possible to observe different gender-oriented words even in activities targeting WO. Let A and C be to Facebook users from Figure 4.10 and user A (i.e., WO in here) is a female. A male word (i.e., *"abi (brother)"*) is observed in a targeting activity from C (or F) due to the following reasons:

- Context: Even it is the first activity located under any activity from WO, user C may type about another user (most likely a male in this case) in the related activity.
- Topic: The topic of activity from the WO important as well, because it can affect what other users say. For instance, let WO shares a photograph of his/her son in a post. In such a case, other users (i.e., his/her friends) mostly type about the son of WO not about her/himself, and the majority of gender words will be male-oriented.
- Word sense: User C may use a gender-oriented word in a non-gender-oriented sense. Let consider two activities, namely, P_1 and P_2 that include the same gender-oriented word *"kız (girl)"* as following:
    - P_1: *"Bugün kız kardeşimin doğum günü (Today is my sister's birthday)"*
    - P_2: *"Kız kulesi küçük, şirin bir kuledir (Girl tower is a small, cozy tower)"*

      The word *"kız"* is used in gender sense in P_1, but the same is not true for the same word observed in P_2.

Detecting such cases requires an effective WSD process which is out of scope for this thesis. Using this idea that users often interact with other users of the same gender, we extract features in this group.

(2) Features based on interacted users: Another idea we use to extract features is that the number of male and female users who write activity on WO's wall may also be discriminative with respect to the gender attribute. As depicted in Figure 4.10, if the number of female users who type under any activity of WO is greater than the number of male users, then WO is most likely a female. Using this idea, we also extract features in this group depending on the following quantities:

- The number of male and female users tagged by WO in any of his/her posts. This information is extracted from the post title (e.g., *"Ahmet, Ali ve Ayşe ile birlikte film seyrediyor (Ahmet, watching a movie with Ali and Ayşe)"*), if exist.
- The number of male and female users who interact with WO in both their own walls and WO's wall.
- Quantities of the activity types.

(3) Features based on WO's activities: This group only considers the activities that are written (i.e., posted) by WO. Extracted features are based on quantities of male and female-oriented gender-words, interacted male and female users (i.e., who disclose gender attributes), and types of activities shared by WO.

Using these three groups of features, we extracted a list of 34 features which are summarized in Table 4.4. Notice that we use gender-oriented words lexicon to count the number of gender-oriented words observed in activities. In this process, each word from this lexicon is searched in related activity by using regular expression fashion, and the obtained value is used to create our features in related groups described above. To avoid biased results, we also use another small lexicon that includes non-gender-oriented words and phrases to remove such words from activities.

Table 4.4. Gender-oriented features and their definitions based on wall contents

No	Definition
1, 2	# of **male/female words** in all activities posted by WO (2 attributes)
3, 4	# of **male/female words** targeting WO (2 attributes)
5, 6	# of **male/female words** filtered by term-gender and targeted WO (2 attributes)
7	**Average # of replies** written under comments by WO (in both WO's wall and other users' walls)
8	**Average # of replies** for other users' comments written under posts by WO
9, 10	**Average # of comments/replies** written under posts by WO (2 attributes)
11, 12	**# of all male/female users** who write an answer under any post or comment by WO (either the target is WO or not, 2 attributes)
13, 14	**# of male/female words** in activities posted by other users under any post or comment by WO (either the target is WO or not, 2 attributes)
15	# total number of Facebook reactions for posts by WO
16, 17	# of **male/female users** who are tagged (e.g., *"Ali is together with Ahmet and Mehmet"*) by WO in any of his/her post (2 attributes)
18, 19	**# of comments/replies** written by WO (2 attributes)
20	# of **posts** written by WO
21, 22	# of posts by WO that include **photograph/important event** (2 attributes)
23, 24	# of posts by WO that include **user-generated text/video** (2 attributes)
25, 26	# of posts by WO that include a **memory/location** (2 attributes)
27, 28	# of posts by WO that include information about a **game/survey** (2 attributes)
29	# of posts by WO that include a **link**
30	# of the **direct post** in WO's wall
31, 32	# of **male/female users** who are responded by WO under their comments and replies (2 attributes)
33, 34	# of **male/female users** who writes a comment or reply under any activity by WO (target is WO, 2 attributes)

Notice that the term-gender filter strips out such terms that their gender orientation does not comply with WO's gender. For instance, let assume that WO is a male and we detected *"amca (uncle)"* and *"kız kardeş (sister)"* terms in any activity on WO's wall. If the term-gender filter is active, it eliminates the word *"kız kardeş"* and keeps the word *"amca"* due to their gender-orientations are female

and male respectively. In other words, the term-gender filter looks for compliance between the gender orientation of the detected term and WO's gender.

4.4.2.3. Kinship Features

These features are employed in our kinship detection task and extracted from wall activity contents of users in kinship datasets (see Section 3.3.5). Extraction of features in this set is based on the kinship lexicon (see Section 3.5.1) that includes Turkish kinship words. This feature set contains nine features, each of which represents one of the generalized kinship types given in Table 3.6. In other words, each of these features corresponds to the observed frequency of kinship words which refer to the related kinship type in activities. For instance, the first feature will take the value of observed frequency of kinship words that are used to refer to the *"Parent"* kinship in activity contents of users who form a kinship pair.

To obtain values of kinship features, we iterated over all activities of both users of each pair and search kinship lexicon words in their activity contents. However, in this phase, we had to face the following challenges:

- In Turkish, it is possible to produce different surface forms of a root word (e.g., *"anne"* → {*"anneler"*, *"annesiz"*, *"annelik"* etc.}).
- As seen from Figure 3.1, in Facebook posts, it is a very challenging task to detect the target user of activity especially when there are two or more activities under a parent activity.
- Users often use some of the kinship terms in their activities even they have not kinship relation in their real lives.
- It is possible to observe multiple kinship words in an activity as users may write about any other user of the topic.

In Turkish, some of the kinship words are homonyms with other non-kinship-oriented words. Therefore, there is a need to detect the sense of a matched word to know whether it is used to refer to a kinship relation or not. This challenge is also observed while extracting gender-oriented features and requires an effective WSD process. Notice that we keep such cases untouched as WSD is out of scope for this thesis and obtain values of kinship features by searching kinship lexicon words in activity contents with the help of regular expressions. In the searching phase, we use the following three different ways to manipulate the observed frequency of kinship words:

- Without Filter (WF): This does not require any condition and if any kinship term is observed in an activity, the value of the corresponding feature is increased.
- Gender Filter (GF): It looks at gender compliance between kinship words and the type of the kinship, otherwise the value of the corresponding feature is decreased. For instance, let the word *"amca (uncle)"* is observed in an activity of user A who is targeting a female user B. Then the value of the feature M4 or S7 (see the complete list of kinship features depending on dataset creation way from Table 3.7. and Table 3.8) will be decreased. This is because the word *"amca"* is a male-oriented kinship word in Turkish.
- Surname Filter (SF): It checks whether a kinship/relationship specified by a kinship-term requires the same surname, if there is no conformity, the value of the corresponding feature is decreased. For instance, if the word *"baba (father)"* is observed in an activity of user A who is targeting user B, it is required that these two users have the same surname to increase the value of the feature S1 or M5 depending on the dataset creation way (see Section 3.3.5)

Notice that these filters are applied easily as our kinship lexicon does not just include kinship words, but also includes gender-orientation and whether referred kinship type requires the same surname as well. We consider all activity types including posts, comments, and replies to search kinship words. However, for each pair of users, we search words only in their activities which follow each other subsequently. This is because it is very challenging to determine the absolute target of child activity as stated before. Note that, we consider direct and event posts to search terms even they do not have child activity since the target user is already clear in these posts.

If any term in any activity has a match in our lexicon, we change the value of the corresponding feature depending on whether the term meets the condition provided by the filter, if applied. If the term meets the condition, we increase the value of the referred feature by 1, otherwise, we decrease by 1. An important thing to note here is that we increase the value of the related feature by 2 if the term is observed in a direct or event post (see Figure 1.6c). This is because it is possible to detect the target user with high accuracy compared to the usual posts and their child activities (i.e., comment or reply).

4.4.3. Profile and Social Interaction Features

These features are extracted both from the profile and social interactions of users. Following sub-heading(s) give detailed information about this set of features.

(1) Gender-Oriented Profile Features: As Facebook insists its members to use their real names, users mostly adhere to this real name policy of Facebook. As such, the display name is one of the most important attributes to extract features, especially for gender inference tasks. In this thesis, gender-oriented features are used in our gender inference task and some of them are newly proposed. The majority of these features are extracted by using a FCG value that is based on FNF and FNM values and obtained as follows:

$$FCG = \begin{cases} 2, & FNM > FNF \\ 1, & FNM < FNF \\ 0 & otherwise \end{cases} \quad (4.3)$$

Gender oriented features are extracted from the following entities of the Facebook:

- Display name: Notice that the first token of WO's display name is taken as his/her first name in this thesis. However, some users in Turkey use a display name with *"TC"* prefix in different formats including *"TC."*, *"T.C."*, *"T.C"*, and so on. Users who have a middle name also mostly abbreviate their first names (e.g., *"J. Jack Sparrow"*). In such cases, the second token is taken as the first name of WO in this thesis.
- Username: The FCG value is obtained for users whose usernames are not a completely numeric string and contain multiple strings which are concatenated by a dot.
- Family members and relationships: Family members and private relations can give clues about WO's gender. Consider two Facebook users A and B and suppose that A shares that B is his wife. If so, A is most likely a male and this information can be accessed from the family members section of A and B (if accepts to disclose this information) on their Facebook profiles. Such a case is also true for the private relationships of Facebook users. In this thesis, the FCG value obtained for, if exist, family members and private relationships of WO, and it is used to extract features. Two kinship groups are used to detect the gender orientation of the disclosed family member(s). These two groups are given in Table 4.5.
- Important events: Important events (i.e., buying a car, moving home, giving birth, etc.) just contain information about important things and can give clues about WO's gender, if disclosed. For instance, if an event from WO's profile says that he/she gave birth *("doğum yaptı"* in Turkish), then

WO is most likely a female. Similarly, if WO reveals that he/she is in a relationship with a female user (i.e., event actor), then this user is most likely a male.

Table 4.5. Gender-oriented kinship types

Group	Orientation	Kinship Types
K1	Male	*brother, uncle, husband, son, father, grandfather*
K2	Female	*sister, aunt, wife, daughter, mother, grandmother*

Based on these points, we extracted 15 different features which are based on profile attributes and social interactions of users. Table 4.6 presents a list of these features and their definitions. Notice that zero represents the cases where related information is not disclosed or could not be obtained by our crawler.

Table 4.6. Gender-oriented features and their definitions based on profile items and social interactions

No	Definition	Value(s)
1	Relation status: single (1), in a relation (2), engaged (3), married (4)	[0-4]
2	FCG value for relationship actor	[0-2]
3	FCG value for WO based on its display name	[0-2]
4	FCG value for WO based on its username	
5	Interested in: females (1), males (2), or males and females (3)	[0-3]
6	FCG based on the # of male (M) and female (F) kinship types of all users who discloses as someone's family member and use the same first name with WO	[0-2]
7	If exists, FCG value based on the # of male (M) and female (F) oriented kinships of WO	[0-2]
8	Total # of friends (t)	t
9	Total # of male (1) and (2) female friends (i.e., friends who disclose gender attributes) of WO based on their FCG values.	[0-2]
10	Overall FCG value for display names of all friends	[0-2]
11	Whether WO has shared and gender-oriented event post or not	[0-1]
12	Whether WO shared an event post to disclose that he/she: is in a relation, is married, is engaged, met someone	[0-1]
13	FCG value for event actor display name, if exists	[0-2]
14	Whether WO shared a biography or not	[0-1]
15	Whether WO shared a quotation or not	[0-1]

4.4.4. Username Features

Even usernames are generally short strings, they often carry information that may reflect a user's characteristics and habits. As such, we extract features from discovered usernames in the 20K dataset to perform user matching task. In this thesis, we extract features from usernames by three aspects, namely, the content of usernames, the patterns used when creating usernames, and the similarity measures of each pair of usernames.

(1) N-gram features: We extract character level n-grams with a length of 2 and obtained $(26 + 10)^2 = 1296$ features, where 26 is the number of letters, 10 is the number of digits, and 2 is the length of n-grams (Wang et al., 2016). Notice that it is not possible to use Turkish letters in usernames.

(2) Pattern Features: This set of features is categorized into four categories which are L(etter)D(igit)-permutation patterns, LD-gram patterns, date, and keystroke patterns (Wang et al., 2016).

LD-permutation patterns include different patterns of letters and digits observed in usernames with limiting the number of continuous letters and digits to do not exceed 3. These patterns are *"only letters"*, *"only digits"*, *"letters + digits"*, *"digits + letters"*, *"letters + digits + letters"*, and *"digits + letters + digits"*.

LD-gram patterns are obtained by converting each letter and digit in usernames with L and D respectively. Using these two letters and length as 6, it is possible to obtain $2^6 = 64$ LD-gram patterns. Then features are extracted based on contained patterns in each username.

Date patterns include widely used date formats which are *"year + month + day"*, *"month + day + year"*, *"day + month + year"*, and *"month + day"*. As usernames may include date information including date of birth, these four features are extracted based on whether date patterns appear in a username or not.

Keystroke patterns are extracted by assigning a coordinate for each character on the keyboard. Then three keystroke pattern features are extracted for each username: (i) whether all characters are in the same row on the keyboard, (ii)

whether every two consecutive characters are adjacent on the keyboard, but not in the same row, and (iii) whether each character is the same with or adjacent to next one on the keyboard. Table 4.7. presents an overview of content and pattern features (Wang et al., 2016).

Table 4.7. Content and pattern features extracted from usernames

Features	Definition	Count
Content Features	Possible bi-gram patterns of letters and digits	1,296
Pattern Features	Letter digit patterns with a length of 6	64
	Letter digit permutations	6
	Date patterns	4
	Keystroke patterns	3

(3) Features based on Similarity Metrics: We extract features based on well-known similarity and distance functions which are mostly employed in both username and display name-based user matching studies (Raad et al., 2010; Peled et al., 2013; Zafarani and Liu, 2013; Goga et al., 2015; Li et al., 2017; Li et al., 2019). Similarity algorithms are often used to measure the similarity between two strings. In this thesis, we use several similarity or distance measures that have the following definitions:

- Edit Distance: It is a way of measuring how dissimilar two strings are (Navarro 2001). Given two strings, edit distance seeks for the minimum number of insertions, deletions, and substitutions to make both strings equal.
- Cosine Similarity: It measures the cosine of the angle between two vectors projected in a multi-dimensional space. It can be derived by using the Euclidean dot product which is as follows:

$$A \cdot B = \| A \| \cdot \| B \| \cos\theta \tag{4.4}$$

As such, given two non-zero vectors of attributes, A and B, the cosine similarity is computed as follows (Malakasiotis and Androutsopoulos, 2007):

$$Sim_{Cosine} = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (4.5)$$

- Jaccard Similarity: It is computed as the number of common terms/letters over the number of all unique terms/letters in both strings (Gomaa and Fahmy, 2013).
- Jaro-Winkler Distance: Jaro distance d_j of two strings s_1 and s_2 is defined as follows (Malakasiotis and Androutsopoulos, 2007):

$$d_j(s_1, s_2) = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \quad (4.6)$$

where m is the number of matching characters and t is the number of transpositions.

- N-gram Similarity: N-gram similarity algorithms compare the n-grams from each character or word in two strings (Gomaa and Fahmy, 2013). In this thesis, n-gram similarity is computed by dividing the number of common n-grams by the number of n-grams in the shorter string.
- Jensen-Shannon Similarity: It is an information-theoretic distributional similarity measure. The basic idea behind this method is that two vectors are similar to the extent that their probability distributions are similar (Ljubesic et al., 2008). Kullback Leibler divergence is the basis to compare two probability distributions as follows:

$$KL(P|Q) = \sum_{i=1}^{|P|} P_i \cdot log \frac{P_i}{Q_i} \quad (4.7)$$

Therefore, Jensen-Shannon similarity is defined as follows (Li et al., 2018):

$$Sim_{JSD} = 1 - \frac{1}{2}\big(KL(P \parallel M) + KL(Q \parallel M)\big) \tag{4.8}$$

where $M = 1/2(P + Q)$.

- LCS Similarity: The LCS is the longest common substring of two or more given strings. Using the number of LCSs and their lengths, the similarity of two strings can be calculated as follows (Li et al., 2017):

$$Sim_{LCS} = \frac{\sum_1^{cn} Len_{lcs}(i) - cn + 1}{len(string1) + len(string2) + \sum_1^{cn} Len_{lcs}(i)} \tag{4.9}$$

where cn represents the number of LCSs and Len_{lcs} is the length of the corresponding LCS.

- LCSeq Similarity: Longest common subsequence is a sequence that presents both of the given two or more sequences. Differently from LCS, a subsequence appears in relative order, but not necessarily contiguous (Liu et al., 2017). For instance, *"xyz"*, *"xyn"*, *"ykm"*, *"xln"*, and *"xzlmn"* are subsequences of *"xyzklmn"*.
- Dynamic Time Warping Distance: It is a function to calculate the similarity between time series (Berndt and Clifford, 1994). However, it can also be employed to measure the similarity between two strings (Keogh and Pazzani, 2000).
- VMN Similarity: It is designed for matching names including one or more words. This algorithm supports the case of swapped names and the cases of partial matches (Vosecky et al., 2009).

Table 4.8. presents an overview of features based on length difference, length ratio, and similarity measures described above.

Table 4.8. Username features based on similarity measures

Feature	Definition
Length Difference	$\Delta Len = abs\big(len(u_1) - len(u_2)\big)$
Length Ratio	$Ratio = \frac{\min\big(len(u_1), len(u_2)\big)}{\max\big(len(u_1), len(u_2)\big)}$
Length of LCSeq Comparing Minimum Length	$Sim_{LCSeq} = \frac{len\big(LCSeq(u_1, u_2)\big)}{\min\big(len(u_1), len(u_2)\big)}$
Length of LCS Comparing Minimum Length	$Sim_{LCS} = \frac{len\big(LCS(u_1, u_2)\big)}{\min\big(len(u_1), len(u_2)\big)}$
Edit Distance Comparing Minimum Length	$Sim_{Edit} = \frac{edit(u_1, u_2)}{\max\big(len(u_1), len(u_2)\big)}$
Jensen-Shannon Similarity of Alphabet Distribution	$Similarity = Sim_{JSD}(u_1, u_2)$ (see Equation 4.8)
Cosine Similarity of Letter Frequencies	$Similarity = Sim_{Cosine}(u_1, u_2)$ (see Equation 4.5)
Jaccard Similarity of Letters	$Sim_{Jaccard} = \frac{len\big(letters(u_1) \cap letters(u_2)\big)}{len\big(letters(u_1) \cup letters(u_2)\big)}$
Jaro-Winkler Distance	$Sim_{Jaro} = d_j(u_1, u_2)$ (see Equation 4.6)
Bi-gram and tri-gram Similarities	$Sim_{N-gram} = \frac{\#\ of\ common}{\#\ of\ n-grams\ in\ shorter\ username}$
VMN Similarity	$Similarity = Sim_{VMN}(u_1, u_2)$ (Vosecky et al. 2009)
DTW Similarity	$Similarity = Sim_{DTW}(u_1, u_2)$ (Keogh and Pazzani, 2000)

Notice that in Table 4.8, u_1 and u_2 represent different usernames of the same individual across two different OSNs.

4.4.5. NER Features

In this thesis, we perform a NER task on the crawled Facebook data. As such, we extract various features from each of the sequence tokens. Notice that sequence represents textual content, while token represents any observed word, punctuation, or numeric value within the textual content in NER. In this thesis, we group our NER features into five main categories which are lexical, gazetteer lookup, word embeddings, contextual and conjunction, and morphological features.

(1) Lexical Features: A feature set in which each feature takes a binary value based on whether it conforms to the requested property or not. Let S and t represent an arbitrary sequence and any token of this sequence respectively. Then lexical features (or LEX in short) are as follows:

- IsFirst: t is the first token of S
- InitCap: The first letter of t is capitalized (e.g., *"Latex"*)
- Upper: t is in uppercase (e.g., *"LATEX"*)
- Lower: t is in lowercase (e.g., *"latex"*)
- MixedCaps: t is in mixed case (e.g., *"LaTeX"*)
- ContainsDigits: t contains digits (e.g., Latex25)
- SingleDigit: t is a single digit (e.g., 5)
- FourDigits: t consists of four digits (e.g., 2018)
- AllDigits: t is only comprised of digits (e.g., 01234)
- Numerical: t is a numerical value (e.g., 0.25)
- AlphaNumeric: t only contains alphanumeric characters (e.g., 2twins4ever)
- MultiDots: t includes multiple dots
- SingleChar: t is a single char
- Punctuation: t is one of the punctuation marks which are available in our list (i.e., {\;:"!?'*.})
- Roman: t is a roman numeral (e.g., II, IV, etc.)
- EndsInDot: t ends with a dot (e.g., *"Latex."*)
- LonelyInitial: t contains a single capitalized letter and ends with a dot (e.g., "G.")
- ContainsDash: t includes dash followed by any number of letters or digits (e.g., 25-02)

- ContainsApostrophe: t includes apostrophe character (e.g., *"Ali'ye"*). Notice that we consider suffixes separated by an apostrophe as a token.
- InPercent: t is in percent format (e.g., %2.5, %3,4, 5% etc.)
- InTimeFormat: t has one of two widely used time formats (i.e., 22:30, 22.30)
- InDateFormat: t has one of four date formats. Note that users can use the dot, dash, forward and backward slashes to write dates (e.g., 01.01.2020, 01-01-2020).
- ClockTerm: t is the word *"o'clock (saat)"* (Seker and Eryigit, 2017).
- PercentageSign: t is the percentage sign (i.e., %) or the word *"percent (yüzde)"* (Seker and Eryigit, 2017).
- DateTerm: t contains one of the following words: *"day (gün)"*, *"tomorrow (yarın)"*, *"night (gece)"*.

Additionally, we extract a few other features after applying lowercasing on the token to prevent an increase of feature space. These features are also as follows (Özkaya and Diri, 2011; Demir and Özgür, 2014; Okur et al., 2018):

- Prefixes: Subsequences (if exists) which are containing the first three and four characters of t
- Suffixes: Subsequences (if exists) which include the last one to four characters of t
- N-grams: Character level 2, 3, and 4 grams of t.

(2) Gazetteer Lookup Features: Gazetteer lookup (or GLU in short) features have a value of 1 or 0 depending on whether a token t matches in the related gazetteer or not. In this thesis, we extract GLU features based on our gazetteers which are described in Section 3.5.2. Using these gazetteers, we extracted seven

different features which are indicating whether the current token is a location, organization, percent, day of the week, and so on.

Notice that activity contents in Facebook often contain grammatical errors and users do not use the apostrophe for proper nouns (e.g., *"ankarada (in Ankara)"*, *"emrenin (Emre's)"*) which makes it too hard to match entities in gazetteers. Therefore, in this thesis, we use the condition as a binary feature that considers whether the original token t or its suggested word s matches in the related gazetteer. We apply this strategy only for Person and Location gazetteers. To get s for mismatched t, we use a procedure which has the following rules:

- The Levenshtein distance (1966) between t and s must be greater than 0.8,
- The length difference between t and s must be less than 4,
- The length of s must be equal or greater than 3,
- t must contain the s that has to be a proper noun.

Table 4.9 presents a non-exhaustive list of suggested tokens for tokens that mismatch with our Person and Location gazetteers.

Table 4.9. An incomplete list of suggested tokens by our strategy instead of mismatched tokens

Token		**Token**	
Original (t)	**Suggested (s)**	**Original (t)**	**Suggested (s)**
iremin	İrem	Trabzonda	Trabzon
nebahatin	Nebahat	Habibesu	Habibe
Kübram	Kübra	istanbularda	İstanbul
mehmetende	Mehmet	salican	Alican
iskenderuna	İskenderun	duygucan	Duygu
Genes	Enes	Cemu	Cem
bursada	Bursa	inegölüler	İnegöl

(3) Word Embedding Features: We use word embeddings of tokens in our NER task as used by Seok et al. (2016). We obtained word embeddings (or VEC features in short) by using the word2vec model with skip-gram architecture with the help of the DL4J library. To achieve this goal, we trained the word2vec model on our hybrid dataset (see Section 3.3.2.2) with different parameter settings where layer size differs from 100 to 300, the window size is set to 5, and minimum word frequency is taken as 1. We do not remove punctuation marks from our training corpus and we applied stemming as word embeddings provide better results in this case for NER task (Ertopçu et al., 2017). If the word2vec model does not have a vector representation for any token, we use zero vector in layer size dimension instead.

(4) Morphological Features: As Turkish is a morphologically rich language, using morphemes as features can improve NER performance. Therefore, we use morphological features as well as other features as done by Yeniterzi (2011), Şeker and Eryiğit (2012), and Kalender and Korkmaz (2017).

```
mustafa + Noun + Prop + A3sg
kemal kemal+ Noun + Prop + A3sg
, + Punc
mersin + Noun + A3sg
gezi + Noun + A3sg + P3sg + Loc^DB + Verb +
    Zero^DB + Adv + While
... + Punc
```

Figure 4.11. An example morphological analysis

Morphological analysis on a token may result in more than one possible parsing due to ambiguity. As such, we perform morphological analyses at the sequence level to get the best parsing. An example sentence level morphological analysis for a sequence *"Mustafa Kemal, Mersin gezisindeyken… (while Mustafa Kemal is on a trip to Mersin…)"* is shown in Figure 4.11. In this thesis, we use the following morphological features in our NER task similar to Seker and Eryigit (2017):

- STM: Stem (i.e., root) of the token (e.g., *"gezi"*),
- POS: Part of speech tag of the token (e.g., *"Noun"*),
- NCS: Noun cases of a token. This feature takes, if exists, one of the following values: Accusative (Acc), Dative (Dat), Ablative (Abl), Locative (Loc), Genitive (Gen), Instrumental (Ins), and Equative (Equ),
- PRP: whether token includes *"Prop"* tag or not, and
- INF: All inflectional tags following part of speech tag.

(5) Contextual and Conjunction Features: Context information around the current word may help to distinguish NEs. We use contextual information of the current token and their conjunctions in a sliding window. We use a window of width 2 and extract contextual features for the current token as following:

- CXW: tokens before and after the current token (Demir and Özgür, 2014; Okur et al., 2018),
- CXP: contextual information of part of speech tags.

Table 4.10 presents an example extraction of contextual features for the token *"C"* in our example sequence.

Table 4.10. An example extraction of contextual features.

Features	**Context Tokens**					**Conjunctions**
	Position in Sliding Window					
	@-2	**@-1**	**@0**	**@1**	**@2**	
CXW	A	B	C	D	E	B@-1&C&D@1, A@-2&C&E@2
CXP	Noun	Noun	Noun	Noun	Punc	Noun@-1&Noun&Noun@1, Noun@-2&Noun&Punc@2

4.5. Feature Selection

In this thesis, some of our tasks utilize features that are extracted from textual content generated by users. In such a case, we are dealing with a huge feature space. Therefore, we use feature selection which is the process of selecting the best K features as a subset of the features occurring in the training set and using this subset as features in the related task. Feature selection makes classifier training more efficient by decreasing the high dimensionality of effective vocabulary. Additionally, feature selection often increases classification accuracy by selecting the most discriminative features (Thabtah et al., 2009).

There are many feature selection methods such as document frequency, information gain, chi-square, mutual information, and so on. In this thesis, we used chi-square which is one of the well-known and most effective (Moh'd A Mesleh, 2007) methods to reduce feature space in our tasks using textual features. Chi-square testing is a well-known discrete data hypothesis testing method from statistics that evaluates the correlation between two variables and determines whether they are independent or correlated (Snedecor and Cochran, 1989). Applying the test for independence over a population of subjects determines if they are positively correlated or not (Thabtah et al., 2009).

Using a two-way contingency table (see Table 4.11) of a feature f and a category c, chi-square value for each feature f in a category c can be computed by the following equation (Moh'd A Mesleh, 2007; Thabtah et al., 2009; Jin et al., 2015):

$$\chi^2(f,c) = \frac{N(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \tag{4.10}$$

Table 4.11. Variables in two-way contingency table for chi-square statistics

Property	**Belong to category *c***	**Do not belong to category *c***	**Sum**
# of instances with feature ***f***	A	B	A + B
# of instances without feature ***f***	C	D	C + D
Sum	A + C	B + D	N = A + B+ C + D

If $\chi^2(f,c) = 0$, the feature f and category c are independent; therefore, the feature f does not contain any category information. Otherwise, the greater the value of the $\chi^2(f,c)$, the more category information the feature f owns (Jin et al., 2015). The score of any feature f on the training set is obtained by averaging or maximizing its category-specific scores as follows:

$$\chi^2_{avg}(f) = \sum_{k=1}^{|c|} p(c_k)\chi^2(f,c_k) \tag{4.11}$$

$$\chi^2_{max}(f) = max_{k=1}^{|c|}\chi^2(f,c_k) \tag{4.12}$$

In this thesis, we employ chi-square for feature selection by using *scikit-learn* implementation.

4.6. Sample Representation

One of the main conditions to train a model is to transform the data into a proper structure that the learning model can deal with. On the ML side, this is often accomplished by transforming each instance into a numerical representation in the form of a vector, where each dimension of the vector corresponds to the value of one of the related features at hand. In this thesis, we use this strategy to convert our dataset into a classification ready structure for our ML-based tasks.

Specifically, in some of our tasks, we use user content to improve the accuracy of our ML models. In such cases, we obtained numerical representation of a text by associating each feature (e.g., word, n-gram, etc.) with a weight value that is often proportional to its frequency on the text (Coban et al., 2018). To assign feature weights, we employ different weighting schemes including Binary, TF, TF*IDF, and Fuzzy which are formulated as follows (Salton and Buckley, 1988; Coban and Ozel., 2018; Uzel et al., 2018):

$$W_{TF}(f,s) = TF(f,s) \tag{4.13}$$

$$W_{Binary(f,s)} = \begin{cases} 1, & W_{TF}(f,s) \geq 1 \\ 0, & otherwise \end{cases} \tag{4.14}$$

$$W_{IDF}(f,s) = log\ \frac{|S|}{|s_i \in S : f \in s_i|} \tag{4.15}$$

$$W_{TF*IDF}(f,s) = W_{TF}(f,s) * W_{IDF}(f,s) \tag{4.16}$$

$$W_{Fuzzy}(f,s) = 1 - \prod_{u \in s}\big(1 - c(f,u)\big) \tag{4.17}$$

where f and s represent any term (i.e., feature) and document (sample or instance) from related document collection S respectively, while u corresponds to observed feature other than feature f. TF is the raw term frequency of feature f in sample s and $c(f,u)$ is obtained as follows:

$$c(f,u) = \frac{\#\ of\ samples\ which\ contain\ both\ features\ f\ and\ u}{\#\ of\ samples\ which\ contain\ at\ least\ one\ of\ features\ f\ and\ u} \tag{4.18}$$

Notice that tf*idf is the most commonly used weighting scheme among traditional weighting methods and Fuzzy weighting considers the non-existence of a term while computing weight value (Uzel et al., 2018).

Another weighting method used in this thesis is the self-information model that uses each feature's self-information as its weight (Wang et al., 2016). The self-information of a specific message m is defined as $I(m) = -logPr(m)$. Therefore, $w(f) = I(f) = -logPr\big(W_{Binary}(f,\cdot) = 1\big)$. As it is not possible to get the true value of self-information, it is estimated on samples by using the following way:

$$W_{Self-Info}(f,s) = \frac{|s \in S | W_{Binary}(f,s)=1|}{|S|} \tag{4.19}$$

Unlike the ML side, representing instances in the form of a vector is not enough for DL algorithms which require to feed the input data in the form of a matrix representation. On the DL side, each instance is often represented by a matrix whose size varies depending on the focused task. For instance, in TC and SA fields, a text instance is often transformed into a matrix with the help of word embeddings. In this kind of representation, the number of rows and columns of an instance matrix is equal to the number of unique features and layer size of word embedding models respectively (see Figure 4.13). DL algorithms do not depend on hand-crafted features as in the ML side and they instead perform hierarchical feature extraction. In this thesis, we used this strategy to prepare our data for our tasks running DL algorithms.

4.7. Handling Class Imbalance

A dataset is imbalanced if the distribution of instances across the classes is not equal. Imbalanced datasets can be observed in many practical application domains such as fraud detection, diagnosing rare diseases, TC, and so on (Nguyen et al., 2011). For this kind of data, the construction of classifiers poses a challenge

for predictive modeling as most of the ML methods are designed around the assumption of an equal number of examples for each class.

Figure 4.12. Data level methods for imbalanced learning

Therefore, three different approaches are devoted to solving the imbalance problem: (i) *data level methods* aim to balance the data by adding new instances into the minority class (i.e., oversampling), or removing many instances from the majority class (i.e., under-sampling), (ii) *algorithm level methods* modify existing classifiers to be more suitable to handle imbalanced datasets, (iii) *hybrid methods*, on the other hand, combine the advantages of the previous two groups (He et al., 2008; Nguyen et al., 2011; Krawczyk, 2016). In this thesis, we use the *data level methods* to handle imbalanced datasets as depicted in Figure 4.12 (Mohammed et al., 2020).

Simple oversampling and under-sampling methods use a random approach to add or remove target instances. However, this often leads to the removal of important instances or adding meaningless new instances (Krawczyk, 2016). Therefore, more advanced data-level methods were also proposed to maintain structure groups and/or generate new data according to underlying distributions (Chawla et al., 2002). The family of well-known oversampling methods includes Random Oversampling, SMOTE (Chawla et al., 2002), Borderline SMOTE (Han et al., 2005), and ADASYN (He et al., 2008). On the other hand, the family of well-known under-sampling methods includes Random Under-sampling, Condensed Nearest Neighbor (Hart, 1968), Tomek links (Tomek, 2010), and Edited Nearest

Neighbors (Wilson, 1972). Besides, it is possible to almost combine all over and under sampling methods. Examples of popular combinations also include SMOTE-Random Under sampling, SMOTE-Tomek links, and SMOTE-Edited Nearest Neighbors.

SMOTE randomly creates synthetic minority class instances on the line segments which are joining each minority class with a number of its neighbors (Chawla et al., 2002; Nguyen et al., 2011). Tomek links (Tomek, 2010) can be defined as a pair of minimally distanced nearest neighbors of opposite classes. If two instances form a Tomek link, then either one of these instances is noise or both are near a border. As such, Tomek links can be used to clean up unwanted overlapping between classes after synthetic sampling where all Tomek links are removed until all minimally distanced nearest pairs are of the same class (He and Garcia, 2009). In this thesis, we employ oversampling and under-sampling methods with the help of the imbalanced-learn (Lemaitre et al., 2017) library for required tasks.

4.8. Classification

In this thesis, crawled data and its subsets are used to perform analyses in different aspects which commonly require processing text-based data. Therefore, our analysis mostly depends on TC techniques.

4.8.1. Naïve Model

Naïve model is a simple classification model that does not use any sophistication and typically makes a random prediction. The performance of the naïve classifier provides a lower bound on the expected performance of all other learning models. In this thesis, naïve classifier is employed with the help of the Dummy Classifier which is implemented in the scikit-learn library (Pedregosa et

al., 2011). This classifier uses five different strategies to make its predictions. These strategies are as follows:

- Stratified: predicts by respecting the class distribution in the training set.
- Most frequent: always predicts the most frequent label in the training set.
- Prior: always predicts the class which maximizes the class prior.
- Uniform: makes uniform predictions at random.
- Constant: always predicts a constant label given by the user.

The naïve classification model is useful as a simple baseline to compare with other (real) classifiers.

4.8.2. Learning Models

In the learning phase, well-known ML and DL algorithms are employed on classification-ready data.

4.8.2.1. Machine Learning

ML is a field of developing algorithms that can access data and use it to learn to emulate human intelligence. It has been applied in diverse fields such as pattern recognition, computer vision, and so on (El Naqa and Murphy, 2015; Jordan and Mitchell, 2015). ML algorithms are often divided according to the nature of the data labeling into supervised, unsupervised, and semi-supervised. Supervised learning needs external assistance for labels of training data that can be predicted or classified.

Unsupervised learning learns features from unlabeled data to recognize previously unseen data. Semi-supervised learning, on the other hand, is a technique that combines the power of both supervised and unsupervised learning (Dey, 2016). In different subtasks of this thesis, supervised learning is applied where the

training data takes a form of a collection of (x, y) pairs and the goal is to produce a prediction y^* for a test instance x^*. Supervised learning algorithms generally make their predictions via a learned mapping function $f(x)$, that gives an output y for each input x (Jordan and Mitchell, 2015). There are different forms of the mapping function including DT, LR, SVM, NB, NBM, and so on. Following subheadings give brief descriptions of such functions used in this thesis.

(1) DT (a.k.a J48 or C4.5): It is a type of tree that groups attributes by sorting them based on their values. Each tree is composed of nodes and branches in which each node represents attributes in a group to be classified, whereas each branch represents a value that a node can take (Dey, 2015).

(2) NB: It is a simple probabilistic classifier based on Bayes' theorem with strong independence assumptions (Khan et al., 2010). It works well on numeric and textual data and easy to both implement and computation when compared to other classifiers (Kibriya et al., 2004). Given $X = (x_1, x_2, \dots x_m)$ as values of m features and $Y = (y_1, y_2, \dots, y_n)$ as values of class attribute, class of instance X can be calculated by the following measure (Zafarani et al., 2014):

$$y^* = arg\max_{y_i} \frac{\left(\prod_{j=1}^{m} P(x_j|y_i)\right) \cdot P(y_i)}{P(X)} \tag{4.20}$$

Notice that NB assumes that $P(X)$ is independent of y_i and features are conditionally independent.

(3) NBM: This classifier uses the Bayesian learning perspective and assumes that feature distributions in samples are generated by a specific parametric model. The parameters are estimated from the training data. NBM is such a parametric model that is commonly used in TC (Su et al., 2011). Given $X = (x_1, x_2, \dots x_m)$ as values of m features and $Y = (y_1, y_2, \dots, y_n)$ as values of class attribute, class of instance X can be calculated as follows:

$$y^* = arg\ \max_{y_i} \frac{P(y_i) \prod_{j=1}^{m} P(x_j|y_i)^{f_i}}{P(X)} \quad (4.21)$$

where f_i is the number of occurrences of the feature x_i in a sample X, $P(x_j|y_i)$ is the conditional probability that a feature x_j may observed in sample X in class y_i, and m is the number of unique features in sample X. On the other hand, $P(y_i)$ is the prior class probability that a sample with the class attribute y_i may be observed among all samples (Kibriya et al., 2004; Su et al., 2011).

(4) SVM: SVM is a linear classifier that is proposed based on statistical learning theory and attempts to learn decision boundary (i.e., maximum margin hyperplane) that provides optimal separation of classes in data (Vapnik, 1995). This decision boundary is usually learned by a small subset of instances which are called support vectors.

If two classes are not linearly separable, the SVM tries to find a non-linear decision boundary by transforming the data with the help of a kernel function. The linear decision boundary which is learned from the transformed data can represent a non-linear partitioning of the original feature space (Khoshgoftaar et al., 2007). Notice that SVM was initially designed for binary classification problems, but it can also be employed in multiclass problems using an appropriate strategy such as the one against one and one against the rest (Vapnik, 1995; Cristianini and Shawe-Taylor, 2000).

(5) SMO: It is the implementation of sequential minimal optimization for training the SVM learner in Weka. Two changes to the default parameters are the complexity constant *'c'* was changed from 1 to 5, and the *'buildLogisticModels'* parameter which allows proper probability estimates to be obtained, was set to *'true'* (Witten et al; 2005; Khoshgoftaar et al., 2007). SMO uses linear kernel by default.

(6) LR: It is also known as *logit* or *MaxEnt* classifier and provides a probabilistic view of regression. LR is a mathematical modeling approach that can

be used to describe the relationship of several X's to a dichotomous dependent variable Y (Kleinbaum et al., 2002). Let us consider a binary classification task with X is a feature vector and Y is a class attribute. LR aims to find probability p such that

$$P(Y = 1|X) = p \tag{4.22}$$

Assuming the probability p depends on X, linear regression can be used to approximate p;

$$p = \beta X \tag{4.23}$$

where β is a vector of coefficients. Here, βX can take unbounded values, while p must be in the range [0, 1] (Zafarani et al., 2014). Therefore, the logit function is used to change the range from [0, 1] to [-∞, +∞] as following (Bewick et al., 2005):

$$logit(p) = ln\frac{p}{1-p} \tag{4.24}$$

The transformed p can be approximated using the following linear function:

$$logit(p) = \beta X \tag{4.25}$$

Combining the previous two equations given above, p is obtained as given in the following equation (Kleinbaum et al., 2002; Zafarani et al., 2014):

$$p = \frac{e^{\beta X}}{e^{\beta X}+1} \tag{4.26}$$

where β's are estimated using MLE that entails finding value(s) of the parameter(s) that provides the maximum likelihood. Two types of approaches are used to employ LR in multi-class tasks. The simpler and popular one is taking multi-classification as a series of binary classification including one-versus-all and one-versus-one (Do and Poulet, 2015).

(7) KNN (a.k.a IBk): It assigns the test instance to the class having most instances among the k neighbors of the test. All neighbors have an equal vote and the class having the maximum number of voters among the k neighbors is chosen (Alpaydın, 2014).

(8) RF: The random forest classifier creates an ensemble of decision trees where each tree is generated by using a random vector that is sampled independently from the input data (Breiman, 1999). RF can be considered as a special case of bagging which is an ensemble learning algorithm that combines predictions of several classifiers by using bootstrapping. Bootstrapping is one of randomness in RF as it is a process of random sampling with replacement from the training data (Khoshgoftaar et al., 2007). Each tree is inducted by using bootstrapping and approximately 2/3 of the instances in the training data are randomly selected to build the related tree. Another source of randomness stems from randomly selecting a subset of attributes to consider at each node of each tree.

In the RF learner, each tree makes its prediction by casting a unit vote for the most popular class to classify an input vector. Any sample is classified by taking the most popular voted class from all the decision trees in the forest (Breiman, 1999).

(9) CRF: It is a probabilistic model (Lafferty et al., 2001) which is proposed to segment and label sequence data. In statistical learning, generative models (e.g., Hidden Markov models, stochastic grammars) are widely used in many different problems in which there is a need to segment and label sequences. The generative models attempt to define a joint probability distribution $p(y, x)$ over input and outputs and they have some limitations making to construct a

distribution difficult especially when the dimensionality of x large and the features have complex dependencies (Sutton and McCallum, 2012).

To overcome such limitations, the CRF algorithm is proposed as a discriminative approach in which the conditional distribution $p(y, x)$ is constructed directly that is all that needed for classification (Sutton and McCallum, 2012). In this thesis, linear-chain CRF is used that has the following formal definition (Lafferty et al., 2001; Sutton and McCallum, 2012; Sutton and McCallum, 2012):

$$P_\theta(y|x) = \frac{1}{Z_x}\exp\left(\sum_{t=1}^{T}\sum_{k=1}^{N}\lambda_k f_k(y_{t-1}, y_t, x_t)\right) \quad (4.27)$$

where;

- $x = x_1, \dots, x_T$ is a sequence of input,
- $y = y_1, \dots, y_T$ is a sequence of predicted labels,
- $f_k(y_{t-1}, y_t, x_t)$ is a feature function for the properties of transition from state y_{t-1} to y_t,
- λ_k is an optimized weight parameter for feature f_k,
- T is the length of input sequence X,
- N is the number of feature functions, and
- Z_x is normalization constant that makes the sum of the probability of all state sequences to one.

For an input sequence x, the most probable label is calculated as following:

$$y^* = arg\max_y P_\theta(y|x) \quad (4.28)$$

4.8.2.2. Deep Learning

As a subfield of ML, DL is concerned with algorithms that are inspired by how the human brain works. In this thesis, two architectures are employed on the

DL side, which are mainly used in categorizing textual data (Hassan and Mahmood, 2017; Uysal and Murphy, 2017).

(1) CNN: CNNs are composed of three main types of layers which are convolutional layer, pooling layer, and fully-connected layer. They have a loss function (e.g., softmax) on the last fully-connected layer as well (Uysal and Murphey, 2017). A general CNN architecture for NLP tasks is depicted in Figure 4.13 (Kim, 2014).

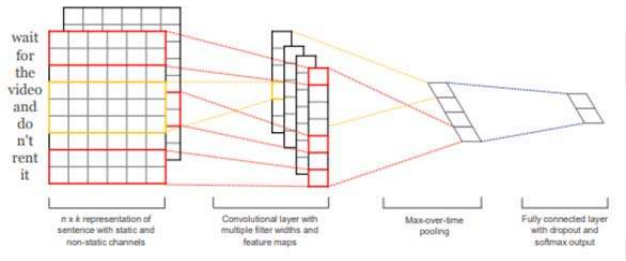


Figure 4.13. CNN architecture with two channels.

To build this model on text data, each text with a length of n is represented as following (Kim, 2014):

$$x_{1:n} = x_1 \oplus x_2 \oplus \dots \oplus x_n \tag{4.29}$$

where $\oplus$ is the concatenation operator. Each convolution step involves a filter $w \in \mathbb{R}^{hk}$, in a window of h words to generate a new feature. In a more general way, the convolution operation can be formulated as follows:

$$c_i = f(w \cdot x_{i:i+h-1} + b) \tag{4.30}$$

where c_i is generated feature from a window of words $x_{i:i+h-1}$, $b \in \mathbb{R}$ is the bias term, and f is a non-linear activation function like ReLU. A new feature map of $c = [c_1, c_2, \dots, c_{n-h+1}]$ is obtained by applying this operation to each possible

window of words in the sentence $x_{1:h}, x_{2:h}, \dots, x_{n-h+1:n}$. Then, a max-over-time pooling is applied over the feature map c (with $c \in \mathbb{R}^{n-h+1}$) to get the maximum value (i.e., $\hat{c} = \max c$) as the feature for the related filter (Kim, 2014).

(2) RNN: Unlike traditional NNs, inputs and outputs are not independent of each other in RNNs. They are called *recurrent* as they perform the same task for every element of sequence with the output being dependent on previous computations (Stojanovski et al., 2016). This means that RNN can use an internal memory that is like remembering information about what has been processed so far (Zhang et al., 2018). Figure 4.14 shows basic RNN architecture before and after unfolding where x_t and h_t are input and hidden state vectors respectively at time step t (Zhang et al., 2018).

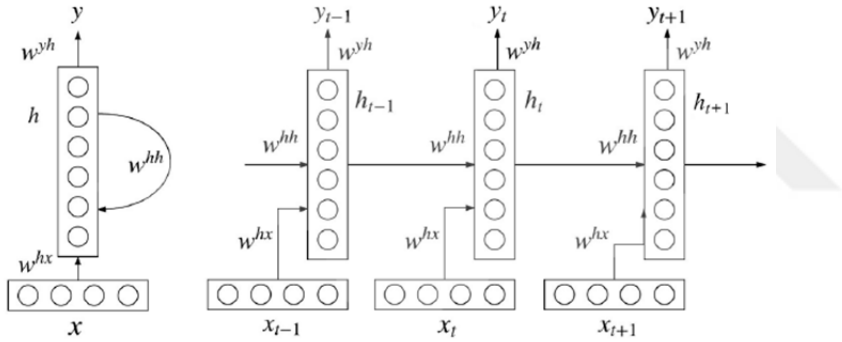


Figure 4.14. Basic RNN architecture before (left) and after (right) the unfolding

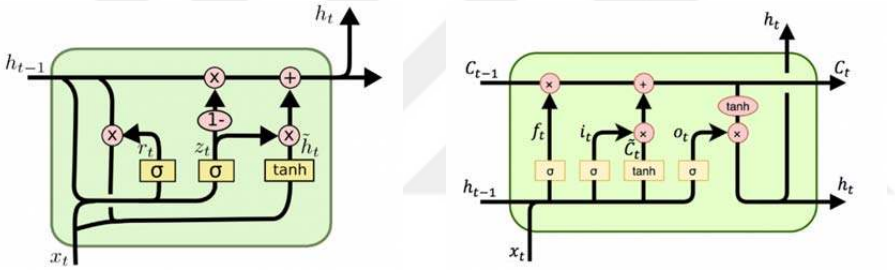
The forward propagation task in RNN can be described by using the following formulas (Zhang et al., 2018; Rysbek, 2019):

$$h_t = f\left(w^{hx}x_t + w^{hh}h_{t-1} + b_h\right) \tag{4.31}$$

$$y_t = g\left(w^{yh}h_t + b_y\right) \tag{4.32}$$

where w^{hx}, w^{yh}, and w^{hh} are learned and shared weight matrices across all steps, whereas f and g are activation functions and b_h and b_y are bias terms. This simple RNN suffers from the exploding and vanishing gradient problem during the backpropagation training stage (Karpathy et al., 2015; Cliche, 2017; Zhang et al., 2018; Rysbek, 2019).

As such, more sophisticated architectures of the RNN are proposed to overcome this challenge. Among these solutions, the commonly used architectures are LSTM and GRU (Karpathy et al., 2015; Santur, 2019). Figure 4.15 (Dprogrammer, 2019) shows LSTM (with input, forget, and output gates) and GRU (with the update and reset gates) cells that are capable of remembering information for a long time (Rysbek, 2019).



(a) GRU (b) LSTM

Figure 4.15. Architectures of the LSTM and GRU cells/units.

The hidden state of the LSTM is computed by the following equations (Cliche, 2017; Rysbek, 2019):

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \tag{4.33}$$

$$f_t = \sigma\big(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f\big) \tag{4.34}$$

$$\tilde{C}_t = tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \tag{4.35}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4.36}$$

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \tag{4.37}$$

$$h_t = o_t \times \tanh(C_t) \tag{4.38}$$

where i_t, f_t, and o_t are input, forget, and output gates respectively. x_t and h_t are input vector and hidden layer values of the cell at time step t. The C_t, $\tilde{C}_t$, and C_{t-1} represent current, candidate, and previous cell states respectively, while σ (i.e., sigmoid) and $tanh$ are activation functions. Similarly, how the GRU cell is updated at each time step t is computed by the following equations (Karpathy et al., 2015; Dprogrammer, 2019; Santur, 2019):

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z) \tag{4.39}$$

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \tag{4.40}$$

$$\tilde{h}_t = \sigma(W_h \cdot x_t + U_h \cdot (r_t \times h_{t-1}) + b_h) \tag{4.41}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t + \tilde{h}_t \tag{4.42}$$

where r_t and z_t represent vectors of reset and update gates differently from the variables used above. Notice that in all equations given above $\times$ represents the element-wise product of two matrices/vectors.

Unidirectional RNNs can also be wrapped in bidirectional layers by allowing bidirectional connections in the hidden layer (Zhang et al., 2018). In bidirectional RNNs (BRNN), the forward RNN reads the input sequences from

start to end, while the backward RNN reads it from end to start (Lipton et al., 2015; Rysbek, 2019).

(3) RNN-CNN and CNN-RNN: It is a common practice to combine RNN and CNN algorithms to improve the accuracy of the related task (Sosa, 2017; Farha and Magdy, 2019). As such, in this thesis, we also combine the best variants of these two well-known DL algorithms to perform SA of Facebook data. Notice that this selection is based on our results (see Section 5.1.3) and we create two model variants which are CNN-GRU and GRU-CNN that are depicted in Figure E.3 and Figure E.4 respectively. The intuition behind this combination is that the first part of the model will extract features, whereas the second part will be able to learn about input text (Sosa, 2017; Farha and Magdy, 2019).

4.8.3. Performance Measuring

This subsection describes methods that are used to validate and evaluate the performance of used models for different tasks.

4.8.3.1. Model Validation

In classification problems, datasets are divided into three disjoint subsets, namely, training set, validation set, and testing set (Zhang and Philip, 2014). Many different methods have been proposed to split the data set into these subsets, but cross-validation is one of the most commonly used strategies among others especially when the dataset is small. There are different variations of cross-validation including k-fold and leave-one-out (Kohavi, 1995).

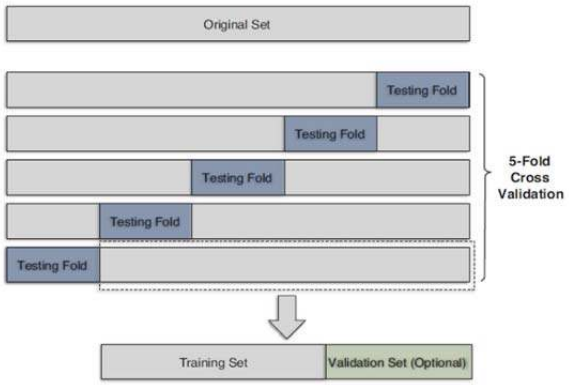


Figure 4.16. An example of 5-fold cross-validation

As shown in Figure 4.16, in k-fold cross-validation, the dataset is divided into k equal-sized subsets, where each subset can be picked as a test set while the remaining $k-1$ subsets are used as the training set (Alpaydin, 2014; Zhang and Philip, 2014). This process runs for k times and each time a different subset is used as the test set. Similarly, leave-one-out picks one single instance in each round, but it suffers from the high time cost problem when the dataset is large.

4.8.3.2. Evaluation Metrics

In classification tasks, the performance of a model is evaluated by using four different parameters which are obtained from the confusion matrix. To simplify, let $y = [y_1, y_2, \dots, y_n]^T$ and $\hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n]^T$ be true and predicted (i.e., by a binary classification model) labels of n data instances respectively. Depending on these true and predicted labels, data instances can be grouped into four main categories which are used to build a confusion matrix (Zhang and Philip, 2014).

Table 4.12. Confusion matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP (True Positive)	FN (False Negative)
Actual Negative	FP (False Positive)	TN (True Negative)

Table 4.12 presents a confusion matrix with four categories of instances for a binary classification (Van Asch, 2013). These four categories, namely, TP, FN, FP, and TN have the following definitions (Coban et al., 2018):

- TP: the number of correctly classified positive instances
- FN: the number of incorrectly classified positive instances
- FP: the number of incorrectly classified negative instances
- TN: the number of correctly classified negative instances.

Based on these definitions from the confusion matrix, well-known evaluation metrics can be computed for classification tasks. These metrics are precision (P), recall (R), accuracy (Acc), and F-measure (F-score or F1) which have the following formal definitions (Alpaydin, 2014):

$$Acc = (TP + TN)/(TP + FN + FP + TN) \tag{4.43}$$

$$P = TP/(TP + FP) \tag{4.44}$$

$$R = TP/(TP + FN) \tag{4.45}$$

$$F - score(\beta) = \frac{(1+\beta^2)\times P\times R}{\beta^2\times P+R} \tag{4.46}$$

In Equation 4.46, β is mostly taken to be 1 and in this case, it is also named as F1 score. Among these metrics, the Acc works well for class balanced scenarios where the number of positive and negative instances are equal or close, but it suffers from a serious problem in evaluating the model's performance for class imbalanced scenarios (Zhang and Philip, 2014).

Averaging these measures can give a view on general results when classification is performed on multiple-class data. Micro-averaged and macro-averaged measures are often used in literature to refer to the averaged results. Let $L = \lambda_j : j = 1 \dots q$ is a set of labels and $B(TP, TN, FP, FN)$ is a binary evaluation measure based on the four categories of the confusion matrix. Then, macro-averaged and micro-averaged results of any measure can be obtained with the help of the following equations:

$$B_{macro} = \frac{1}{q} \sum_{\lambda=1}^{q} B(TP_\lambda, TN_\lambda, FP_\lambda, FN_\lambda) \quad (4.47)$$

$$B_{micro} = \frac{1}{q} \sum_{\lambda=1}^{q} \frac{c_\lambda}{n} B(TP_\lambda, TN_\lambda, FP_\lambda, FN_\lambda) \quad (4.48)$$

where c_λ is the number of instances in the training set with class label λ and n is the total number of instances in the training set (Van Asch, 2013). These metrics are employed in other classification related tasks as well. NER is such a task in which performance evaluation of a NER system is performed based on P, R, and F1 score. However, unlike classical classification tasks, the values of these metrics are obtained by using different criteria as the boundaries and categories are ambiguous in the NER task. The MUC and CoNLL metrics are commonly used techniques in related conferences to rank the capability of systems in terms of finding boundaries of entities with their correct types (Nadeau and Sekine, 2007).

The MUC metric has partial matching in its nature and it is the average F-measure of MUC TEXT and MUC TYPE. The MUC TYPE evaluates the performance of assigning the correct NE type to each token without considering whether the boundaries are detected correctly, whereas the MUC TEXT only evaluates NE boundaries without looking if the correct type is assigned or not (Jiang et al., 2016).

In the MUC calculation, three values are kept for both TYPE and TEXT which are COR (i.e., the number of correct answers), ACT (i.e., the number of actual system guesses), and POS (i.e., the number of possible entities in sequence). Then MUC score is micro-averaged F1 (see Equation 4.48.) of P and R where P is being the percentage of NEs found by the system which are correct and R is being the percentage of NEs present in the sequence which are found by the system (Nadeau and Sekine, 2007; Eken 2015). In the NER task, P and R values are obtained by the following equations:

$$P = COR/ACT \tag{4.49}$$

$$R = COR/POS \tag{4.50}$$

CoNLL metric, on the other hand, requires an exact matching criterion and evaluates an assignment to be correct only when the type and boundary of an entity are determined correctly. CoNLL is also micro-averaged F1 of P and R values. Table B.1 presents an example labeling for an arbitrary sequence "*Robert and John Briggs Jr contacted Wonderful Stockbrokers Inc in New York to sell their shares in Acme*" and true predictions in terms of MUC and CoNLL metrics. In this example, COR = 4 (2 TYPE + 2 TEXT), ACT = 10 (5 TYPE + 5 TEXT), and POS = 10 (5 TYPE + 5 TEXT).

The P is therefore 40% and R is 40% and the MUC score is 40%. On the other hand, the CoNLL score is 20% as P and R have equal percentages of 20%. This is because there are 5 entities in sequence and this makes 5 predictions from which the only one exactly matches within the sequence (Nadeau and Sekine, 2007). As seen from this toy example, CoNLL is stricter than MUC as it concentrates on finding phrase-level NEs, while MUC has the advantage of taking into all possible types of errors.

4.9. Privacy Risk Scoring

In this section, we present risk scoring methods and how we computed social tie strengths between users in measuring the risk of privacy.

4.9.1. Privacy Scoring Methods

The first study that provides a mathematical methodology for privacy measuring over OSNs is proposed by Liu and Terzi (2010). Their study is based on an $n \times N$ response matrix R which stores privacy levels of all N users for all n items and $R(i,j)$ represents privacy setting/level of user j for item i. The matrix R can be constructed in two different ways, namely, dichotomous and polytomous. In dichotomous case, each cell in R takes values in $\{0,1\}$, while it takes any positive integer values in $\{0,1,\dots,l\}$ in polytomous case. $R(i,j)=0$ means that user j keeps item i private in both dichotomous and polytomous cases. $R(i,j)=1$ in the dichotomous case means that user j makes item i public. On the other hand, in polytomous case, $R(i,j)=k$ with $k \geq 1$ means that user j reveals item i to users who are at most k links away from user j. Table 4.13 presents a sample R matrix in dichotomous case (Pensa et al., 2019; Coban et al., 2020b) where items represent a set of profile items revealed by OSN users such as age, gender, political view, and phone number.

Table 4.13. A sample R matrix in dichotomous case

User	Items (Topics)					
	A	B	C	D	E	F
u_1	1	0	1	1	0	0
u_2	0	1	0	1	0	0
u_3	1	1	0	0	1	0
u_4	1	0	0	1	0	0

Liu and Terzi proposed methods to measure the privacy of OSN users based on the R matrix. In the dichotomous case, the privacy score of an arbitrary user j is calculated as follows:

$$PR(j) = \sum_{i=1}^{n} \beta_i \times V(i,j) \quad (4.51)$$

where β_i represents the sensitivity of item i, while $V(i,j)$ stands for visibility of item i with respect to the user j. In the polytomous case, this calculation is as follows:

$$PR(j) = \sum_{i=1}^{n} \sum_{k=0}^{l} \beta_{ik} \times V(i,j,k) \quad (4.52)$$

where β_{ik} is the sensitivity of item i with respect to level k and $V(i,j,k)$ is the visibility of item i for user j at level k.

In the equations above, privacy scores can be computed using both observed and true visibility of items. In dichotomous case, observed visibility of item i for user j equals to $V(i,j) = \mathrm{I}_{(R(i,j)=1)}$, where $\mathrm{I}_{condition}$ is a simple function that returns 1 when *"condition"* is true. True visibility is also computed as $V(i,j) = P_{ij}$, where $P_{ij} = Prob\{R(i,j) = 1\}$.

In polytomous case, the observed visibility is calculated as $V(i,j,k) = \mathrm{I}_{(R(i,j)=k)} \times k$, while the true visibility is computed as $V(i,j,k) = P_{ijk} \times k$, where $P_{ijk} = Prob\{R(i,j) = k\}$ (Liu and Terzi, 2010). The authors use true visibility to compute privacy scores based on the IRT where the goal is to measure the ability of an examinee, the difficulty of any question, and the probability of an examinee to correctly answer a given question. If the response of an examinee j to a question q_i is in the dichotomous case (i.e., correct or wrong), then the probability that j gives the correct answer is obtained as follows:

$$P_{ij} = \frac{1}{1+e^{-\alpha_i(\theta_j-\beta_i)}} \quad (4.53)$$

where θ_i is the ability of examinee j, while α_i and β_i parameters are used to represent discrimination power and difficulty of a question q_i (Liu and Terzi, 2010).

The authors use Equation 4.53 to exploit the IRT in privacy score computation where they consider each user as an examinee and each profile item as a question. They use Newton-Raphson (Ypma, 1995) and Expectation-Maximization algorithms to estimate parameters α_i and β_i for n items ($1 \leq i \leq n$) and θ_j for N users ($1 \leq j \leq N$) in Equation 4.53. For polytomous case, the authors perform these calculations by transforming the response matrix R into $(l+1)$ dichotomous response matrices (Liu and Terzi, 2010).

After parameter estimation, they use sensitivity (i.e., β_i) values and true visibilities $P_{ij} = Prob\{R(i,j) = 1\}$ of all items in Equation 4.51 or Equation 4.52 to calculate the privacy score of users. The authors call this way of privacy score computation Pr_IRT. They also introduced a naïve privacy score (Pr_Naive in short) which is a simple way of computing privacy score for OSN users. In the naïve method, basic idea is that the higher the sensitivity of an item, the smaller number of users who are willing to disclose it. Let $|R_i|$ represents the number of users who set $R(i,j) = 1$, then the sensitivity β_i in dichotomous case computed as follows:

$$\beta_i = \frac{N - |R_i|}{N} \tag{4.54}$$

Sensitivity is also computed in the polytomous case by generalizing the equation given above as follows:

$$\beta_{ik}^{*} = \frac{N - \sum_{j=1}^{N} I_{(R(i,j) \geq k)}}{N} \tag{4.55}$$

Then β_{ik} values are computed depending on the privacy level k as follows: if the k equals to the minimum (i.e., 0) or maximum (i.e., l) privacy level, then the $\beta_{ik} = 0$ or $\beta_{ik} = l$. On the other hand, for $k \in 1, \ldots, l-1$, the β_{ik} is obtained with the following equation:

$$\beta_{ik} = \frac{\beta^*_{ik} + \beta^*_{i(k+1)}}{2} \tag{4.56}$$

Visibility computation in the naïve approach assumes independence between items and users and estimates the probability $P_{ij} = Prob\{R(i,j) = 1\}$ as follows in the dichotomous case:

$$P_{ij} = \frac{|R_i|}{N} \times \frac{|R^j|}{n} \tag{4.57}$$

where $|R^j|$ is the number of items that user j sets $R(i,j) = 1$. The visibility in the polytomous case also obtained by estimating the probability $P_{ijk} = Prob\{R(i,j) = k\}$ as follows:

$$P_{ijk} = \frac{\sum_{j=1}^{N} I_{(R(i,j)=k)}}{N} \times \frac{\sum_{i=1}^{n} I_{(R(i,j)=k)}}{n} \tag{4.58}$$

After obtaining sensitivity and visibility values, the naïve privacy score can be computed by using Equation 4.51 or Equation 4.52. As stated in Section 2.3, this work has led to a general idea in the privacy scoring of users. As such, the majority of subsequent studies used Liu and Terzi's (2010) idea of mathematical measurement of privacy risk of users based on the R matrix. One of these studies is performed by Pensa et al. (2019) that suggests two privacy risk scoring methods which are IPS and NPS.

In this thesis, we use Pr_IRT, IPS and NPS methods for privacy analysis (see Section 5.2.3) of users in the crawled data. IPS method is based on the R matrix and requires sensitivity and visibility calculations in a similar but different way from Liu and Terzi's (2010) work. These two main components are computed thanks to the R matrix. Sensitivity of an item (or topic) j for any visibility degree (or privacy level) $h = \{1, \dots, l-1\}$ is calculated as follows (Pensa et al., 2019):

$$\sigma_{jh} = \frac{1}{2}\left(\frac{n-\sum_{i=1}^{n}\mathbb{I}_{(r_{ij}\geq h)}}{n} + \frac{n-\sum_{i=1}^{n}\mathbb{I}_{(r_{ij}\geq h+1)}}{n}\right) \quad (4.59)$$

where n is the number of users and $\mathbb{I}_A$ is a function that returns 0 or 1 when condition A is false or true. On the other hand, when h takes one of the extreme values (i.e., $h = 0$ or $h = l$) this calculation changes as follows:

$$\sigma_{j0} = \frac{n-\sum_{i=1}^{n}\mathbb{I}_{(r_{ij}\geq 1)}}{n} \text{ or } \sigma_{jl} = \frac{n-\sum_{i=1}^{n}\mathbb{I}_{(r_{ij}\geq l)}}{n} \quad (4.60)$$

Visibility of item j for any degree $h = 0, \ldots, l$ with respect to user i is obtained with the following equation:

$$v_{ijh} = \frac{\sum_{i=1}^{n}\mathbb{I}_{(r_{ij}=h)}}{n} \times \frac{\sum_{i=1}^{m}\mathbb{I}_{(r_{ij}=h)}}{m} \times h \quad (4.61)$$

where m represents the number of items. Then privacy risk $\bar{\rho}_p = (v_i, t_j)$ that a user v_i has with respect to a given item t_j is calculated with the help of the following equation (Pensa et al., 2019):

$$\bar{\rho}_p(v_i, t_j) = \frac{\rho_p(v_i, t_j)}{\max_{v_k \in V} \rho_p(v_k, t_j)} \quad (4.62)$$

where V represents a set of all users and

$$\rho_p(v_i, t_j) = \sum_{h=0}^{l} \sigma_{jh} \times v_{ijh}. \quad (4.63)$$

The $\max_{v_k \in V} \rho_p(v_k, t_j)$ is, on the other hand, the maximum value of Equation 4.63 among all users. Finally, the overall IPS $\rho_p(v_i)$ for any given user v_i obtained as follows:

$$\rho_p(v_i) = \sum_{j=1}^{m} \bar{\rho}_p(v_i, t_j) \tag{4.64}$$

The NPS requires running personalized PageRank (Brin and Page, 1998; Jeh and Widom, 2003) algorithm over IPS values with the following equation (Pensa et al., 2019):

$$P = dA^T P + \frac{(1-d)}{\sum_{k=1}^{n} \rho_p(v_k)} \rho \tag{4.65}$$

where d is damping factor, $\rho = \left[\rho_p(v_1), \ldots, \rho_p(v_n)\right]^T$ and A is an $n \times n$ symmetric matrix in which each element $a_{ij} = a_{ji} = 1/\deg(v_i)$. After obtaining P that includes NPS values of users, a min-max normalization is applied to adjust these values to have the same scale as IPS values. Consequently, the NPS calculation with this normalization is as follows:

$$NPS(v_i) = p(v_i) \cdot \frac{\max\{\rho_p(v_j)\} - \min\{\rho_p(v_j)\}}{\max\{p(v_j)\} - \min\{p(v_j)\}} \tag{4.66}$$

where $p(v_i)$ and $NPS(v_i)$ represent NPS and recomputed NPS values for the user (i.e., node) v_i respectively. NPS method is proposed to improve privacy score measuring with respect to the characteristics of the OSN.

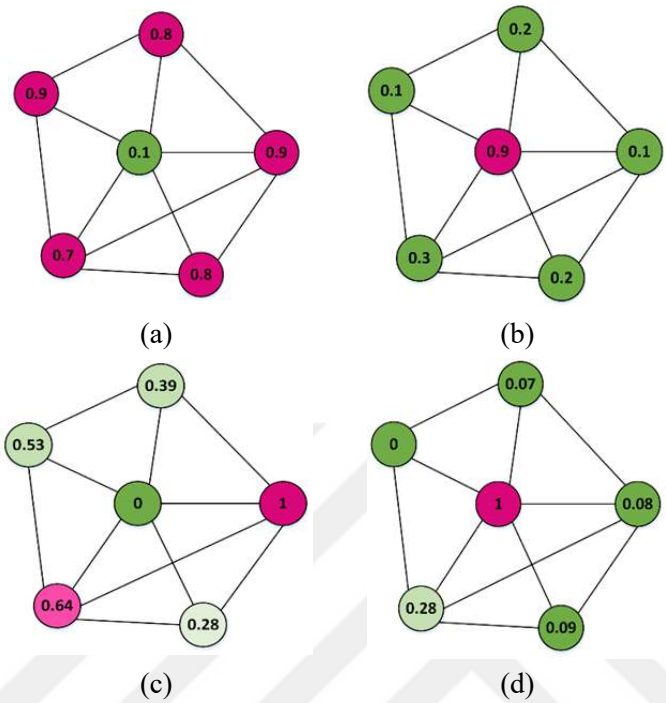


Figure 4.17. Characteristics of two networks with respect to the IPS and NPS values.

A sample NPS calculation is depicted in Figure 4.17 where (a) and (b) show two networks in which the central node (i.e., user) is privacy-aware and unaware with respect to their IPS values, while (c) and (d) show characteristics of these two networks with respect to the NPS values of nodes. Since NPS values are scaled, a zero value does not indicate that the relevant node is not at risk (Coban et al., 2020b).

4.9.2. Measuring Social Tie Strengths

NPS method which is proposed by Pensa et al., (2019) adopts a well-known page rank algorithm to compute network-aware privacy risk scores of users. They only consider intrinsic privacy risks and social connections of users. Thus, they run page rank over graph representation of the OSN data in which all edges

(i.e., friendship links) are assumed to have equal weights (see Figure 4.17). However, an individual does not have the same social tie strengths (Banks and Wu, 2009) with all of his/her friends in the both online and offline world. As depicted in Figure 4.18 (Hangal et al., 2010), OSN users may have different intensities of social interactions with other users.

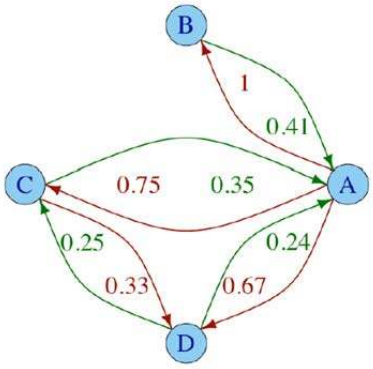


Figure 4.18. A sample graph of four OSN users having different social tie strengths with each other.

As such, in this thesis, we suggest and use social tie strengths in NPS calculation. Intuitively, several socially-oriented tasks (such as social search, community discovery, link prediction, etc.) could benefit from embedding tie strength into their algorithmic frameworks (Hangal et al., 2010). Therefore, tie strength measuring has received much attention from researchers in recent years. It is one of the hot topics in the OSN research field and many studies try to solve this task using different techniques including supervised learning (Kahanda and Neville, 2009; Krakan et al., 2018), regression models (Liu et al., 2020), similarity-based models (Seo et al., 2017), simple partial (Rodriguez et al., 2012) or linear functions (Ilić et al., 2016; Krakan et al., 2018), and so on.

A common point among existing studies is that they try to build models that can output a label (e.g., strong, weak) or a score representing social strength between two OSN users. These models depend on tie strength components and

produce their outputs by using weights (i.e., parameter importance) of these components. In this thesis, we use real-world Facebook data and we only know whether there exists a friendship link between two users or not. As the supervised learning approach requires labeled training data, we adopt and use a linear function to measure tie strength between users.

Table 4.14. Our tie strength components

Component	Code
Bidirectional components	
Whether user A and user B have the same gender	SGEN
The number of common friends	CFRI
The number of liked pages in common	CPAGE
Whether user A and user B work in the same company	SCOMP
Whether user A and user B went to the same school	SEDU
Whether user A and user B are from the same hometown	SHT
Whether user A and user B live-in the same place	SLI
Whether user A and user B have the same job title	SJOB
Unidirectional components	
The number of events that user A attended with user B	EVNT
Whether user A reveals that he/she is in a relationship with user B	PREL
Whether user A reveals that user B is his/her family member	FMEM
The number of posts in which user A tagged user B	PATAG
The number of posts in which user A tagged only user B	PSTAG
The number of directly commented posts (i.e., A writes the first comment for any post by user B)	CMDIR
The number of indirectly commented posts (i.e., A writes a comment in any order but not the first for any post by B)	CMIND
The number of directly replied comments (i.e., A writes the first reply for any comments by B)	RPDIR
The number of indirectly replied comments (i.e., A writes a reply in any order but not the first for any comment by B)	RPIND
The number of direct posts shared by user A in the wall page of user B	DRPST

Our model uses weights of tie strength components which are presented in Table 4.14. Inspiring by Wang et al's self-information method (2016), in this thesis, we adopt and use the self-information model to compute the weights of our components.

If we have c components $\lambda_1, \lambda_2, \dots, \lambda_c$, we then construct a binary vector $B_{e_{uv}}$ to represent a directed edge e_{uv} from user u to v (i.e., $u \rightarrow v$):

$$B_{e_{uv}} = \langle \mathcal{I}_{\lambda_1}(e_{uv}), \mathcal{I}_{\lambda_2}(e_{uv}), \dots, \mathcal{I}_{\lambda_c}(e_{uv}) \rangle \tag{4.67}$$

where $\mathcal{I}_{\lambda}(e_{uv})$ is the component indicator function which indicates whether e_{uv} satisfies component λ as follows:

$$\mathcal{I}_{\lambda}(e_{uv}) = \begin{cases} 1 & if\ e_{uv}\ satisfies\ the\ component\ \lambda \\ 0 & otherwise \end{cases} \tag{4.68}$$

Next, we adopt the self-information model given in Equation 4.19, which is actually a weighting method, to compute component weights as follows (Wang et al., 2016):

$$W(\lambda_i) = \widehat{Pr}\left(\mathcal{I}_{\lambda_i}(.) = 1\right) = \frac{\left|e \epsilon E \middle| \mathcal{I}_{\lambda_i}(e) = 1\right|}{|E|} \tag{4.69}$$

where $W(\lambda_i)$ is the weight of component λ_i and equal to its self-information which is defined as $I(m) = -logPr(m)$ where $\Pr(m)$ represents the probability that message m is chosen from all possible choices in the message space. By the way, $W(\lambda_i) = I(\mathcal{I}_{\lambda_i}(.) = 1) = -logPr(\mathcal{I}_{\lambda_i}(.) = 1)$ where it is not possible to get the true value of $Pr\left(\mathcal{I}_{\lambda_i}(.) = 1\right)$. As such, we obtain the value of $W(\lambda_i)$ by estimating the $Pr\left(\mathcal{I}_{\lambda_i}(.) = 1\right)$ value on our directed edge set E.

Finally, the tie strength of an edge in our method is calculated as the sum of multiplication of the values of tie strength components with the corresponding weight of components. Mathematical model of our method is as follows:

$$tie\_strength(e_{uv}) = \sum_{i=0}^{c} O_{\lambda_i(e_{uv})} \times W(\lambda_i) \tag{4.70}$$

where $O_{\lambda_i(e_{uv})}$ is the observed value of the component λ_i in edge e_{uv}, while $W(\lambda_i)$ stands for the weight of the component λ_i.

5. RESULTS AND DISCUSSIONS

The following two main sub-headings segment and present results of our tasks which are grouped under two main sections, namely, *"social network analysis and mining"* and *"attribute inference and privacy risk scoring"*.

5.1. Social Network Analysis and Mining

In this subsection, we present the results of our experiments which are related to SNA and mining, and performed on both snapshots of the crawled data and its subsets.

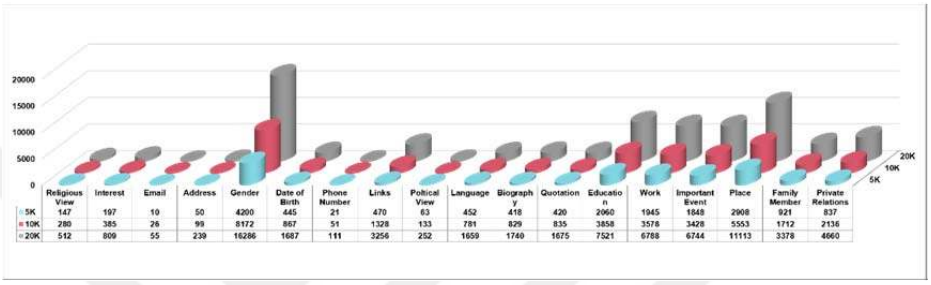
5.1.1. Sharing Analysis of Crawled Data

In this task, we first focus on the problem of efficient collection of large-scale, content-rich, and accurate Facebook data. We present an algorithmic data collection methodology (see Section 4.2) that is applicable to any OSN. Using this approach, we implemented a crawler which has various advantages over existing OSN data collection efforts. The key differences are as follows (Coban et al., 2020a):

- supports multi-threaded data collection with consistency guarantees,
- is resilient against access restrictions imposed by the OSN service provider,
- collects every bit of publicly shared data and respects BFS order from the seed account, and
- independent of the frequent API updates pushed by the OSN service provider.

We utilize Facebook OSN to exemplify the advantages of our crawler in collecting OSN data. Over roughly two months, we collected three different

snapshots of Facebook data. Upon completion of crawling, we performed analyses to uncover sharing behaviors by providing insight over Facebook usage behavior of Turkish users: what, when, and how they share and with whom they share data. Our analysis mainly concerns with what kind of information users tend to share about themselves and what type of content they mostly post on their timelines.



	Religious View	Interest	Email	Address	Gender	Date of Birth	Phone Number	Links	Political View	Language	Biography	Quotation	Education	Work	Important Event	Place	Family Member	Private Relations
5K	147	197	10	50	4200	445	21	470	63	452	418	420	2060	1945	1848	2908	921	837
10K	280	385	26	99	8172	867	51	1328	133	781	829	835	3858	3578	3428	5553	1712	2136
20K	512	809	55	239	16286	1687	111	3256	252	1659	1740	1675	7521	6788	6744	11113	3378	4660

Figure 5.1. Publicity frequencies of disclosed user attributes in three snapshots

We performed network analysis along with detailed statistical analyses of the crawled data in terms of page publicity, basic attributes, friendship and relations, and wall activities. To perform statistical analysis of public OSN data, we used our tool (see Figure A.7) that uses aggregate statistical queries of relational and graph DBMSs. Following sub-headings present the most interesting subset of our analysis results for the largest snapshot, but the rest of the results of our analyses are presented in Appendix A.

(1) Publicity of Pages: Privacy-aware Facebook users keep their wall and friend pages private. Table A.7 presents the number of users who make their pages public, private, or partially public in the crawled snapshots. As seen from this table, users mostly tend to hide their friend list but, the same is not true for wall activities. To exemplify for the largest snapshot, only 7% (i.e., 1,421) of users hide wall pages, whereas 58% (i.e., 11,675) of them completely hide friends page. Note that partially hidden means that one can only see common friends of WO.

(2) Basic Attributes: Some of the profile information on Facebook is compulsory (e.g., e-mail, username) to sign-up, while the rest is often disclosed by

willingnessly to increase digital existence. Figure 5.1 shows the summary of publicity ratios for basic attributes among users in our crawled snapshots. As seen from this figure, phone number and e-mail are the least shared attributes, while the most shared attributes are gender and place. When we have a closer look at the basic attributes, we also observe that majority of users have ungraduated education and interested in females. English and Turkish are mostly disclosed languages, while among users who reveal religious view attribute, the majority of those express their belief as Muslim. Besides, users mostly share their Instagram accounts on their Facebook profiles to stay in connection with other OSNs.

Phone number and address attributes are usually disclosed by accounts (e.g., local company) which are used for business purposes. Using birth-date information, we also calculated the age of users and observed that majority of them are in the age range of 31 to 40. Additionally, our simple analysis on usernames shows that the majority of users prefer to select a username with any string (i.e., *"jack.sparrow.25"*), while a minority of users select a username that is completely composed of numeric values. Our analysis of users' places makes it clear that they mostly live in big cities (e.g., Istanbul, Bursa, Ankara) and those living in these cities mostly come from another city.

(3) Friendship and Relations: Users share their family members and private relations as well as friendships. When we explore our data, we find out that users who have a private relationship mostly state that type of relationship is marriage. The most shared family membership (or kinship) types are cousin and brother.

(4) Wall Activities: We explore wall activities of users depending on both basic attributes of users and meta-data (i.e., time of posting, via, place, etc.) information of activities. According to our basic analysis, users prefer the *"like"* reaction with a percent of 98 to express their feelings about any content. They often share more content in the winter months (i.e., most in January) and the most common time that users are least active is between 02:00 hours and 06:00 hours.

Important events are shown as a post on the wall page by Facebook. Considering events as posts, our analysis also discovers that the most shared event types are *starting work* and *graduation* among users. With a deeper analysis, we also find that male users have more wall activity than female users. Using titles and contents of posts, on the other hand, we categorized posts and observed that users mostly post contents which are including raw text or any photograph.

As seen from Appendix A, the revealed data has a similar trend in three snapshots. Our analysis in this task only concerns a simple statistical analysis of crawled data in this section. However, it is still possible to learn more from the data such as users' past and current locations, attitudes, attended groups, favorite music, social connectivity, and so on. This is because even though OSNs provide privacy settings to enable their users to control what information can be shared with whom, many OSN users do not sufficiently leverage these settings and share public information which is arguing that they have nothing to hide or they do not understand how the privacy settings really work. Such behavior threatens the individual privacy and security of both the public sharers and those users that are connected to them on the same OSN. The main reason behind this behavior is due to the users often want to increase their digital existence by providing more information about them. Even though this is a risky matter, data provided by users make OSNs a huge and rich source of data about people.

The following sections give results of our other tasks that have such a purpose by mining the OSN data.

5.1.2. NER Over Crawled Data

NER over OSNs remains to be a challenging task as user-generated contents are often containing noisy texts that have grammatical errors. In this task, therefore, we developed a CRF-based NER system (see Figure B.2) to extract NEs from Facebook posts. Contributions of this task can be summarized as follows:

- we present FBNER (see Table 3.3) dataset that contains manually labeled 5K Facebook posts written in Turkish language and
- we present comparable results on the FBNER dataset using well-known datasets from different domains such as news, and tweets.

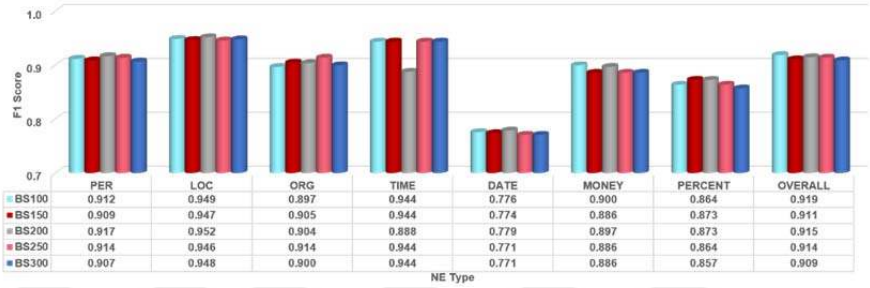
In this section, we presented our NER task's results which are obtained by using a CRF-based NER system that utilizes features described in Section 4.4.5. We obtained our results in two-step experiments. In the first phase, we aimed to detect the best feature set and investigate the robustness of our NER system. For this purpose, we obtained our results on the Train7 and WFS7 datasets as training and test sets respectively as done by Şeker and Eryigit (2017). To detect the best feature set, we started with the base feature set (i.e., LEX) and include other extracted feature sets in each step.

Table 5.1. F1 results on Train7 and WFS7 datasets with different feature sets.

Feature Set	**PER**	**LOC**	**ORG**	**TIME**	**DATE**	**MONEY**	**PERCENT**	**OVERALL**
LEX **(Base Set)**	0.852	0.928	0.866	0.882	0.738	0.886	0.873	0.875
+ GLU	0.909	0.937	0.877	0.882	0.726	0.875	0.847	0.898
+ VEC	0.910	0.938	0.874	0.882	0.738	0.864	0.890	0.899
+ POS	0.907	0.939	0.883	0.882	0.738	0.864	0.890	0.901
+ STM	0.910	0.938	0.886	0.882	0.744	0.864	0.883	0.902
+ PRP	0.915	0.937	0.888	0.882	0.750	0.864	0.873	0.904
+ NCS	0.915	0.937	0.888	0.882	0.741	0.864	0.833	0.903
+ INF	0.909	0.938	0.877	0.882	0.743	0.875	0.876	0.900
+ CXW	0.915	0.946	0.897	0.914	0.730	0.871	0.879	0.908
+ CXP	0.915	0.949	0.892	0.944	0.774	0.886	0.864	0.911
All – NCS – INF **(Best Set)**	0.912	0.949	0.897	0.944	0.776	0.900	0.864	**0.911**

As seen in Table 5.1, we obtained the highest result when all feature sets are employed together. However, we excluded NCS and INF features from the best set due to these feature sets cause to decrease in overall success. The best feature

set, namely, BS includes LEX, GLU, VEC, POS, STM, PRP, CXW, and CXP features and achieves an F1 score of 0.911. Note that in this step, VEC features are employed with a layer size of 100. Therefore, we also investigated the effect of the layer size of VEC features that are included in the BS features.

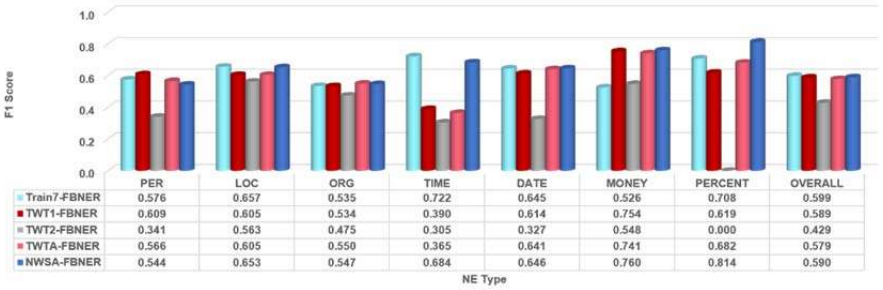


	PER	LOC	ORG	TIME	DATE	MONEY	PERCENT	OVERALL
BS100	0.912	0.949	0.897	0.944	0.776	0.900	0.864	0.919
BS150	0.909	0.947	0.905	0.944	0.774	0.886	0.873	0.911
BS200	0.917	0.952	0.904	0.888	0.779	0.897	0.873	0.915
BS250	0.914	0.946	0.914	0.944	0.771	0.886	0.864	0.914
BS300	0.907	0.948	0.900	0.944	0.771	0.886	0.857	0.909

Figure 5.2. Effect of layer size for VEC features used in the best feature set BS.

As seen from Figure 5.2, we use different layer sizes (l) where $l \in \{100, 150, 200, 250, 300\}$ and obtained the best result as an F1 score of **0.915** when employing VEC features with layer size of 200. Notice that in Figure 5.2, the numerical value next to the BS represents the layer size for VEC features included in the best set.

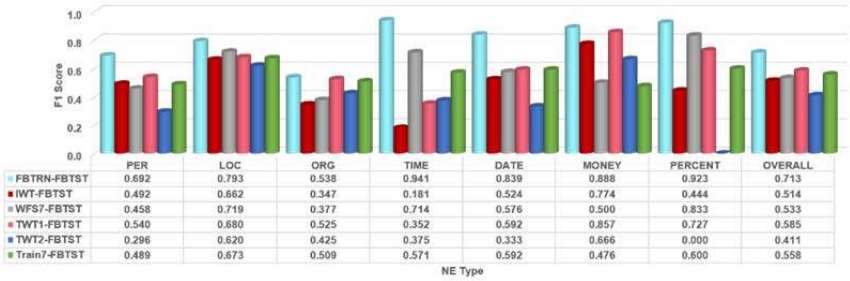
Our results in the first phase show that our NER system achieves a slightly different F1 score (i.e., **0.915**) from the best F1 score (i.e., **0.923**) which is obtained by Şeker and Eryigit (2017), even we use different preprocessing techniques, gazetteers, and feature sets. This makes it clear that our system is quite successful and robust in the discovery of NEs. As we obtained the best F1 score with the BS200, we use this feature set in the rest of our NER experiments.



	PER	LOC	ORG	TIME	DATE	MONEY	PERCENT	OVERALL
Train7-FBNER	0.576	0.657	0.535	0.722	0.645	0.526	0.708	0.599
TWT1-FBNER	0.609	0.605	0.534	0.390	0.614	0.754	0.619	0.589
TWT2-FBNER	0.341	0.563	0.475	0.305	0.327	0.548	0.000	0.429
TWTA-FBNER	0.566	0.605	0.550	0.365	0.641	0.741	0.682	0.579
NWSA-FBNER	0.544	0.653	0.547	0.684	0.646	0.760	0.814	0.590

Figure 5.3. F1 scores obtained on different pairs of train-FBNER datasets

Similar to the first phase, the second phase includes two-step experiments on different training and test sets. First, we perform experiments on the FBNER dataset by training our system with well-known datasets (see Table 3.3) from other domains. Notice that some of the datasets presented in Table 3.3 have fewer NEs even they have more sequences than the FBNER. As such, these datasets have not been used in this step of our experiments which are conducted just on the Train7, TWT1, TWTA, NWSA, and TWT2 datasets. Figure 5.3 depicts the obtained results in this step.



	PER	LOC	ORG	TIME	DATE	MONEY	PERCENT	OVERALL
FBTRN-FBTST	0.692	0.793	0.538	0.941	0.839	0.888	0.923	0.713
IWT-FBTST	0.492	0.662	0.347	0.181	0.524	0.774	0.444	0.514
WFS7-FBTST	0.458	0.719	0.377	0.714	0.576	0.500	0.833	0.533
TWT1-FBTST	0.540	0.680	0.525	0.352	0.592	0.857	0.727	0.585
TWT2-FBTST	0.296	0.620	0.425	0.375	0.333	0.666	0.000	0.411
Train7-FBTST	0.489	0.673	0.509	0.571	0.592	0.476	0.600	0.558

Figure 5.4. F1 scores obtained on different pairs of train-FBTST datasets

Next, we aimed to perform NER experiments by using such datasets that have fewer NEs than the FBNER as well. Following our purpose, we tested our system over the FBTST dataset by training it with different datasets including the FBTRN dataset. Figure 5.4 also shows the obtained results in this second step. Based on all of the results in our NER task, we report the following findings:

- Our NER system trained with the Train7 dataset achieves the best F1 score of **0.599** on the FBNER dataset.
- The worst results are obtained when the system is trained with the TWT2 dataset.
- TWT1 dataset produces slightly different results than the Train7 when they are used individually.
- Training the NER system with the FBTRN produces the best F1 score of **0.713** on the FBTST dataset.

To summarize, training our system with datasets from the news domain often gives better results. The performance of the NER task depends on the domain, tagging format, scope, and size of the datasets used. In this task, we first employed our NER system on the Train7 and WFS7 datasets which are from the news domain to capture the best feature set. In the second phase, we investigated whether news or user-generated contents are more suitable to train a Facebook NER system. In this phase, we performed two-steps experiments by using the best feature set to reach our goal.

In summary, we achieved the best result as an F1 score of **0.713** when we use subsets of the FBNER dataset. We also obtained the best result as an F1 score of **0.599** when we train our system with the Train7 dataset. In the second phase, we obtained the worst results once we use the TWT2 dataset for training the system. The main reason for this is that except for *"Person"* entities, the number of NEs for TWT2 is less than the other datasets. For instance, this dataset only contains three samples in the *"Percent"* type and this is not enough for CRF to learn and distinguish *"Percent"* entities from others. Training our NER system with datasets from the news domain often produces better results. However, to obtain better results on user-generated content, it is required to have both training and test sets to be from the same domain.

5.1.3. SA of Crawled Data

In this task, we focus on document-level SA on Facebook OSN in the context of the Turkish language. We carried out extensive experiments to compare ML and DL approaches and explore the best learning strategy for SA from Facebook data in Turkish. Besides, we also explored the relationship between user attributes such as gender and age, and the sentimental polarity of the activities of these users. Contributions of this task can be summarized as follows (Coban et al., 2021):

- The performance of various approaches including ML-based, lexicon-based, and hybrids of the two on SA of Facebook data is experimentally analyzed.
- Well-known DL algorithms are applied to SA of Facebook data to extensively compare their performance with ML-based and lexicon-based methods.
- The polarity of Facebook activities with respect to the several user/activity attributes are extracted and empirically analyzed.
- The activities users are more engaged with and the specific reactions users prefer are analyzed experimentally.
- The accuracy effects of including emoticons as features for the problem of SA are explored.
- The effects of applying feature selection during preprocessing on the accuracy for both ML and DL algorithms are investigated.
- A manually labeled SA dataset including Turkish texts collected from public activities is created (see Section 3.3.2.4).

We believe that the language context, scale of the experimental dataset used, both depth and breadth of applied techniques, and our deeper analyses separate this task significantly from existing works.

In this section, we report the results of our extensive experiments in two parts. Notice that to evaluate the performance of any classifier, we use the accuracy metric for all of the experiments and configure classifiers to run with 5-fold cross-validation. We first performed experiments to explore which learning method and feature set are most performant on the model selection dataset. Secondly, we conducted experiments by using the best model which is detected in the first step, and, finally, we explored several relationships between activities and their writers.

5.1.3.1. Choosing The Best Model

To detect the best classification model, first of all, we compare the ML and DL methods and try to determine the most accurate classification model for SA from our dataset. We run all ML algorithms with default parameter settings except for the following: the kernel is linear for SVM and the maximum number of iterations is 1,000 for LR. Notice that in this task, we apply basic or linguistic preprocessing on activity contents depending on the feature model. Our linguistic preprocessing includes common steps which are described in Section 4.3.

Figure 5.5. List of positive and negative emojis which correspond to emoticons used in SA task.

On the other hand, our basic preprocessing (see Section 4.3), in this task, includes the following additional steps together with common ones:

- We apply emoticon encoding to keep emoticons within texts. This step is employed optionally at first to explore the effect of emoticon features on the performance. In this step, we use emoticons to encode positive ones with *"PosEmoticon"* and negative ones with *"NegEmoticon"*. Figure 5.5 depicts a list of the corresponding emojis for our emoticons.
- Removing person names. Especially first names are often used as a common word in a daily language in Turkish. For instance, the word *"Gül (rose)"* may be a person's name and a verb (i.e., *"Gül-mek (laugh)"*) in a sentence. Removing such vowel names may prevent biased results and this step is only applied to the model selection data manually.
- Removing mentions (e.g., @someone) and URLs.
- Making the *"w→v"*, *"$→ş"*, and *"q→k"* character conversions as users often prefer to use these instead of the Turkish equivalents (e.g., *"a$qım→aşkım (my love)"*, *"dewrem→devrem (my friends)"* etc.).
- Correcting informal use of formal words (e.g., *"ağbi→abi"*, *"eyw→eyvallah"*, etc.) based on our lexicon of informal words (see Section 3.5.1).
- Making *"10→on"* conversion when it comes before the word *"numara (number)"*. Notice that Turkish Facebook users may use these two words together to show their positive opinions about anything.

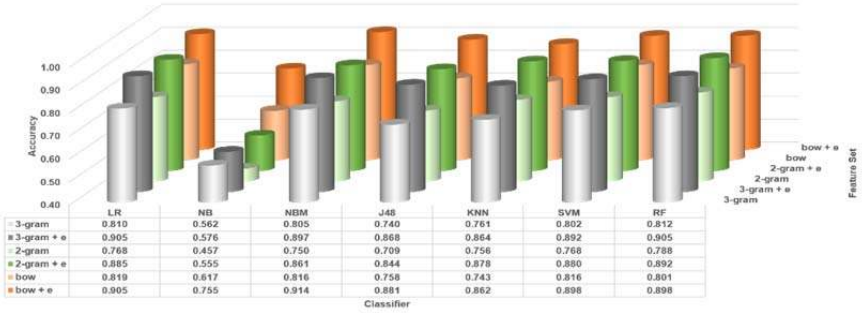
In this task, we use the NLTK package to remove stopwords and perform linguistic preprocessing with the help of the Zemberek.

(1) Performance of ML-based Classifiers: On the ML side, we named each experiment with a prefix *"ExML"* and performed the following experiments respectively:

- ExML1: Detecting the effect of using emoticons as features, the best performant feature set, and weighting scheme.
- ExML2: Exploring the effect of feature selection.
- ExML3: Using the mean of word embeddings as input features.
- ExML4: Using inferred document vectors as input features.
- ExML5: Feature extension by using polarity features.

In ExML1, we apply preprocessing, feature extraction, term weighting, and classification steps respectively. In feature extraction, we extracted 8,866 bow, 805 bi-gram (2-gram), and 7,549 tri-gram (3-gram) unique content features. In the term-weighting phase, we use binary, tf, tf*idf, and fuzzy schemes which are described in Section 4.6. In the classification step, on the other hand, we use well-known ML classifiers including LR, NB, NBM, J48, KNN, SVM, and RF. As stated before, we apply preprocessing based on the feature model and bow features are extracted with linguistic preprocessing to avoid a huge number of features, while n-gram features extracted with basic preprocessing as the n-gram model is language independent.

We applied our basic preprocessing in two different ways by opening and closing the emoticon encoding step to investigate the effect of keeping emoticons within texts on the results. Figure 5.6, Figure 5.7, Figure 5.8, and Figure 5.9 show experimental results of this step in which +e represents using encoded emoticon features.



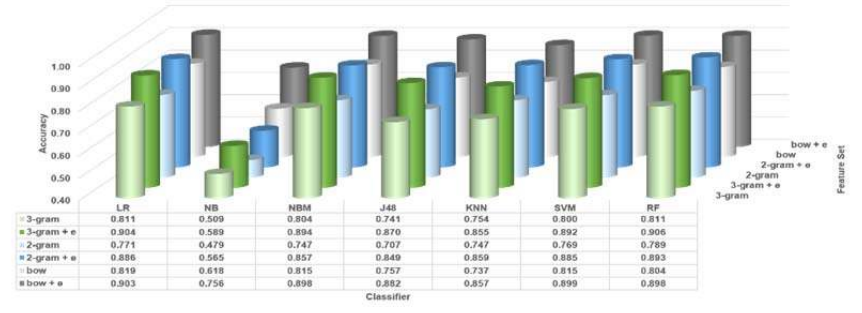
	LR	NB	NBM	J48	KNN	SVM	RF
3-gram	0.810	0.562	0.805	0.740	0.761	0.802	0.812
3-gram + e	0.905	0.576	0.897	0.868	0.864	0.892	0.905
2-gram	0.768	0.457	0.750	0.709	0.756	0.768	0.788
2-gram + e	0.885	0.555	0.861	0.844	0.878	0.880	0.892
bow	0.819	0.617	0.816	0.758	0.743	0.816	0.801
bow + e	0.905	0.755	0.914	0.881	0.862	0.898	0.898

Classifier

Figure 5.6. Obtained accuracies on binary-weighted data for different feature sets and classifiers.

As seen in Figure 5.6, the most successful classifiers are often LR and RF, while the worst one is NB. NBM classifier achieves better results on bow features. Tri-gram features provide better results than bi-gram features and bow features outperform n-gram features. On the binary-weighted data, we achieved the best result with an accuracy of **0.914** with the help of the NBM classifier on bow+e features. It is also clear that using encoded emoticon features improves our results in all cases.

As seen from Figure 5.7, tf weighting scheme produces slightly different results which tend to show similar behavior with the results of binary-weighted data. The best result of binary-weighted data is obtained with an accuracy of **0.904** with the help of the LR classifier on 3-gram+e features.

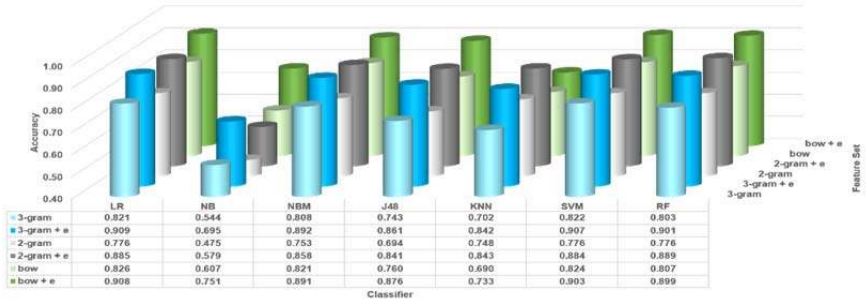


	LR	NB	NBM	J48	KNN	SVM	RF
3-gram	0.811	0.509	0.804	0.741	0.754	0.800	0.811
3-gram + e	0.904	0.589	0.894	0.870	0.855	0.892	0.906
2-gram	0.771	0.479	0.747	0.707	0.747	0.769	0.789
2-gram + e	0.886	0.565	0.857	0.849	0.859	0.885	0.893
bow	0.819	0.618	0.815	0.757	0.737	0.815	0.804
bow + e	0.903	0.756	0.898	0.882	0.857	0.899	0.898

Classifier

Figure 5.7. Obtained accuracies of tf weighted data for different feature sets and classifiers.

Our results on tf*idf weighted data (see Figure 5.8) show that LR and SVM often outperform other classifiers and the best result is obtained with an accuracy of **0.909** using LR classifier with 3-gram+e features.

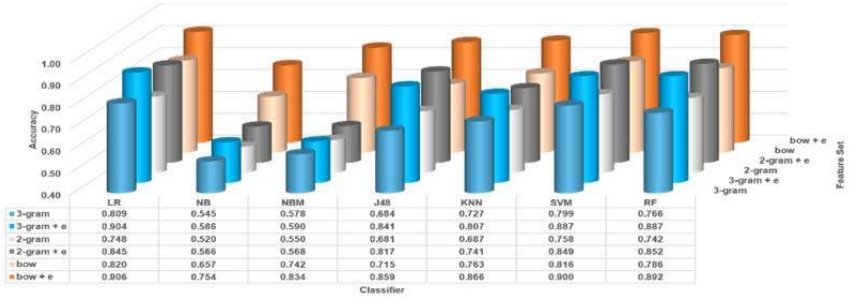


	LR	NB	NBM	J48	KNN	SVM	RF
3-gram	0.821	0.544	0.808	0.743	0.702	0.822	0.803
3-gram + e	0.909	0.695	0.892	0.861	0.842	0.907	0.901
2-gram	0.776	0.475	0.753	0.694	0.748	0.776	0.776
2-gram + e	0.885	0.579	0.858	0.841	0.843	0.884	0.889
bow	0.826	0.607	0.821	0.760	0.690	0.824	0.807
bow + e	0.908	0.751	0.891	0.876	0.733	0.903	0.899

Figure 5.8. Obtained accuracies of tf*idf weighted data for different feature sets and classifiers.

Similarly, LR and SVM often produce better results on the fuzzy weighted dataset as depicted in Figure 5.9. The best result in this phase is obtained with an accuracy of **0.906** by using the LR classifier on bow+e features.

When we consider our complete results in this step, we observed that our results tend to show similar behavior for binary, tf, tf*idf, and fuzzy weighted schemes. Even our results are slightly different, it is clear that higher results obtained when using emoticon features and binary and tf*idf weighting schemes often outperform the tf and fuzzy schemes. The best accuracy (i.e., **0.914**) in this step is obtained by the NBM on a feature set that includes binary-weighted bow and emoticon features (i.e., bow+e). Therefore, in the rest of our experiments, we use the letter *"A"* to denote this best performant feature set.

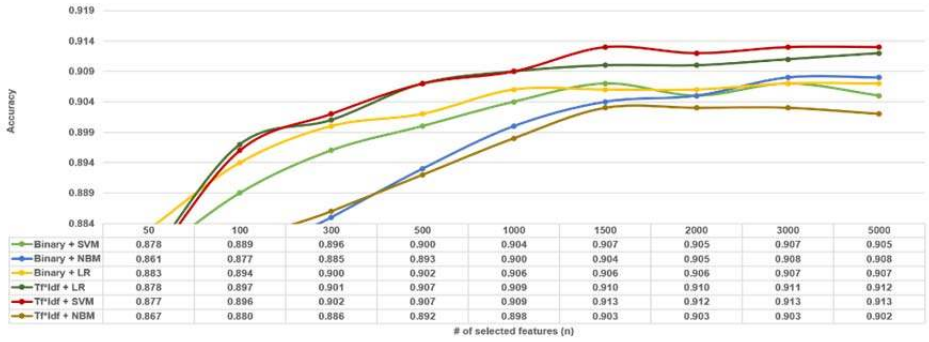


	LR	NB	NBM	J48	KNN	SVM	RF
3-gram	0.809	0.545	0.578	0.684	0.727	0.799	0.766
3-gram + e	0.904	0.586	0.590	0.841	0.807	0.887	0.887
2-gram	0.748	0.520	0.550	0.681	0.687	0.758	0.742
2-gram + e	0.845	0.566	0.568	0.817	0.741	0.849	0.852
bow	0.820	0.657	0.742	0.715	0.763	0.816	0.786
bow + e	0.906	0.754	0.834	0.859	0.866	0.900	0.892

Figure 5.9. Obtained accuracies of fuzzy weighted data for different feature sets and classifiers.

In ExML2, we selected binary and tf*idf schemes and LR, SVM, and NBM classifiers to perform experiments on the feature set A. The reason for this is that selected classifiers and weighting schemes often produce better, but slightly different results in the ExML1. In this step, we use chi-square as a feature selector which needs to take the number of features to be selected (n) as a parameter. However, detecting the optimum value of the n is hard and varies depending on the dataset at hand. As such, we use a simple step by step approach where the variable n is varied between 50 and 5,000 by increasing its value by 10 in each step.

Our results are depicted in Figure 5.10 that does not present all results for parameter n in this range to reduce complexity. As seen from the figure, feature selection produces slightly different results especially using a lower number of features. The best result is obtained with an accuracy of **0.913** with the SVM classifier by using the most informative 1,500 bow features (including emoticon features). In the rest of our experiments, we also use *"S"* to denote these selected best set of features.



	50	100	300	500	1000	1500	2000	3000	5000
Binary + SVM	0.878	0.889	0.896	0.900	0.904	0.907	0.905	0.907	0.905
Binary + NBM	0.861	0.877	0.885	0.893	0.900	0.904	0.905	0.908	0.908
Binary + LR	0.883	0.894	0.900	0.902	0.906	0.906	0.906	0.907	0.907
Tf*Idf + LR	0.878	0.897	0.901	0.907	0.909	0.910	0.910	0.911	0.912
Tf*Idf + SVM	0.877	0.896	0.902	0.907	0.909	0.913	0.912	0.913	0.913
Tf*Idf + NBM	0.867	0.880	0.886	0.892	0.898	0.903	0.903	0.903	0.902

Figure 5.10. Obtained accuracies feature set A with respect to the classifier, the number of selected features, and weighting scheme.

In the rest of our experiments, we use feature sets A and S to explore the effect of feature selection on both ML and DL sides. If the related model requires word embeddings as input, we use a simple strategy in which if any word in a text is not observed in the S, we skip it and do not query the word embedding model to get its vector. Notice that we obtained word embeddings from the relevant data by applying emoticon encoding as well. This is because feature sets A and S include encoded emoticon features.

In ExML3, we represented each text by taking the average of the vectors of words which are observed in related text. To obtain distributed representations of words we re-trained our WE_{FB} models in accordance with our task-dependent linguistic preprocessing and obtained five different models with different layer sizes. We also use other previously trained WE_{TR} and WE_{Fast} models to investigate the effect of the domain of the word embedding data on results.

Table 5.2 presents related results of this experiment, where subscript + and – in the model name represent basic and linguistic preprocessing applied on the model selection data respectively. As seen from the table, linguistic preprocessing improves results on previously trained word embedding models. The feature set S also produces slightly different, but better results than feature set A. LR and SVM classifiers obtained the best accuracies as 0.889, 0.770, and 0.822 for the WE_{FB},

WE_{TR}, and WE_{Fast} models respectively. This shows that training word embedding model on a corpus from the same domain produces better results.

Table 5.2. Obtained accuracies for different word embedding models with respect to the classifier, feature set, and layer size.

WE Model	Layer Size	Feature Set	Classifier	
			LR	SVM
$WE_{TR(+)}$	400	A	0.765	0.769
		S	0.767	**0.770**
$WE_{TR(-)}$		A	0.726	0.725
		S	0.767	**0.770**
$WE_{Fast(+)}$	300	A	0.782	0.801
		S	0.807	**0.822**
$WE_{Fast(-)}$		A	0.795	0.809
		S	0.798	0.816
$WE_{FB(+)}$	100	A	0.874	0.871
		S	0.878	0.877
	200	A	0.881	0.878
		S	0.882	0.882
	300	A	0.885	0.883
		S	**0.889**	**0.883**
	400	A	0.883	0.879
		S	0.884	0.883
	500	A	0.885	0.880
		S	0.886	0.882

In ExML4, we use distributed document vectors of texts to feed the classifiers. In this step, document vectors are inferred by the PV_{FB} models and the classification task is performed again by the LR and SVM classifiers. Notice that PV_{FB} models are trained by applying linguistic preprocessing on word embedding data as in the previous step. As seen from Figure 5.11, paragraph vectors produce worse results than the word vectors and classifiers are not sensitive to document vector of texts.

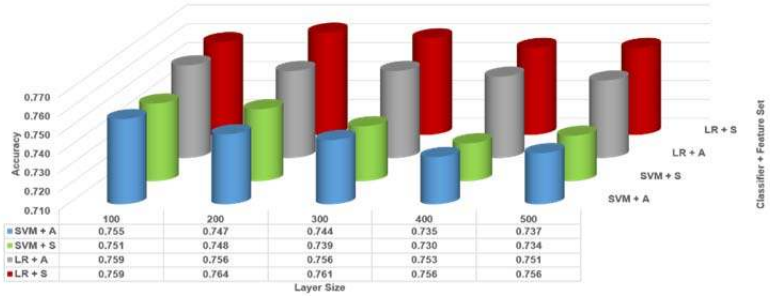


Figure 5.11. Obtained accuracies for different paragraph vector models with respect to the classifier, feature set, and layer size.

In ExML5, which is the final step of our experiments on the ML side, we performed feature extension by including the following features into the S:

- WE_{FB300}: average word vector for each text from the WE_{FB} model with layer size of 300. We select this model as it was the most successful model in the previous step of our experiments.
- Polarity features: F1 and F2 features that are obtained by a lexicon-based approach which is described in Section 4.4.2.1.

Table 5.3 presents our results which show that feature extension does not improve the accuracy. Polarity features have a negative effect on the accuracy and the best result is obtained with an accuracy of 0.889 with the help of the LR classifier.

Table 5.3. Obtained accuracies for the single or combined use of features employed on the ML side.

Feature Set	**Classifier**	
	LR	SVM
S + F1	0.853	0.850
S+ F2	0.856	0.853
S + F1 + F2	0.856	0.851
S + WE_{FB300}	0.889	0.884
S + WE_{FB300} + F1 + F2	0.888	0.884

Our overall results on the ML side present the following findings in summary:

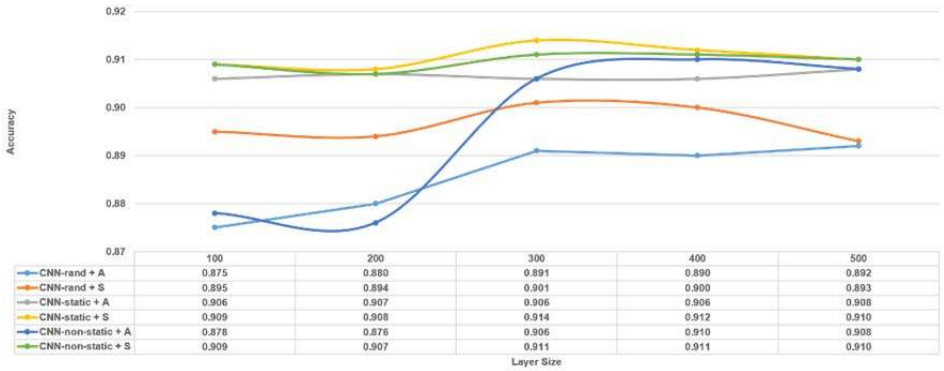
- bag-of-words (i.e., bow) feature model is the best model to represent texts, and using emoticons as features has a significant positive effect on results.
- feature selection often improves accuracy by using a lower number of features.
- tf*idf is generally the best weighting scheme.
- Paragraph vectors and feature extension methods are not able to improve accuracy.
- word vectors do not improve the best accuracy, but they outperform paragraph vector and feature extension approaches.
- obtaining higher accuracy with distributed representations of data requires training the model over a dataset from the same domain with the training set.
- the best accuracy on the ML side is **0.914** without feature selection, while it is **0.913** with feature selection
- even the best accuracy is obtained by the NBM, higher results are often obtained by the LR and SVM classifiers.

(2) Performance of DL-based Classifiers: On the DL side, we only use word embeddings to build our learning models. This is because utilizing DL algorithms in the TC field depends on word embeddings and DL algorithms automatically discover features using these embeddings. Notice that we again use the feature set A and S on the DL side as in the ML side to explore the effect of feature selection on the performance of the DL algorithms. On this side, we use WE_{FB} word embedding models to obtain vectors of words as it was the most successful word embedding model on the ML side. We implement the DL algorithms using Keras library and run them on Colab which enables the research

community to use a powerful GPU without any charge. We named each experiment with a prefix *"ExDL"* on this side and performed the following experiments respectively:

- ExDL1: Performing CNN classification with different variants.
- ExDL2: Performing RNN classification with LSTM and GRU cells.
- ExDL3: Performing classification by combining the best RNN and CNN structures.

In ExDL1, we performed experiments with three variants of the CNN algorithm using different layer sizes of the WE_{FB} models. For this purpose, we first created a CNN structure (see Figure E.1) which is identical to Kim's CNN (2014) and used parameter settings of this CNN where we use: filter sizes of 3, 4, and 5 with 128 filters, a dropout rate of 0.5, and batch-size of 64. We also configured our CNN to run with 11 epochs based on our initial experiments which are showing that it often achieves the best result with 11 epochs. We then injected our WE_{FB} models in CNN-static and CNN-non-static variants. We also executed the CNN-rand with random embeddings where layer sizes are selected as the same with injected WE_{FB} models to make a fair comparison.



	100	200	300	400	500
CNN-rand + A	0.875	0.880	0.891	0.890	0.892
CNN-rand + S	0.895	0.894	0.901	0.900	0.893
CNN-static + A	0.906	0.907	0.906	0.906	0.908
CNN-static + S	0.909	0.908	0.914	0.912	0.910
CNN-non-static + A	0.878	0.876	0.906	0.910	0.908
CNN-non-static + S	0.909	0.907	0.911	0.911	0.910

Figure 5.12. Obtained accuracies for CNN variants with respect to the different feature sets and layer sizes in WE_{FB} models.

Figure 5.12 shows obtained accuracies for different CNN variants with respect to the different layer sizes and feature sets. As seen from the figure, variations produce slightly different results from each other, but the best accuracy is obtained as **0.914** by CNN-static by using the feature set S and layer size of 300. This shows that feature selection improves accuracy as observed on the ML side.

Besides, CNN reaches the best accuracy obtained on the ML side and outperforms the majority of the ML classifiers even without feature selection. An important thing to note here is that CNN achieves these results only by using 300-dimensional feature vectors, while ML algorithms achieve similar results on a huge number of features without feature selection and 1,500 features even with feature selection.

As on the ML side, these results show that the WE_{FB} model with a layer size of 300 is more successful in capturing semantic information between words. Therefore, we employ the WE_{FB} model with a layer size of 300 in the rest of our experiments.

In ExDL2, we use the RNN algorithm to perform SA experiments. Before running our experiments, we conducted a parameter optimization task at first by using Bayesian optimization with the help of the skopt package. We conducted optimization on a bidirectional RNN with LSTM cells to detect the number of layers and network parameters. We searched many parameters including the number of LSTM layer(s) and neurons for each LSTM layer, the number of dense layer(s) and neurons for each dense layer, whether to use dropout and batch normalization, dropout rate (if dropout is used), and optimizer.

After searching, we detected our optimum RNN structure to include three stacked bidirectional LSTM layers having 16 neurons each, a dropout rate of 0.2, an optimizer with *rmsprop*, without batch normalization, and a dense layer before the output layer. Figure E.2 depicts the summary of our optimum RNN structure after the parameter optimization process. Next, we performed initial experiments to detect the batch size and the number of epochs for better results. Based on these

experiments, we also detected that our optimum RNN provides better results with 2 epochs and a batch size of 8.

Using our RNN with these parameter settings, we performed experiments by changing the recurrent unit and type to explore their effect on results as well. Figure 5.13 depicts our results which are obtained with LSTM and GRU units with unidirectional and bidirectional types of our optimum network. Notice that instead of the regular LSTM and GRU, we use CuDNNGRU and CuDNNLSTM variants that can only be run on GPU with Tensorflow backend.

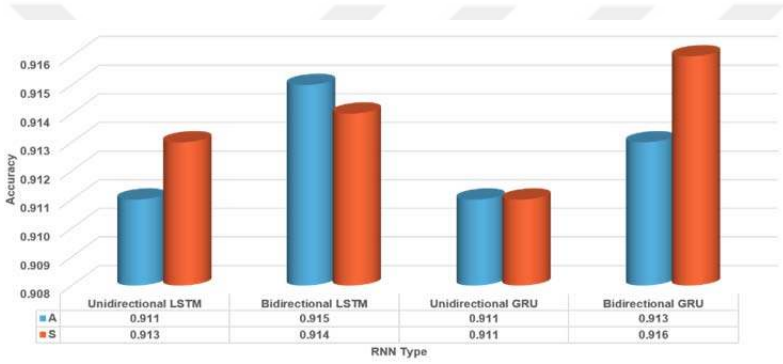


Figure 5.13. Obtained accuracies for our stacked RNN with respect to the type (unidirectional/bidirectional), cell (LSTM/GRU), and feature set (A/S)

As seen from Figure 5.13, results are slightly different, but bidirectional GRU achieves an accuracy of **0.916** with selected features and outperforms other configurations. These results make it clear that bidirectional RNN is better than unidirectional one and using feature set S (i.e., feature selection) improves accuracy. Even results are slightly different from the results of the CNN, our RNN outperforms CNN and other ML classifiers even without feature selection.

In ExDL3, we combined CNN and RNN algorithms by using the optimum structure and parameter settings from the previous two experiments. As stated before, bidirectional GRU and CNN-static are the best performant variants in previous experiments. Therefore, we combined these two variants and obtained

CNN-GRU and GRU-CNN models (see Figure E.3 and Figure E.4) with the same parameter settings. We configured these combined (or hybrid) models to run with 3 epochs and a batch size of 8. Figure 5.14 depicts obtained results which show that combined models produce lower results when compared to the results of their single runs.

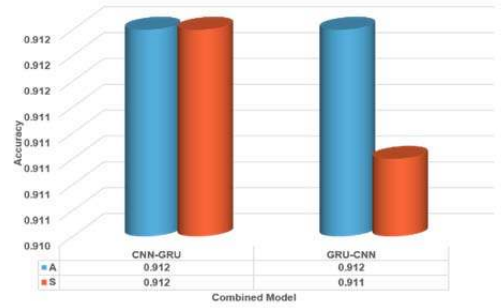


Figure 5.14. Obtained accuracies with combined models for different feature sets

Our overall experimental results on the DL side present the following findings in summary:

- feature selection improves accuracy as on the ML side.
- the static variant of the CNN reaches the best accuracy (i.e., **0.914**) obtained on the ML side.
- bidirectional RNN with GRU cells achieves an accuracy of **0.916** which is the best result obtained in our SA task.
- combining CNN and RNN algorithms has no positive effect on the results.

5.1.3.2. Sentiment Analysis of Facebook Activities

In this subsection, we use the evaluation dataset (see Table 3.2) to extract statistics which we believe are one of the main contributions of our SA task. For this purpose, we use the best learning model (i.e., bidirectional RNN with GRU cells) and feature set (i.e., the feature set S) on the WE_{FB} model with a layer size of

300 which are detected in the previous step of our experiments. We aim to explore the followings:

- Polarity distribution of activities with respect to gender and age-range of users, month, and hours.
- Whether Facebook reactions can reflect the sentiment of activities.
- Whether users are more engaged with positive or negative activities.

To achieve these goals, we predicted the polarity of each activity in the evaluation dataset. A sample list of activities with gender and age information of their writers is presented in Table E.1 where the polarity of each activity is predicted by using the best learning model.

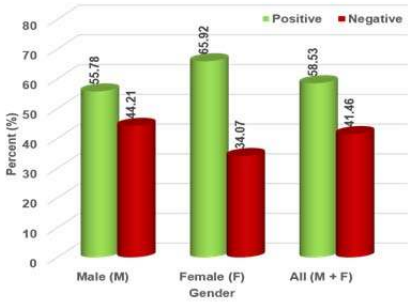
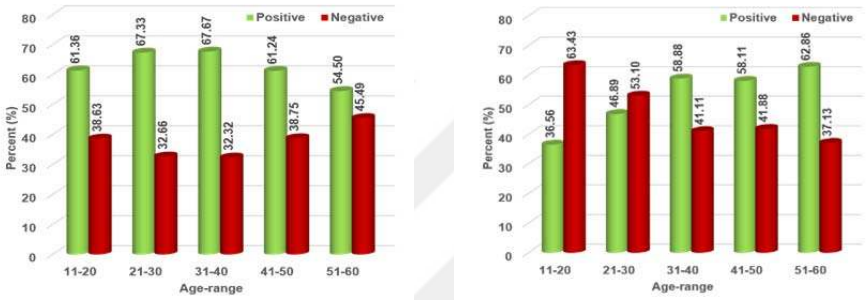


Figure 5.15. Polarity distribution of activities in the evaluation data with respect to the gender of users

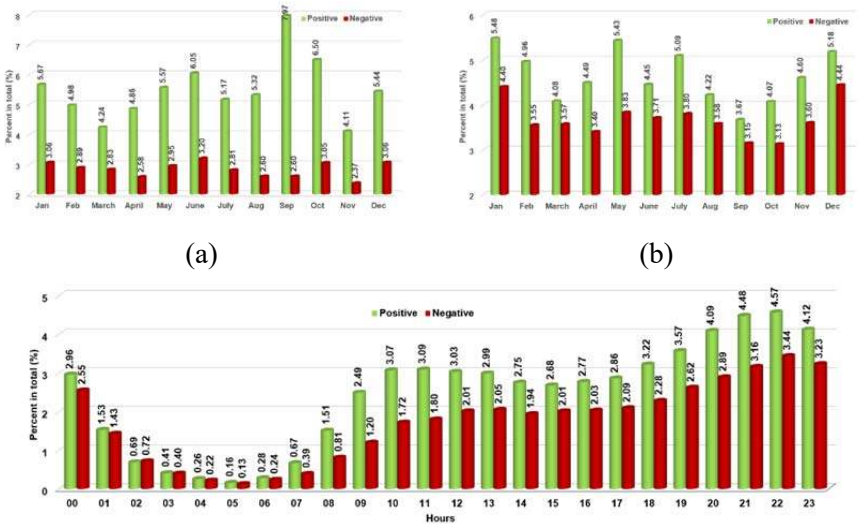
In the first step, we extracted the polarity distribution of activities, months, and hours. Figure 5.15 presents the polarity distribution of activities written by all users, male, and female users, respectively, while Figure 5.16a and Figure 5.16b show percentages of positive and negative activities of female and male users of different ages. As seen in Figure 5.15, the majority of activities often have positive polarity for both male and female users. However, the difference rate between positive and negative activities is low for males when compared to females.

On the other hand, Figure 5.16a and Figure 5.16b show that female users (at any age-range) tend to post positive activities, while the reverse is true for male users at age-range 11-20 and 21-30 years. Male users often post more positive activities as they get older, while the reverse is true for females. The most negatory users among female users are at age-range 51-60, while those are at age-range 11-20 for males.



(a) (b)

Figure 5.16. Polarity distribution of activities with respect to the age-ranges of writers: (a) for females and (b) for males



(a) (b)

(c)

Figure 5.17. Polarity distribution for activities of users: (a) for females over months, (b) for males over months, and (c) for all users over hours

We then investigated the polarities of activities over months for both male and female users. Figure 5.17a and Figure 5.17b present the results of this experiment. The sharing trend over months has similar behavior for male and female users. Females share more amount of positive activities in September, while males share in January. On the other hand, females post more negative activities in January and December, while males post in December.

As the last step, we then also explored the distribution of all activities over hours with their polarities. As seen from Figure 5.17c, distributions of positive and negative activities have similar trends over hours and the number of activities decreases both during working hours and after midnight.

Table 5.4. Percentages of Facebook reactions for activities of male, female, and all users.

Polarity	Facebook Reactions					
	wow	angry	sad	haha	love	like
			All Users			
Positive	0.079	0.488	1.138	1.048	1.792	95.45
Negative	0.058	0.126	0.239	0.370	1.642	97.56
			Males			
Positive	0.070	0.515	1.186	1.131	1.747	95.34
Negative	0.060	0.154	0.255	0.411	1.479	97.63
			Females			
Positive	0.137	0.309	0.816	0.486	0.924	96.15
Negative	0.049	0.033	0.182	0.227	2.201	97.30

In the second step, we dig into the relationship between polarities and reactions of activities. Notice that we eliminated activities which do not have any reaction from the evaluation data before performing this experiment and obtained a sub-dataset that includes 87.4K activities. Results of this experiment are given in Table 5.4 which makes it clear that *"like"* is the most preferred reaction, while

"wow" and *"angry"* are the least used reactions by users. If we ignore the *"like"* reaction, the reaction that users tend to mostly use is the *"love"* reaction.

When we consider all activities, our results show that there are small differences between the percentages of the reactions in positive and negative activities. There is also a similar trend in Facebook reaction distribution for activities written by both male and female users. For male users, only the *"like"* reaction has a higher percentage of observability in negative messages than in positive messages, while the reverse is true for other reactions in positive activities. On the other hand, for female users, the *"love"* and *"like"* reactions have higher percentages of observability in negative messages than in positive messages.

Another interesting thing to dig into deeper is exploring the engagement of users on activities. As such, finally, we extracted several statistics for child activities of parent activities which are contained in our evaluation dataset.

Table 5.5. ACA and PDCA values for responses of parent activities in the evaluation data with respect to the parent activity type and user gender.

Parent Activity Type	**Polarity**							
	Positive (Pos)				Negative (Neg)			
	Count	Responses			Count	Responses		
		ACA	PDCA (%)			ACA	PDCA (%)	
			Pos	Neg			Pos	Neg
All Users								
Post	36,731	1.777	73.97	26.02	35,543	1.970	45.94	54.05
Comment	28,380	0.650	85.81	14.18	18,550	0.863	77.18	22.81
Males								
Post	25,296	1.921	71.93	28.06	27,893	2.063	44.24	55.75
Comment	19,388	0.694	84.51	15.48	14,234	0.872	78.49	21.50
Females								
Post	11,435	1.457	79.94	20.05	7,650	1.629	53.86	46.13
Comment	8,992	0.556	89.07	10.92	4,316	0.835	71.72	28.27

These statistics are the average number of child activities (ACA) and the polarity distribution of these child activities (PDCA). Notice that in this step, we

consider replies and comments with their relative replies as child activities of comments and posts respectively. Table 5.5 presents our results which show that the number of child activities under the negative parent activities is greater than the positive ones for both males and females.

The average number of child activities for parent activities by males is also greater than the average value for females. Further, it is clear that the majority of child activities under positive parent activities of males are positive and this is also true for females. On the other hand, negative activities often have positive child activities for both males and females. These results show that users are more engaged with negative parent activities. Parent activities of male users have a higher number of child activities when compared to those of female users.

Our overall results and experienced challenges in the SA task show that lexicon-based feature generation does not improve the SA results. The main reason for this is that formal lexicon words do not match with any informal words in the activities which often contain grammatical errors. The word embedding model, on the other hand, can capture more semantic information when it is trained on a corpus from the same domain. This result has been observed since when the training dataset is chosen from the same domain as the test set, it is more probable that both sets include similar words, phrases, or sentence structures.

For SA, using DL models is a better option to build a learning model as they often produce better results than the ML methods. DL models also employ a much lower number of features (i.e., word embeddings) with respect to ML models applied with bag-of-words features. We suggest testing this finding for other languages as well. We would also like to note that further improvements are also possible if DL algorithms run over a better-distributed representation of the data to learn long-term dependencies and capture local features. It should be kept in mind that determining both the optimum structure and parameters of the network for DL algorithms is a challenging task. Nevertheless, parameter optimization is very helpful to achieve such a task as we observed in this task.

Our SA task involves two main conceptual novelties which are: (i) we applied feature selection for both DL and ML methods, although feature selection is a common practice only for the ML side, (ii) we build our own RNN structure which achieves the best result in this task.

SA of Facebook activities shows that majority of the posted activities are positive, but this is not true for male users at age range 11-20 and 21-30. Based on this interesting output, users in this group can be said to have more aggressive and negative thoughts. Another interesting finding is that negative posts receive more likes compared to positive ones, and there is a slight difference between the percentages of Facebook reactions for positive and negative activities. Some of the reactions have a conflict with the polarity of the messages. For instance, love reaction is expected to have a higher percentage in positive messages, but in our case, this is not true for male users. This is because users may use positively/negatively oriented reactions for messages related to negative/positive cases as well.

5.2. Attribute Inference and Privacy Risk Scoring

This subsection presents the results of our experiments which are related to our attribute inference and privacy risk scoring tasks.

5.2.1. User Profile Matching

Matching accounts across different OSNs is helpful in different aspects, but it may be considered as an adversarial attack when it is performed with nefarious intentions. This section presents the results of our user matching task based on usernames. User matching is often solved by building a learning function f. Let u_1 and u_2 represent two distinct usernames in two different OSNs respectively. The learning function f can be formulated as follows (Zafarani and Liu, 2013; Wang et al., 2016):

$$f(u_1, u_2) = \begin{cases} 1, & if\ u_1\ and\ u_2\ belong\ to\ the\ same\ user \\ 0, & otherwise \end{cases} \quad (5.1)$$

In this thesis, we learn the f function in two different approaches. We first try to match users by using a simple model that gives its decision based on the similarity between the two usernames. Secondly, we employ different ML classifiers to build a binary classification model. Finally, we build a combined model that uses ML classifiers on a combined (i.e., extended) list of features.

In the first approach that we call similarity comparison, the f function uses an inner similarity or matching method to compute a score on feature vectors of which of each one stands for a single username. Afterward, the f function returns 1 or 0 depending on the score calculated by the inner method. The f function, that gives its decision by comparing the computed similarity score with a predefined threshold value, is shown as follows (Wang et al., 2016):

$$f(u_1, u_2) = \begin{cases} 1, & if\ sim(u_1, u_2)\ \geq \tau \\ 0, & otherwise \end{cases} \quad (5.2)$$

where $sim(u_1, u_2)$ is the inner similarity method and τ is the predefined threshold value. To compute the inner similarity between two usernames, we represent usernames by using both binary and self-information vector models (or weighting schemes) that are previously described in Section 4.6.

In the binary classification approach, on the other hand, the username dataset is converted into a classification-ready structure. In this structure, each instance is represented as a feature vector that each dimension stands for a value of each of the extracted features. Afterward, a binary classifier is trained on this dataset to build the learning function f. Upon completion of the training, the binary classifier is expected to output whether a new pair of usernames that constitute a negative or positive instance match or not. In this task, we employ the binary

classification approach with the help of Weka's well-known ML algorithms including NB, SMO, RF, J48, SVM, and NBM.

The matching task is performed on the username dataset (see Section 3.3.4) that includes both positive and negative pairs of usernames. Notice that a positive pair is composed of two different usernames that belong to the same user, while a negative pair includes the usernames of two different users. To obtain classification results, the dataset is passed through a simple preprocessing in which lowercase conversion and punctuation removal steps are applied to usernames. Then, as stated above users are matched by using two different approaches.

We first applied the similarity comparison approach with the feature set given in Table 4.7. To obtain the results of this approach, we used different threshold values to observe the optimum value. We also represented usernames by using the self-information vector model to compare it against the binary vector model. Figure 5.18 presents our results obtained with the similarity comparison approach.

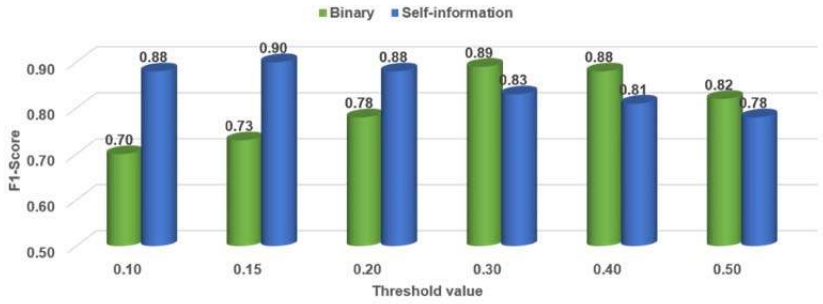


Figure 5.18. Results of the similarity comparison approach with respect to different vector models and threshold values.

As seen from Figure 5.18, matching success tends to increase as threshold value increases in the binary vector model, while the opposite is true for the self-information model. The binary vector model achieves the best F1 score of 0.89 when the threshold is set to 0.3, while the self-information model obtains the best

F1 score of 0.90 with a threshold of 0.15. This shows that the results of two vector models are slightly different, but the self-information model provides a better result than the binary representation of feature vectors.

Secondly, we employed the binary classification approach using the feature set summarized in Table 4.8. In this phase, we trained different classifiers and investigated the effect of the selected classifier on the performance. We used default parameter settings for all of the classifiers which were configured to run with 10-fold cross-validation. Figure 5.19 presents our results obtained with the binary classification approach. As seen from Figure 5.19, the F1 scores of classifiers are slightly different, but the most successful classifier is SMO with the best F1 score of **0.921**. This shows that the binary classification approach outperforms the similarity comparison approach even though it uses far fewer features.

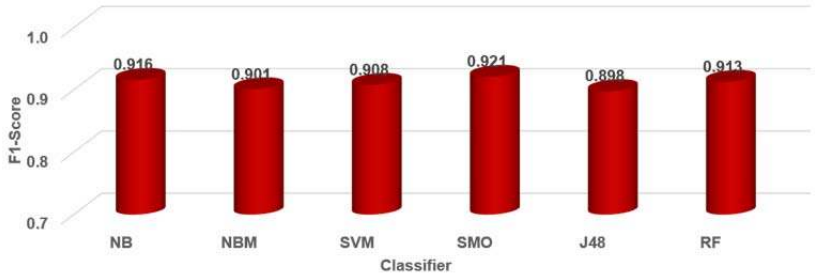


Figure 5.19. Results of the binary classification approach with respect to different classifiers.

Next, we employed the combined model that uses the extended feature space obtained by concatenating features of Table 4.7 and Table 4.8. We would like to note that ML classifiers require one feature vector per instance. As such, we have to convert two feature vectors into a single instance vector when we employ features of Table 4.7. In order to do so, we take the average ($\bar{x}$) or sum (+) of each feature's values and created our final instance vector. We then combine this final instance vector with the features of Table 4.8 to obtain our extended feature space.

This simple process is depicted in Figure 5.20. Notice that we represent features of Table 4.7 by using binary or self-information vector models and create a final instance vector by taking the sum or average of feature vectors so as to explore effects on classification performance.

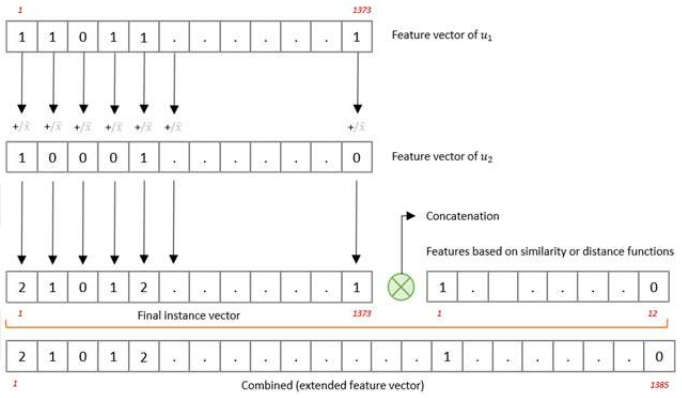
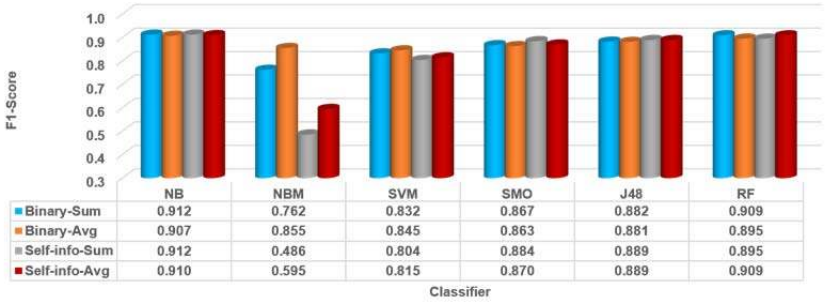


Figure 5.20. Feature extension in the combined model

Our combined feature vector now consists of 1,385 features. Using the combined model, we obtained results for different classifiers with 10-fold cross-validation.



	NB	NBM	SVM	SMO	J48	RF
Binary-Sum	0.912	0.762	0.832	0.867	0.882	0.909
Binary-Avg	0.907	0.855	0.845	0.863	0.881	0.895
Self-info-Sum	0.912	0.486	0.804	0.884	0.889	0.895
Self-info-Avg	0.910	0.595	0.815	0.870	0.889	0.909

Figure 5.21. Results of the combined model with respect to the classifier, feature representation model, and conversion operator.

As seen in Figure 5.21, the combined model does not improve the best F1 score of the binary classification approach. The best F1 score is obtained as a value

of 0.912 with the help of the NB classifier, even though classifiers produce slightly different results. Taking the average of vectors of usernames provides slightly different results than the sum operator. On the other hand, the binary representation of username feature vectors seems to be a better choice than using the self-information vector representation. Excluding the NBM, it is clear that classifiers often produce slightly different results. NBM classifier is mostly the worst one as it works well for data that can easily be converted into frequency values, such as word counts in text.

Finally, we explored the most important features by using the mutual information (Peng et al., 2005) method on the best performant representation of data, where features of Table 4.7 are represented by the binary vector model and were converted to a final instance vector using the sum operator. This final instance vector is then concatenated with the corresponding vector of features that were extracted with functions of Table 4.8. Using this best representation, we obtained feature dependencies with respect to the mutual information scores. Figure 5.22 depicts the top 20 most important features.

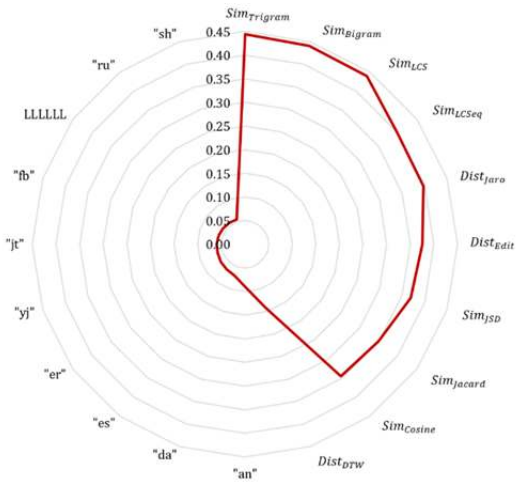


Figure 5.22. A spider chart of top 20 features with respect to their mutual information scores.

As seen from Figure 5.22, the most important features depend on similarity or distance functions including n-gram similarities, the similarity between the longest common substrings (Sim_{LCS}), the similarity between the longest common subsequences (Sim_{LCSeq}), Jaro-Winkler distance ($Dist_{Jaro}$), edit distance (Sim_{Edit}), and so on. Additionally, the LLLLLL feature is the most important one among letter digit pattern features. It is also clear that bigram patterns of letters (i.e., *"an"*, *"da"*, *"es"*, and so on) are among the most important features.

Using these mutual information scores, we build a simple yet effective experiment scenario that involves incorporating a feature selection process in our classification task. We selected the top-n features based on their mutual information scores. We then used the selected features in the classification phase. We used the NB classifier because it was the most successful classifier in the combined model. Figure 5.23 shows the results of this final experiment where the number of features to select (i.e., n) takes different values in a range between 10 and 1,000.

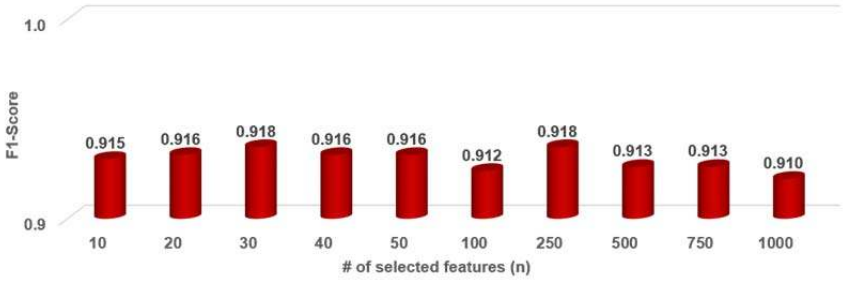


Figure 5.23. Results of the NB classifier with respect to the number of selected features.

As seen in Figure 5.23, feature selection makes a very low contribution to the best result of the combined model. When the number of selected features is 30 or 250, the NB classifier obtains its highest result of 0.918, which is higher than the best F1 score (i.e., 0.912) of the combined model employed without feature selection. On the other hand, the best F1 score obtained with the feature selection

falls behind that of the best F1 score (i.e., 0.921) of the binary classification approach employed without feature extension.

In this task, we studied the problem of user matching across different OSNs. In order to do so, we first created a dataset by crawling Facebook which is one of the most popular OSNs. As users are able to disclose their other accounts on their Facebook profiles, we could collect the other OSN accounts of users. We then performed username-based profile matching which may fail in many cases, since users may use different usernames and, in some cases, the username is not available. Additionally, in some OSNs like Foursquare, the username is a numeric string, which is automatically generated by social sites (Li et al., 2018). These reasons make the username based matching method fragile.

On the contrary, even a matching system uses other information (e.g., profile attributes, wall contents, etc.), cross-referencing users across different OSNs is not an easy task as users may fill their profiles with different (even fake) details. Therefore, username-based matching needs to be studied further and it is more preferable to use such a system as a part of a complete matching system.

5.2.2. Private Attribute Disclosure

This subsection presents the results of our kinship and gender detection tasks that each of which can actually be considered as an attribute inference task.

5.2.2.1. Kinship Detection

As stated before, Facebook users mostly prefer this platform to keep their interactions with their friends and family members. As seen in Figure 1.5, users can disclose their kinship relations on Facebook. This type of disclosure is mostly observed in profiles of privacy-unaware users and it may lead to different privacy threads. Let consider an arbitrary Facebook user *Senem* and her family and

relationship interactions which are shown in Figure 1.5. In this case, different privacy threads may occur.

For instance, if an adversary has access to the account of *Senem*, it can infer that *Yaren* is the daughter of *Senem* and there is a child-parent relation between them. Notice that *Senem* disclosed the sensitive information of *Yaren* even if she does not want to reveal this information. It is possible to extend this inference for other disclosed members as depicted in Figure 5.24.

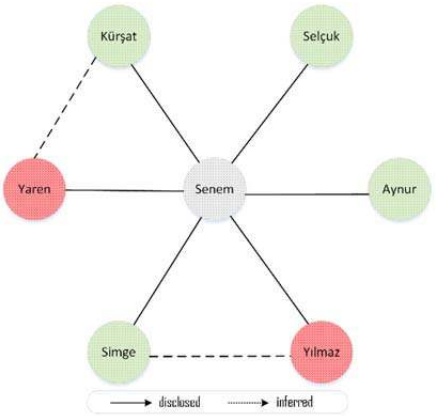


Figure 5.24. An undirected graph for kinship interactions among users from Figure 1.5.

Possible inferences, in this case, are as follows: (i) *Kürşat* is the father of *Yaren*, (ii) *Yılmaz* is the cousin of *Simge*, (iii) *Aynur* may be the sibling of *Selçuk*, (iv) *Simge* and *Yılmaz* may be the child of *Selçuk* or *Aynur*. Notice that Figure 5.24 only depicts (i) and (ii) relations as others require verification based on additional steps. Another possibility is that adversary can easily learn the mother's maiden name of *Yaren*. This is because *Selçuk* is the uncle of *Senem* and his surname enables the adversary to learn and use mother's maiden name of *Yaren* for malicious activities.

As stated in Section 2.3, many previous studies aim to infer different attributes of Facebook users. However, content-based kinship inference has not been evaluated in the context of any OSNs including Facebook. Facebook users

communicate with each other via short text messages (i.e., activities) and often use kinship words in their activities as depicted in Figure 5.25, where kinship words in Turkish are underlined with red ink.

Therefore, the question guiding this task is *"is it possible to detect kinship between two Facebook users?"*. To explore this question, in this task, we first created our base user set that includes kinship pairs of Facebook users. Next, we crawled the wall activities of all users in this set as this task aims to detect kinship between two users based on wall contents. Even this task is related to other well-known tasks such as kinship determination and user relationship classification, it seems that this is the first research into fine-grained kinship detection of Facebook users.

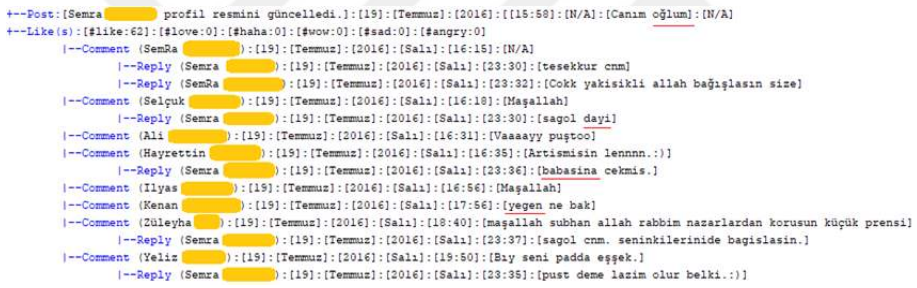


Figure 5.25. Screenshot of our crawler for an arbitrary post and its relative comments and replies.

This section presents the experimental results of the kinship detection task which are obtained by our baseline and lexicon-based matching model. Notice that in this task, we configured our models including baseline to run with 10-fold cross-validation and we measured the performance of the models using the weighted-average F1 score. In the first step of our experiments, we use the Naïve (i.e., Dummy) classifier as our baseline model to obtain a lower bound on the expected performance of the lexicon-based matching model. As our kinship datasets are imbalanced, we employ the Naïve classifier with different strategies and obtained results with respect to the use of oversampling and dataset.

As seen from Table 5.6, oversampling, which is performed using the SMOTE algorithm, does not have a positive effect on results and stratified classification strategy provides better results. This is because oversampling only changes the number of instances and the Naïve classifier makes its decision randomly. On the other hand, the baseline model achieves better results on datasets that include inferred kinship relations. Our baseline model produces the best result as an F1-score of 0.25 on the M_{RI} dataset. An important thing to note here is that applying over/under-sampling in a correct way during cross-validation requires applying the related method to the training set only. Then it is needed to evaluate the model on the stratified but non-transformed test. In this task, we achieve this by defining a pipeline that first transforms the training set with SMOTE and fits the model. We then use this pipeline to evaluate the model on non-transformed test data.

Table 5.6. F1 scores for baseline model with respect to the dataset, oversampling, and classification strategy

Classifier	**Over Sampling**	**Dataset**			
		S_R	S_{RI}	M_R	M_{RI}
$Naive_{stratified}$	✗	0.22	**0.23**	0.24	**0.25**
	✓	0.13	0.14	0.18	0.19
$Naive_{most\_frequent}$	✗	0.17	0.17	0.17	0.17

In the second step of our experiments, we employed our lexicon-based matching model which is mainly based on features (see Section 4.4.2.3) which are extracted from activity contents. Notice that performing kinship detection based on activity contents may be achieved by using classical TC techniques. However, in that case, we first need a labeled corpus to train the learning model, and annotating the corpus is a challenging task. Besides, it would be too hard to extract discriminative features from contents as fine-grained kinship verification only can be achieved by focusing on kinship-oriented words. In addition, as stated before in

previous sections, extracting meaningful information from Facebook activities has the following main challenges:

- It is very difficult to detect who responded to whom especially when the number of child activities high.
- Detecting the topic of any post is not easy to achieve and users may write about anything in their posts. For instance, in Figure 5.25, the wall owner (i.e., *Semra*) has shared a photo of her son, and other users mostly write about her son, not herself. This makes it too hard to extract meaningful information about the kinship relation of WO with other users.

Therefore, in this task, we use a hybrid lexicon-based ML model, which utilizes kinship features and transforms the dataset into a classification-ready structure to build an ML classifier. In this model, we represent each kinship relation with a vector such that each dimension of an instance vector corresponds to the total frequency of kinship words in related types observed in activity contents.

Let A and B be two Facebook users and A is the *"Parent"* of B. Let A uses 1, 2, 1, and 5 word(s) which refer to the cousin, sibling, parent, and child kinship types respectively in his/her activities targeting B. Similarly, let B uses 2, 6, 1, and 1 word(s) which refer to the same kinship types respectively. In the S_R and S_{RI} datasets, this relation is represented by forming two separate instance vectors by oppositely concatenating these feature values. In this case, the first instance vector would be $\langle 1, 2, 1, 5, 2, 6, 1, 1, \mathrm{S2}\rangle$, whereas the second vector would be $\langle 2, 6, 1, 1, 1, 2, 1, 5, \mathrm{S1}\rangle$.

In the M_R and M_{RI} datasets, on the other hand, we represent this relation by forming a single instance vector such that each dimension is the sum of the related feature values. In this case, the mutual instance vector would be $\langle 3, 8, 8(2+6), \mathrm{M7}\rangle$. We would like to note that we also combine features into their mutual equivalents

(see Section 3.3.5) as we do for instance vectors. After transforming the kinship datasets into a classification-ready structure, we build our lexicon-based matching model with help of well-known ML classifiers such as RF, LR, NB, DT, KNN, and SVM.

In this second step of our kinship detection experiments, we used the lexicon-based matching model in different cases where the classifier, filter, sampling technique, and dataset are different. We first explored the effect of filters which are employed in the feature extraction phase and obtained results on the kinship datasets with WF, GF, and SF filters (see Section 4.4.2.3) respectively.

Table 5.7. F1 scores for the lexicon-based matching model with respect to the filter, classifier, and dataset.

Filter	**Classifier**											
	RF		LR		NB		DT		KNN		SVM	
	S_R | S_{RI}											
WF	0.31	**0.41**	0.22	0.25	0.24	0.26	0.28	0.38	0.28	0.35	0.19	0.23
+ GF	0.33	0.39	0.22	0.26	0.25	0.28	0.30	0.36	0.28	0.33	0.22	0.26
+ SF	0.33	0.40	0.22	0.26	0.25	0.27	0.30	0.36	0.28	0.33	0.21	0.26
	M_R | M_{RI}											
WF	0.33	**0.39**	0.22	0.23	0.24	0.26	0.32	0.36	0.30	0.33	0.20	0.22
+ GF	0.32	0.37	0.22	0.25	0.24	0.26	0.32	0.35	0.28	0.33	0.22	0.25
+ SF	0.34	0.39	0.23	0.26	0.24	0.26	0.32	0.36	0.31	0.35	0.21	0.26

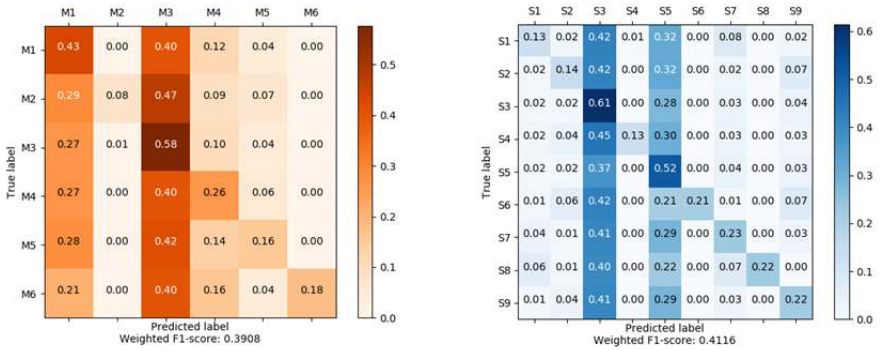
Table 5.7 presents our results which show that datasets including inferred kinship relations provide better results again. Classifiers are not sensitive to any filter and the most successful classifier is the RF, while the worst one is often SVM. Our lexicon-based hybrid ML model achieves the best results as F1-scores of 0.41 and 0.39 on the S_{RI} and M_{RI} datasets respectively.

Table 5.8. F1 scores obtained by lexicon-based hybrid matching model utilized with sampling techniques

Sampling		Dataset	
Over	Under	S_{RI}	M_{RI}
✓	✗	0.40	0.38
✓	✓	0.39	0.38

These results show that our model outperforms the baseline model by using the RF classifier. Besides, we performed experiments to explore the effects of sampling techniques on the results of the lexicon-based hybrid ML model. In this phase, we run the most successful classifier RF on the S_{RI} and M_{RI} datasets that provide the best results in the previous experiments. As seen from Table 5.8, sampling techniques do not improve the performance of the lexicon based-model with their single or combined runs.

Notice that over and under-sampling is applied with the help of the SMOTE (Chawla et al., 2002) and Tomek Links (Tomek, 2010) algorithms respectively (see Section 4.7).



(a) (b)

Figure 5.26. Normalized confusion matrices for the best classification experiments conducted on (a) S_{RI} and (b) M_{RI} datasets with the help of the RF classifier.

Finally, to give deeper insight, we also present a confusion matrix for our best classification experiments in Figure 5.26 which shows that one or two sides (i.e., mutual) relations with the type of sibling and cousin are more distinguishable classes in the kinship datasets. Nevertheless, it is still too hard to classify instances into their actual classes.

Facebook users often share their kinship and private relationships which may cause privacy and security risks. Such information can be learned by analyzing the wall contents of users even they do not reveal their family members and relationships. As such, in this task, we studied automatic kinship detection of Facebook users based on their wall contents. To achieve this goal, we proposed a lexicon-based matching approach that gives lexicon-based features as inputs for ML classifiers. Our results show that there needs large improvement especially in the context of extracting valuable information from wall contents. Performing this task on wall contents is very challenging as creating a ground-truth base user set is quite hard as it requires discovering a greater number of kinship pairs in OSN. However, even if this is met, the majority of pairs possibly will be discarded from the base set as some users keep their wall pages private or do not share anything.

Facebook users interact with each other by typing messages under any post by their friends or other sources such as liked pages. This makes it quite hard to detect the absolute target user of an activity especially when the number of relative activities is too high. Even if the targeted user is detected, there are still some other challenges that stem from the Turkish language. This is because users often write activities using informal language and without adhering to any grammatical rules. Additionally, it is possible to use kinship-oriented words in a different sense in Turkish. To detect the sense of a word, word sense disambiguation or fine-grained NER can be employed, but these tasks require applying NLP steps on large training sets.

Applying gender and surname filters have some drawbacks. This is because Facebook users may not reveal their genders and real surnames and it is

too hard to detect the correct value of this information. Finally, Facebook users may reveal any of their friends as their family members, especially as a sibling. It is quite challenging to detect whether these users are family members in real life.

5.2.2.2. Gender Detection

This section presents experimental results that are related to our gender inference task. In this task, we created a user set to use in accordance with our task at first. To create the user set, we select the 20K snapshot of crawled users and applied the following filters:

- User must make his/her wall page and gender attribute public.
- There must be at least one activity written by WO on his/her own wall.
- In the wall of WO, there must be at least one activity, posted by any of his/her friends, that targets WO.

As seen in Table 5.9, applying these filters have resulted in 8,563 male users and 4,211 female users. Afterward, we selected an equal number of male and female users to create our balanced final user set that contains 4,211 users in two gender categories.

Table 5.9. The number of users in the 20K snapshot with respect to applied filters

Filter / Criterion	# of users		
	Male	Female	Total
Public gender (see Figure 5.1)	9,800	6,486	16,286
+ Public wall	9,323	6,070	15,393
+ Wall content	8,563	4,211	12,774
= Final user set	4,211	4,211	8,422

After creating our user set, we employed four different models to infer the gender attribute of Facebook users. These models are as follows:

- L1: This is our baseline model that makes its decision by random prediction. L1 model randomly predicts the gender of the related user without using any extra information.
- L2: This model uses profile and social interaction (i.e., mainly relationship and friendship connections) information of users. In this model, we extract and use features which are presented in Table 4.6.
- L3: This model is based on the wall contents of users. The extracted information depends on whom WO answers, who answers WO, and which users tagged with WO in any activity. In this model, we use features that are summarized in Table 4.4 and extracted by a lexicon-based approach that counts the number of gender-oriented words in activities to extract features.
- L4: This model uses wall contents for gender inference, but differently from the L3 model, it uses a basic idea that male and female users use different words, phrases, and emoticons in their activities. L4 model is trained and tested on the wall datasets (see Section 3.3.2.3) using classical TC techniques.

To build a learning model, we use Weka's well-known classifiers which are NB, NBM, SVM, J48, RF, and IBk. All classifiers are used with default parameter settings and models are evaluated by a 5-fold cross-validation technique. Performance evaluation, on the other hand, is performed by using the accuracy (see Section 4.8.3.2) metric. First, we obtained results with the L1 model which is our baseline, and presented related results in Figure 5.27.

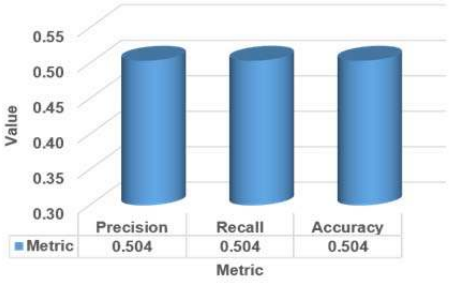


Figure 5.27. Average values of evaluation metrics for the L1 model

As we expected, the L1 model achieves an accuracy of 0.504 on our user set. In the second step, we performed experiments using the L2 model with well-known classifiers. As seen in Figure 5.28, classifiers produce slightly different results from each other. Nevertheless, the best result is obtained by the RF classifier, while the worst is obtained by the NBM classifier. In the L2 model which uses profile information of users, we achieved an accuracy of 0.981 as compared to 0.504 which is obtained by our baseline.

Next, we performed experiments using the L3 model that extracts features from various information (e.g., writer, the gender of the writer, etc.) of the activities (not from their contents) found on WO's wall. It also makes use of male and female-oriented words in activities to extract features. Notice that in this model, only gender-oriented words are considered not all content of the activities used for feature extraction.

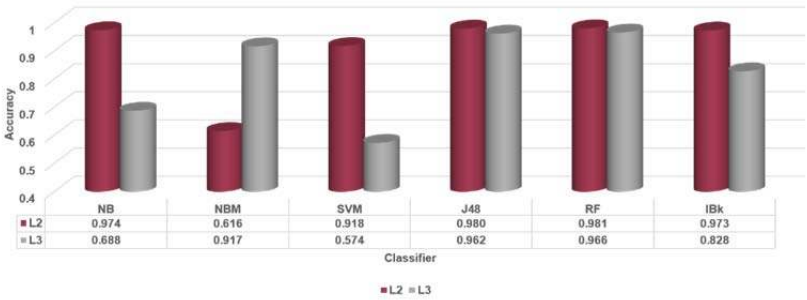


Figure 5.28. Obtained accuracies for L2 and L3 model with respect to different classifiers.

As seen in Figure 5.28, the best result is obtained again by the RF classifier, while the worst is obtained by SVM. In the L3 model, we achieved an accuracy of 0.966 as compared to 0.504 and 0.981 which are obtained by the L1 and L2 models respectively.

Finally, we used the L4 model for gender inference of users. As stated before, this model utilizes TC methods and uses bow and character level n-gram features which are extracted from the activity contents. In this model, we use the wall datasets and explore which type of activity is better to infer WO's gender in the content-based analysis. The experiments are conducted in two different ways in this model.

On the wall datasets, we first applied both basic and linguistic preprocessing to remove uninformative content. For results of the L4 model, the results which are obtained with linguistic preprocessing shown with ✓, while results obtained with basic preprocessing shown with ✗. After preprocessing, we used classical TC steps to perform the gender inference task. Firstly, we obtained bow and n-gram features and applied term weighting with three different schemes including binary, tf, and tf*idf (Section 4.6). Table 5.10 presents the number of features with respect to the feature model, preprocessing level, and wall dataset in the first way.

Table 5.10. The number of unique features obtained from the wall datasets with respect to preprocessing level, feature model, and document set

Feature Set	**Preprocessing**	**Document Set**		
		WD1	WD2	WD3
Bow	✗	223,577	594,695	480,009
	✓	24,117	62,295	48,651
Bi-gram	✗	990	1,020	1,021
Tri-gram	✗	11,222	12,474	11,974

Table 5.10 shows that content-based analysis with the classical bow and n-gram features poses a challenge that it has to handle too high vector space

especially in the bow model with basic preprocessing. Notice that we applied linguistic preprocessing only for feature models that use word-level information. Basic preprocessing, on the other hand, is only applied to character-level feature models. Following the feature extraction, in the first way, we performed classification on the wall datasets and obtained results which are presented in Table 5.11.

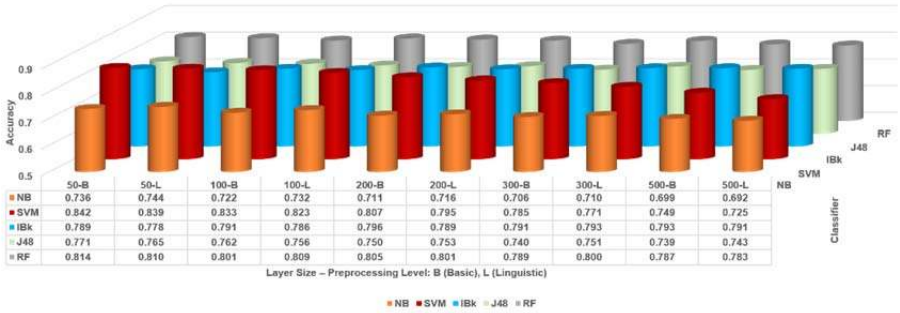
As seen from Table 5.11, we obtained the best F1-scores as 0.862, 0.826, and 0.836 on the WD1, WD2, and WD3 datasets respectively. The most successful classifier is generally NBM, while the worst one is the IBk. Among the weighting schemes, tf often produced better results than the others. In the n-gram model, there is a direct proportion between length and classification performance and tri-grams generally outperform the bi-grams. In all cases of the first way in the L4 model, the best accuracy (i.e., 0.862) is obtained on the WD1 dataset that includes activity contents targeting WO. These results in the first way show that classical feature models fall behind that of the L2 and L3 models compared to their accuracies of 0.981 and 0.966 respectively.

In the second way, we used distributed representations of words to represent documents in the L4 model. To create distributed representations of the wall datasets, we first trained our word2vec model on the word embedding data (see Section 3.3.2.1) which includes all activity contents of 20K of users. Notice that in this step, we use the DL4J tool to build the distributed representations of data and use different layer sizes n where $n \in \{50, 100, 200, 300, 500\}$ to explore its effects on results.

Table 5.11. Obtained accuracies for the L4 model with respect to the feature set, classifier, dataset, and weighting scheme

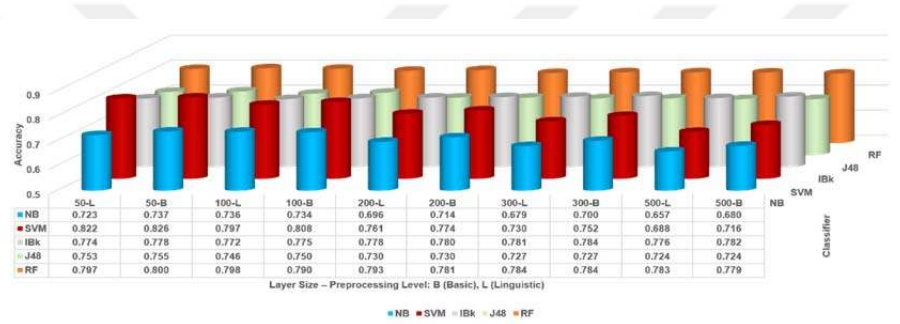
CLS	Feature Set	WD1			WD2			WD3		
		binary	***tf***	***tf*idf***	***binary***	***tf***	***tf*idf***	***binary***	***tf***	***tf*idf***
NB	*bow*	0.634	0.629	0.640	0.596	0.597	0.597	0.629	0.592	0.592
	2-gram	0.597	0.629	0.628	0.607	0.642	0.646	0.627	0.683	0.689
	3-gram	0.650	0.684	0.682	0.643	0.663	0.662	0.683	0.716	0.716
NBM	*bow*	0.830	**0.862**	0.843	0.823	**0.826**	0.820	0.823	**0.836**	0.828
	2-gram	0.736	0.790	0.750	0.691	0.713	0.718	0.754	0.797	0.771
	3-gram	0.821	0.830	0.822	0.750	0.737	0.757	0.811	0.815	0.815
SVM	*bow*	0.613	0.718	0.741	0.585	0.690	0.696	0.621	0.720	0.748
	2-gram	0.806	0.811	0.773	0.759	0.758	0.737	0.807	0.803	0.781
	3-gram	0.819	0.816	0.844	0.764	0.777	0.786	0.790	0.794	0.823
J48	*bow*	0.804	0.806	0.806	0.774	0.787	0.765	0.804	0.804	0.802
	2-gram	0.682	0.694	0.673	0.652	0.650	0.654	0.627	0.696	0.688
	3-gram	0.760	0.768	0.764	0.682	0.687	0.682	0.747	0.747	0.749
RF	*bow*	0.740	0.750	0.743	0.687	0.707	0.707	0.741	0.756	0.745
	2-gram	0.671	0.714	0.675	0.630	0.656	0.644	0.703	0.730	0.706
	3-gram	0.708	0.720	0.715	0.639	0.661	0.659	0.712	0.724	0.729
IBk	*bow*	0.644	0.622	0.592	0.659	0.670	0.686	0.692	0.664	0.690
	2-gram	0.643	0.622	0.596	0.608	0.648	0.585	0.667	0.684	0.610
	3-gram	0.631	0.610	0.607	0.626	0.614	0.606	0.662	0.646	0.642

We applied configuration with minimum word frequency 1, learning rate 0.025, iterations 5, window size 5, and epochs 5. Notice that we use different values from their typical values for each epoch and iteration to enhance training success. We select the number of epochs lower than 15 as greater values may cause overfitting (Tezgider et al., 2018). We trained two different distributed models for each of the wall datasets depending on the preprocessing level as basic or linguistic. We then used the mean vector of observed words' vectors to represent each document in the related dataset. Figure 5.29, Figure 5.30, and Figure 5.31 present our results which are obtained under different circumstances. Notice that we excluded the NBM classifier as it cannot handle negative values.



	50-B	50-L	100-B	100-L	200-B	200-L	300-B	300-L	500-B	500-L
NB	0.736	0.744	0.722	0.732	0.711	0.716	0.706	0.710	0.699	0.692
SVM	0.842	0.839	0.833	0.823	0.807	0.795	0.785	0.771	0.749	0.725
IBk	0.789	0.778	0.791	0.786	0.796	0.789	0.791	0.793	0.793	0.791
J48	0.771	0.765	0.762	0.756	0.750	0.753	0.740	0.751	0.739	0.743
RF	0.814	0.810	0.801	0.809	0.805	0.801	0.789	0.800	0.787	0.783

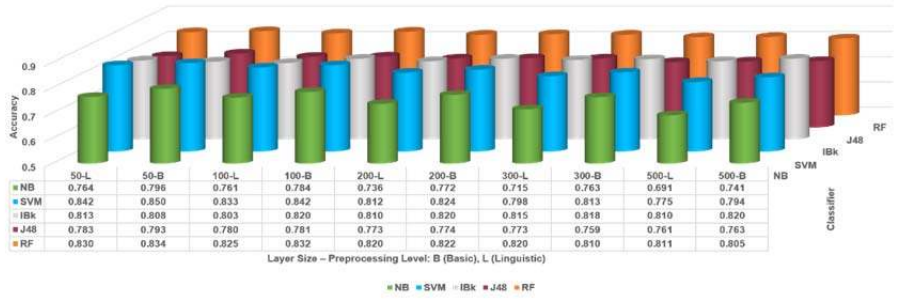
Figure 5.29. Obtained accuracies in L4 model for WD1 dataset with word embedding features



	50-L	50-B	100-L	100-B	200-L	200-B	300-L	300-B	500-L	500-B
NB	0.723	0.737	0.736	0.734	0.696	0.714	0.679	0.700	0.657	0.680
SVM	0.822	0.826	0.797	0.808	0.761	0.774	0.730	0.752	0.688	0.716
IBk	0.774	0.778	0.772	0.775	0.778	0.780	0.781	0.784	0.776	0.782
J48	0.753	0.755	0.746	0.750	0.730	0.730	0.727	0.727	0.724	0.724
RF	0.797	0.800	0.798	0.790	0.793	0.781	0.784	0.784	0.783	0.779

Figure 5.30. Obtained accuracies in L4 model for WD2 dataset with word embedding features

As seen from these figures, the most successful classifier is SVM, while the worst one is NB. For all datasets, the best accuracies are obtained with a layer size of 50, and classification performance is often decreased while the layer size has an increase.



	50-L	50-B	100-L	100-B	200-L	200-B	300-L	300-B	500-L	500-B
NB	0.764	0.796	0.761	0.784	0.736	0.772	0.715	0.763	0.691	0.741
SVM	0.842	0.850	0.833	0.842	0.812	0.824	0.798	0.813	0.775	0.794
IBk	0.813	0.808	0.803	0.820	0.810	0.820	0.815	0.818	0.810	0.820
J48	0.783	0.793	0.780	0.781	0.773	0.774	0.773	0.759	0.761	0.763
RF	0.830	0.834	0.825	0.832	0.820	0.822	0.820	0.810	0.811	0.805

Figure 5.31. Obtained accuracies in L4 model for WD3 dataset with word embedding features

The word2vec model is not sensitive to the preprocessing level as results obtained with basic preprocessing are slightly different from the results of the linguistic preprocessing. The best result is obtained with an accuracy of 0.850 on the WD3 dataset and this shows that the L4 model can not outperform the L2 and L3 models with this second way of experiments as well.

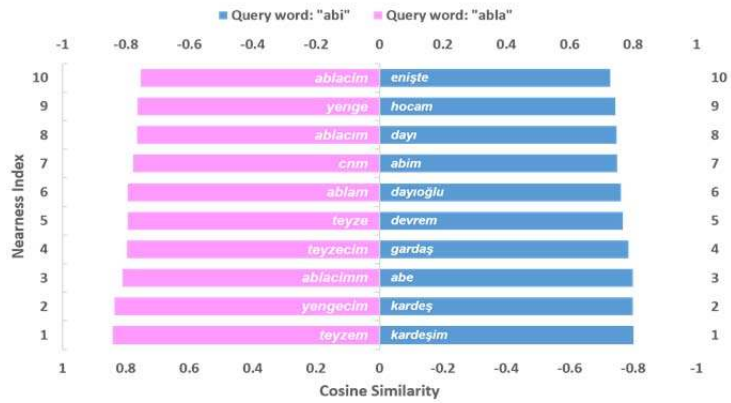


Figure 5.32. Top 10 nearest words for query terms *"abi"* and *"abla"* with respect to the cosine similarity

The word2vec model is, on the other hand, very successful when compared to the classical feature models (i.e., bow and n-gram) which are used in the way of experiments in the L4 model. It achieved slightly different results just by using word vectors with a length of 50, while classical feature models need tens of thousands of features to achieve an approximate result. Notice that the word2vec model outperforms classical feature models on the WD3 dataset and it produces slightly different results on the WD1 and WD2 datasets as well. Figure 5.32 presents the top 10 nearest words for both male-oriented word *"abi"* and female-oriented word *"abla"* with respect to the cosine similarity over the WD3 dataset. As seen in Figure 5.32, the word2vec model is very successful in clustering semantically similar words.

In the last step of our experiments, we employed our models together by combining their features. For this purpose, in each model, we selected the best dataset and feature set that provide the best accuracy in the related model. For instance, in the first way of the experiments in the L4 model, the best result was obtained with tf*idf weighted bow features on the WD1 dataset. As such, we name this configuration as $L4_{bow}$ and the best configuration in the second way of experiments as $L4_{WE}$, that obtained the best result on the WD3 dataset using word embeddings with a layer size of 50. Afterward, we get the best performant experimental setup for each model and combine them to explore whether combining these models improve our results.

Table 5.12. Obtained accuracies for combined models with respect to different classifiers

Combined Model	**Code**	**Classifier**				
		NB	SVM	IBk	J48	RF
L2 + L3	C1	0.805	0.876	0.968	0.979	**0.982**
L2 + $L4_{WE}$	C2	0.936	0.917	0.968	0.979	**0.981**
L2 + $L4_{bow}$	C3	0.650	0.703	0.946	**0.979**	0.894
L3 + $L4_{WE}$	C4	0.800	0.615	0.819	0.958	**0.964**
L3 + $L4_{bow}$	C5	0.671	0.842	0.589	**0.961**	0.875
$L4_{bow}$ + $L4_{WE}$	C6	0.642	0.709	0.675	**0.826**	0.785
C1 + $L4_{WE}$	C7	0.905	0.589	0.968	0.977	**0.981**
C1 + $L4_{bow}$	C8	0.683	0.795	0.943	**0.977**	0.913
C1 + C6	C9	0.685	0.795	0.951	**0.980**	0.899

Table 5.12 presents obtained results which make it clear that only combined model that is coded with C1 improves performance. This model provides an accuracy of 0.982 with help of the RF classifier. The accuracy obtained with the C1 model is the highest result among all experiments that were conducted so far.

In this task, we studied the problem of gender detection of Facebook users from Turkey. For this purpose, we employed ML methods on the L2, L3, and L4 models which use profile information, wall interactions, and wall contents of users

respectively. The profile-based approach provides better results than the content-based approach. The main reason for this is that the content-based approach requires effective feature extraction, while the profile-based approach only uses revealed information of users as features. In the L2 and L3 models, the majority of features are extracted depending on the genders of users who are part of WO's social network. The gender of any user in this network is inferred by voting the number of male and female users who use the same first name in OSN. Our results show that this first name centric strategy is successful and the first name is very effective in the gender detection task.

It is possible to infer gender attribute if the username has been selected to include the user's first name, even if the account is completely private. In addition, users' social interactions (friendship, family membership, posting an activity on any befriended user's wall) are also effective as well as public profile data in gender detection. Despite the challenges of the L3 model, it produces the second-best results. This shows that wall activities and other interacting users in these activities are very effective in determining the gender of a Facebook user. However, in this model, the inference is very difficult and often requires language-dependent operations to detect whether an observed gender-oriented word in any activity is used in a gender sense. We think that this model's success can be improved by making several improvements such as applying morphological analysis and WSD to eliminate words used in a non-gender-oriented sense. However, the use of informal language among Facebook users will harm the performance of these tasks.

The L2 and L3 models have a significant advantage in terms of performance thanks to their low feature space, but the L4 model applied with the bow and n-gram features have a huge number of features which makes gender classification computationally expensive. However, it is possible to overcome this challenge without loss of performance by using the word embeddings in this model. Using word embeddings to represent texts often produces better results than

the classical bow and n-gram features if the word2vec model is trained on a sufficiently large dataset.

The L4 model falls behind that of the L2 and L3 models. The main reason for this is that Facebook activity contents are often dirty and contain many typos. This is the main factor of the fact that a huge number of features are obtained in the bow model even though our wall datasets have low sample sizes. In the L4 model, the best result is achieved on the WD1 dataset. This proves that activities targeting WO are more effective in determining his/her gender.

In Facebook, determining the topic of the wall activities is challenging as posts may include different types of data such as text, photograph, video, and so on. Another challenge is detecting to whom an activity targets among users who posted before under the same parent activity. Developing effective solutions to these problems will make it possible to further increase the performance of the L3 and L4 models.

5.2.3. Privacy Measuring

OSN users share their personal information and other things about different aspects of their lives. Disclosure of sensitive information exposes the users to privacy risks such as identity theft, digital stalking, etc. As such, we perform privacy analysis and privacy scoring tasks over the crawled data. This section presents the results of these tasks in the following sub-headings.

5.2.3.1. Privacy Analysis

In this section, we present our results of the privacy analysis task which we perform on both 5K and 10K snapshots. As stated in Section 2.3.1, the majority of existing works in this field collect privacy preferences of OSN users with the help of online or offline surveys (Coban et al., 2020b). Unlike these studies, our privacy analysis is performed on the crawled data which is a real-world OSN data. Our data

includes data of Turkish Facebook users and this is another difference in our analysis with existing works. There are two previous works (Kılıç and Inan 2019a; Kılıç and Inan 2019b) that have similar aim with this task, but they are different from our analysis in terms of the OSN and item set considered in privacy scoring.

In this task, we use both IPS and NPS methods to obtain the privacy scores of users. These calculations require building the R matrix at first. As such, we selected 21 different items of users and created 5000 x 21 and 10000 x 21 R matrices for 5K and 10K snapshots respectively. Table 5.13 presents a list of our items.

Table 5.13. List of items used in privacy analysis of users

Items		
Phone number	Other OSN account(s)	Wall page publicity
Religious view	Language	Friends page publicity
Interest	Important event(s)	Education status
Political view	Relational status	Email
Work/Job	Family membership	Address
Place (hometown)	Biography	Gender
Place (lived-in)	Quotation	Date of birth

Notice that studies based on surveys or simulations take privacy preferences with respect to the items from a survey audience or a random distribution respectively. However, in this task, we use real-world OSN data and follow some rules to build our R matrices. This is because we collect OSN data by a crawler, which is not able to determine the exact privacy level of a discovered item in the crawling phase. In detail, the crawler sees OSN from the perspective of the seed user that is used to start the BFS search. Let our OSN data is modeled by using a graph and uses notation described in Section 4.1.

As stated before in Section 4.9.1, the R matrix stores the privacy levels/preferences of n users with respect to m items. r_{ij} represents the value of i-th row and j-th column of the R matrix and this value corresponds privacy level or

visibility degree (h) of the user $v_i \in V$ for any item $t_j \in T$ in a range of $[0, l]$. Now, we are ready to give details of our rules used when building the R matrices.

Let user $u = v_i$ represent any friends of the seed user. If the crawler does not see item t_j, then it is certain that $r_{ij} = 0$. However, if the crawler sees/discovers the item t_j, then this just means that $r_{ij} > 0$. Similarly, let user $w = v_i$ is the second-level connection (i.e., a friend of seed user's friend) of the seed user. If the crawler sees an item t_j, then $r_{ij} > 1$, and $r_{ij} \leq 1$ otherwise (Coban et al., 2020b).

Similar uncertainties exist for all levels of the seed user's connections. This can be neglected as the direct connections of the seed account correspond to a very small proportion of users on the network. For friends of its friends, any account that is visited after the 2nd neighborhood is unlikely to be the friend of the seed account. Therefore, it is highly probable that the information that the crawler sees in such an account will have a visibility degree of 3. For these reasons, in this task, the matrix R was constructed according to the following rules:

- If user $u = v_i$ is a friend of the seed user and the crawler sees the value of the item t_j, then $r_{ij} = 1$. Otherwise, $r_{ij} = 0$.
- For all other accounts including the seed user's friends of friends ($w = v_i \in V$) if the crawler sees the value of the item t_j, then $r_{ij} = 3$. Otherwise, $r_{ij} = 0$.

Depending on the preference of the account visited by the crawler, the list of friends can be private (h = 0), public (h = 3), or partially public (i.e., only common friends can be seen, h = 2). For this reason, h = 2 visibility level is also used exceptionally only for the friend list attribute/item.

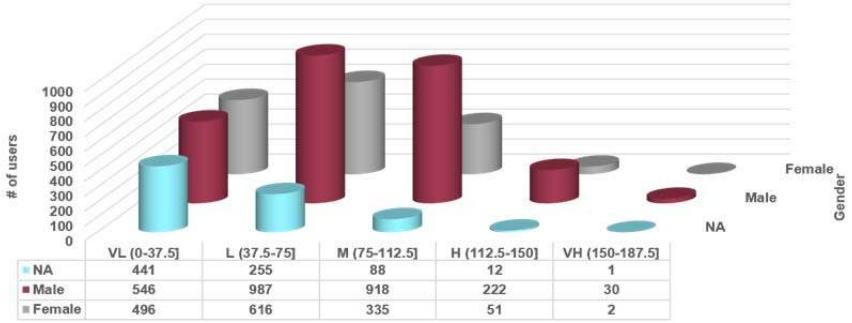
After building our R matrices, we calculate sensitivities and visibilities of items with the help of Equation 4.59, Equation 4.60, and Equation 4.61

respectively. Next, the IPS and NPS values of users are computed by using Equation 4.64 and Equation 4.66 respectively. In this phase, the PageRank algorithm is employed with the help of the NetworkX package with a 0.85 damping factor and 100 iterations. Table 5.14 presents sensitivities for some of the items and NPS and IPS values for randomly selected five users.

Table 5.14. Sensitivity values of a non-exhaustive list of items and privacy scores for randomly selected five users.

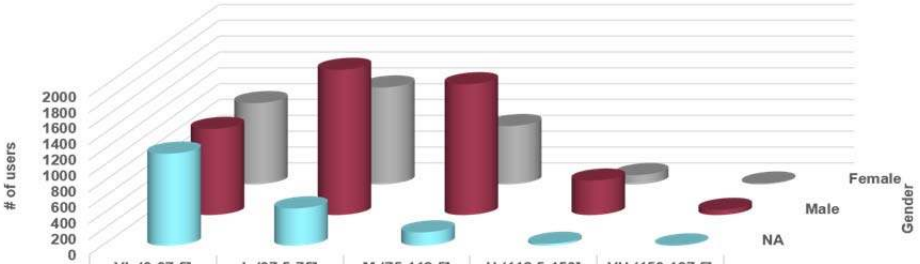
User	**Items**			**Privacy Scores**	
	Address	Gender	Friend List	IPS	NPS
ayse**	0.33	0.33	0.26	6.93	0.004
busra**	0.38	0.38	0.30	8.08	0.011
hatun**	0.11	0.11	0.08	2.31	0.009
aerm**	0.55	0.55	0.44	11.55	0.008
cahi**	0.16	0.16	0.13	3.46	0.012

As seen from Table 5.14, the risk effect of the attributes (except the friend list) to the users is the same, and not all attribute values are given to save space. After computing the IPS and NPS values, we performed analyses to explore the relationships among age, gender, and privacy attitudes of users. For this purpose, we applied equal width binning on the IPS values and created five risk groups including *"VL (Very Low)"*, *"L (Low)"*, *"M (Medium)"*, *"H (High)"*, and *"VH (Very High)"*. Then, we obtained the number of users with respect to both gender and IPS values. We perform this experiment over both 5K and 10K datasets. Figure 5.33 and Figure 5.34 depict our related results of this experiment for snapshots of the 5K and 10K respectively. Notice that IPS values are multiplied by 10 to make the display easier.



	VL (0-37.5]	L (37.5-75]	M (75-112.5]	H (112.5-150]	VH (150-187.5]
NA	441	255	88	12	1
Male	546	987	918	222	30
Female	496	616	335	51	2

Figure 5.33. The number of users in 5K with respect to the IPS level and gender



	VL (0-37.5]	L (37.5-75]	M (75-112.5]	H (112.5-150]	VH (150-187.5]
NA	1162	470	168	24	3
Male	1086	1832	1650	434	64
Female	1025	1222	735	115	9

Figure 5.34. The number of users in 10K with respect to the IPS level and gender

Next, we analyzed users who disclosed the date of birth to examine privacy awareness and the relationship between their ages. In this experiment, for both of the datasets, users who are grouped under the risky group medium, high, or very high with respect to their IPS risk scores are included and the results are given in Table 5.15.

Table 5.15. Age-range distribution for users who are being in the risky group according to the IPS values and sharing the date of birth attribute

Snapshot	Age Range			
	0-20	21-40	41-60	61+
5K	2	121	53	4
10K	6	237	87	6

Finally, we applied equal width binning on the NPS values and obtained five levels as we do for IPS scores in the previous experiment. In our last experiment, we explored the number of users with respect to the NPS level and gender. Figure 5.35 and Figure 5.36 show our results of this experiment for both 5K and 10K snapshots respectively.

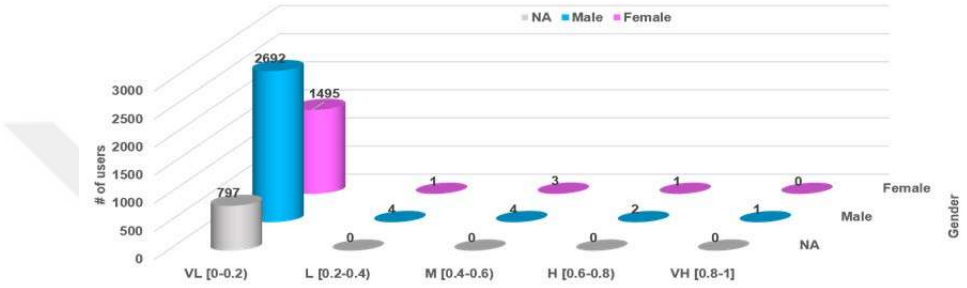


Figure 5.35. The number of users in 5K with respect to the NPS level and gender

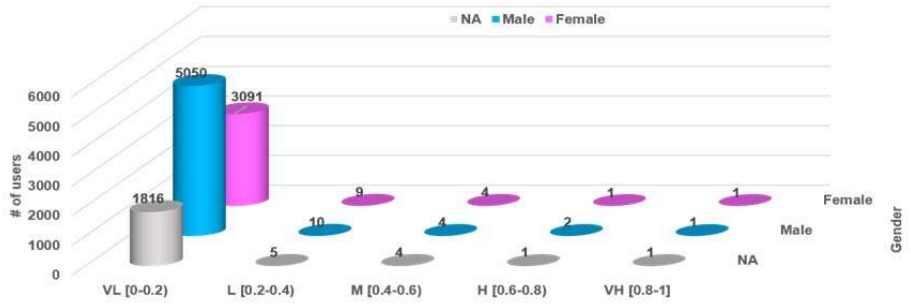


Figure 5.36. The number of users in 10K with respect to the NPS level and gender

Our results make it clear that the number of users with high IPS and NPS scores is low for both of the data sets. When the NPS values are taken into consideration, the risk level of the users is decreased. On the other hand, when analyzed in terms of IPS values, middle-aged (21-40) users were found to be riskier compared to other age groups. Besides, it has been observed that users' NPS values are generally increasing as the size of the network increases (Coban et al., 2020b).

Users in the risky groups are included in this group because they interact with more users and share more information. Among users who share their gender, males are often more likely to have a higher privacy risk than females. The method of obtaining the R matrix has a direct effect on risk measurement. Therefore, when collecting automatic data for privacy risk scoring purposes, a precise measurement must be able to accurately determine the extent of the appearance of the information by the crawler, but this is very difficult to achieve for Facebook.

In this task, it has been assumed that information that the crawler cannot see is hidden, which means the best scenario in terms of privacy. However, the information assumed to have been hidden may have been disclosed to any part of the network even if not the whole, which indicates that the number of risky users is higher than measured.

5.2.3.2. Privacy Scoring

Conventional privacy-related studies mainly focus on protecting or measuring identities or private attributes of users. In particular, most of the existing studies measure the privacy risk of users just based on their disclosed information. However, a lot of user's information including personal aspects can be extracted from OSN data. Therefore, in this task, we implemented a privacy risk evaluation approach such that its contributions to the existing literature can be summarized as follows:

- Unlike most of the existing studies, we perform privacy scoring over real-world Facebook data.
- We use different privacy risk scoring methods and explore correlations between them.
- We perform attribute inference tasks based on contents and network structure information of users to show that users are actually at higher risk than they believe.

- We adopt the self-information feature weighting method to compute the non-symmetric social tie strengths of users.
- Unlike existing studies, we use and show that social tie strengths have to be taken into consideration when computing their centrality-based (or network-aware) risk scores.

We would like to note that this task is an extended study of our preliminary privacy analysis task (see Section 5.2.3.1) and to the best of our knowledge, it is the first try into privacy scoring of Facebook users from Turkey.

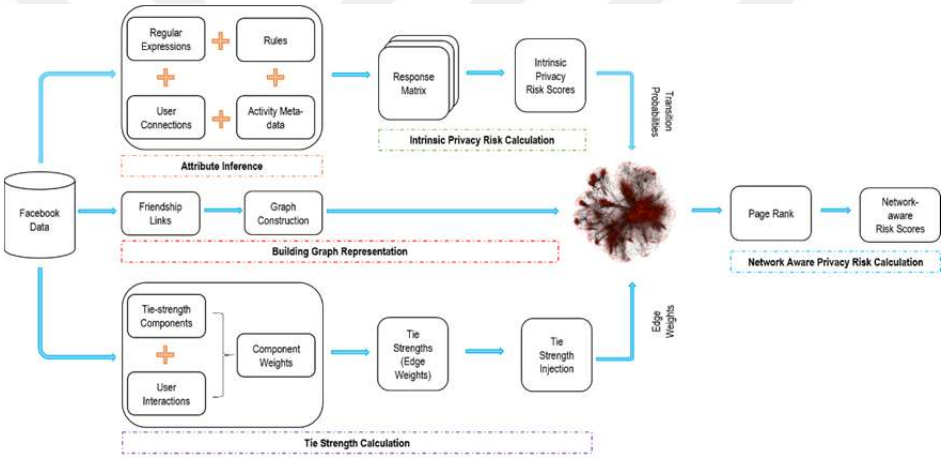


Figure 5.37. Flowchart of our privacy risk scoring.

In this task, we work with the largest sub-graph generated by our crawler. This graph contains basic profile information of 20K users alongside their friendship links (approx. 4M for all discovered accounts and 402,3K for users in this largest sub-graph) and wall activities (over 5.9M) which are comprised of posts, comments, and replies (see Table A.1 and Table A.2). We would like to note that only 256,979 of 402,3K friendship links disclosed by users, but we inferred private friendships using bidirectional structure (see Figure 1.3) of Facebook.

Figure 5.37 depicts a flowchart of our privacy scoring framework that includes the following steps:

- Attribute Inference: This step uses regular expressions, rules, user connections, and activity meta-data information to infer the private attributes of users.
- Building graph representation: Converts relational Facebook data into a graph data structure by using friendship links of users.
- Tie strength calculation: Calculates tie strength between two befriended users based on our tie strength components (see Table 4.14). Computed tie strengths are injected into the graph as being edge weights.
- Intrinsic privacy risk calculation: Computes intrinsic privacy risk scores based on sensitivity and visibility components which are obtained from the R (i.e., response) matrix. Computed risk values are then injected into the graph as being transition probabilities of nodes.
- Network aware privacy risk calculation: Performs privacy risk measuring by running personalized page rank algorithm over an injected graph.

To represent OSN data, we again use our graph notation where $G(V, E)$ represents our digraph, V is a set of nodes, and E is a set of directed edges. Each user in V may disclose information related to a set of topics T (see Section 4.1 for more detail about our graph notation).

In this task, we consider 16 different attributes in measuring the privacy risk of users. We would like to note that information contained within OSN data is difficult to correctly interpret through just using rules, regular expressions, and so on. This is because people are prone to use sarcastic phrases or provoking words that cannot be easily interpreted in their textual contents. Additionally, OSN data is often incomplete. As such, we do not include some attributes (e.g., religious view)

in our current analysis, and leave them untouched for future improvements. Table 5.16 presents a list of considered attributes such that we try to infer their values based on contents and social connections of users.

Table 5.16. List of considered attributes in our privacy scoring framework.

Attribute/Item	Code	Attribute/Item	Code
Friend List	AL	Political View	SG
Email	MA	School	OKL
Birth Date	DG	Place (Hometown)	HT
Age	Y	Place (Lived-in)	YY
Phone Number	TN	Places (Other)	DY
Gender	C	Family Membership	AKR
OSN Account(s)	CSA	Relationship Status	ILD
Marriage Date	ET	Having Child Status	CCK

To obtain privacy risks of users, we first perform attribute inference and created response matrices using different models which we call L1, L2, and L3. Next, build our response matrices and obtained intrinsic privacy risk scores of users. Afterward, we computed social tie strengths between users and injected our sub-graph to represent edge weights. Finally, we utilized the page rank algorithm on this weighted and directed graph to obtain network-aware privacy risk scores of users. We would like to note that we use IPS and Pr_IRT methods to compute intrinsic privacy risks of users. To reduce complexity, we call the Pr_IRT method as IRT in all experimental evaluations and results related to this task. To compute network-aware privacy risk scores, on the other hand, we use the NPS method as in our preliminary analysis. The following subsections present the results of each sub-task in our flowchart.

(1) Results of attribute inference: In this phase, we divide attribute values into three groups which include revealed or inferred values. These groups are as follows:

(A) L1: Revealed attributes: This model only considers revealed attribute values while populating the response matrix.

(B) L2: Inferred attributes from contents: Uses meta-data and contents of all activities of a user in his/her wall along with his/her profile items to infer private attribute values. This model utilizes regular expressions, some rules, and meta-data in the inference phase. We group attributes into six groups based on used inference ways which are as follows:

- G1: This group includes MA, TN, and CAS attributes such that their values are inferred by using regular expressions. For instance, we search URLs in attribute fields and the activity contents of a user. Next, we filter URLs to select ones which include OSN names (e.g., Twitter, Instagram, etc.) and extract the username from these URLs. Similarly, we search email addresses and phone numbers within the activity contents and attribute fields of users.
- G2: This group includes ILD and CCK attributes such that their values are inferred by using some basic rules. To infer these attributes, we first check related fields in both WO's and his/her friends' profiles. If there is not any information, we use our simple rules to infer attribute value based on wall activities. For instance, to infer the ILD attribute of WO, we first seek for relationship fields of his/her friends' profiles to check whether they have disclosed a private relationship with WO. If we find such a user, we mark WO's ILD attribute as inferred. Otherwise, we apply two simple rules to infer the ILD attribute. As a first step, we take activities written by WO and search some multi-word phrases (e.g., *"düğünüm var", "artık evliyim", "evlendik", "düğünüme bekliyorum", "evlilik yıldönümü", "eşimle beraber",* etc.) which strongly imply that WO's relationship status changed. If we could not find anything in the first step, we then consider activities written by friends of WO. We again search some multi-word phrases (e.g., *"mutluluklar", "ömür boyu mutlu", "bir yastıkta", "güzel gelinim", "eşinle birlikte", "mutlu mesud",* and so on.) in such activities to

infer ILD attribute. We would like to note that multi-word phrases are created based on our observations. We apply the same steps also for the CCK attribute with different multi-word phrases such as *"analı babalı", "bebeğin çok tatlı", "yanaklarını yerim", "allah bağışlasın",* and so on.

- G3: This group contains AL, DG, Y, ET, and OKL attributes such that their values are inferred mainly from wall activities, interacted users under wall activities, and meta-data of activities. For instance, to infer the friend list of WO, we search users who interact (i.e., posting a status, commenting on WO's post, etc.) with WO on both their own wall pages and WO's wall. To infer DG, we search direct and event posts (see Figure 1.6) including some multi-words phrases (e.g., *"doğum günü", "nice yaşlara", "tarihinde doğdu",* etc.) and take the most frequently observed posting date as DG (i.e., date of birth) of WO. To infer the Y attribute, we only consider event posts that show the exact date of birth information. To infer the OKL attribute, we first search event posts that give information about educational changes of users (e.g., *"okula başladı", "okulu terk etti", "mezun oldu"* etc.). If we could not find such posts, we use our lexicon of schools and universities (see Section 3.5.1) and try to match any of the school names within any part of activity contents. On the other hand, we infer the ET attribute by taking the posting date of posts such that they include multi-words phrases such as *"düğünüm var", "evlendim",* and so on.
- G4: This group contains HT, YY, DY attributes which are inferred by iterating over all textual contents of users and employing a lexicon-based searching. To infer these attributes, we use a common lexicon that includes district and province names in Turkey. However, we apply simple rules to differentiate the values of these attributes. For instance, if we find a place name in the activity contents of WO, we also search some other words (e.g., *"baba ocağı", "memleketim", "doğduğum yer", "bizim köy",* etc.)

in preceding and following words, if exist, to add this place as one of the possible candidates of HT attribute. Similarly, we try to infer YY and DY attributes. We do not apply any additional search for DY attributes, whereas we search for some other words (e.g., *"ilk iş günü", "iş yerim", "okulum", "taşındım", "yeni evim",* etc.) as we do for the HT attribute. We would like to note that we also consider posting location information, if exist, to infer possible candidates of users' places.

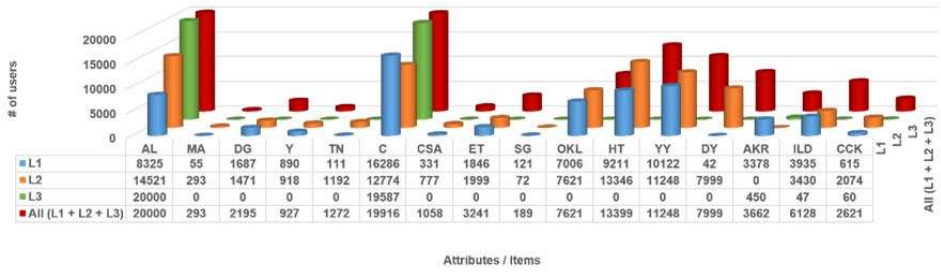
- G5: This group is only including the SG attribute which is inferred just by using users' liked pages of users. To achieve this task, we used our dictionary of political pages (see Section 3.5.1) and inferred which party WO supports. At this stage, we reached a decision by looking at which of the six parties (i.e., JDP, NMP, RPP, HP, DPP, and GP) was observed more frequently among the political pages that WO liked.
- G6: This group only includes the C attribute which is inferred by building the best model in our gender detection task which is presented in Section 5.2.2.2.

(C) L3: Inferred attributes from social connections: This model performs attribute inference only by using social connections of users and mainly depends on the bidirectional nature of Facebook. The attributes which are taken into account by this model are as follows:

- AL: We extract the friend list of users by performing reverse social engineering.
- C: We infer user gender by using the popularity of his/her first name among other users in the crawled snapshot.
- AKR, ILD, and CCK: We infer these types of family memberships based on the knowledge that edges are bidirectional (see Figure 1.3) on

Facebook. For private relationship status (i.e., ILD), we only search for any user who discloses that he/she is in relation with WO. For family membership attributes, on the other hand, we perform inference by using WO's all family memberships which are disclosed by both WO and his/her friends. For instance, let A, B, and C be three Facebook users and A discloses that he is the son of B, while C reveals that she is the wife of B. Using these connections, we can infer that A is also the son of C, even C does not reveal this information. We perform this inference just one time to avoid making a biased inference, as users can also share unrelated individuals as their family members. Note that we also used this simple inference attack in our kinship task to enlarge our datasets (see 5.2.2.1).

The number of revealed (for L1) and inferred (for L2 and L3) attributes with respect to the users are depicted in Figure 5.38. As seen from Figure 5.38, when we consider overall inferred attributes, our results show that the most inferred ones are AL, C, HT, and YY.



	AL	MA	DG	Y	TN	C	CSA	ET	SG	OKL	HT	YY	DY	AKR	ILD	CCK
L1	8325	55	1687	890	111	16286	331	1846	121	7006	9211	10122	42	3378	3935	615
L2	14521	293	1471	918	1192	12774	777	1999	72	7621	13346	11248	7999	0	3430	2074
L3	20000	0	0	0	0	19587	0	0	0	0	0	0	0	450	47	60
All (L1 + L2 + L3)	20000	293	2195	927	1272	19916	1058	3241	189	7621	13399	11248	7999	3662	6128	2621

Figure 5.38. The number of revealed (L1) and inferred (L2 and L3) attributes with respect to attributes of users.

Using the L2 model, we can perform inference tasks for the majority of items. The attributes whose values are most inferred are AL, C, OKL, HT, YY, and DY, whereas the least inferred ones are MA and SG. In this model, AKR and SG attributes are difficult (please see results of our kinship detection task related to the

inference of the AKR attribute) to correctly interpret through automatic tools. As such, we do not include these items in our content-based inference task in the L2 model. Notice that the SG attribute is inferred just based on liked pages of users.

The L3 model, on the other hand, can be employed to infer AL, C, SG, OKL, HT, YY, DY, AKR, ILD, and CCK attributes. However, we employed the L3 model only for a few of these attributes due to users' information in our crawled snapshot is incomplete. The results in Figure 5.38 shows that it is possible to infer a complete list of a user's friend list (i.e., AL), family memberships (i.e., AKR and CCK), and private relationship (i.e., ILD). The popularity of an attribute within the overall network can help to infer some attributes like C, even the first neighborhood (i.e., friends) of WO is private or incomplete.

After completing attribute inference, we measured how our L2 and L3 models are accurate. In order to do so, we used a simple approach such that if there is an intersection between revealed and inferred value(s), we assume that the related model's inference is true and false otherwise. We would like to note that most of the attribute inference attacks in the L2 model produce a list including possible candidate values for the related attribute. This is because we use regular expressions, rules, lexicons to infer the majority of the attributes in the L2 model. Using this approach, we obtained accuracies and show related results in Figure 5.39, where 0 means that we do not include the related attribute in the inference task for the corresponding model.

As seen from Figure 5.39, the L2 model often achieves considerable accuracy even it infers attribute values from textual contents and liked pages of users. In the L2 model, attributes (e.g., HT, YY, DY) which are unpredictable by using a regular format or some specific rules, inferred with low accuracies, while others inferred by regular expressions and specific rules obtained with high accuracies.

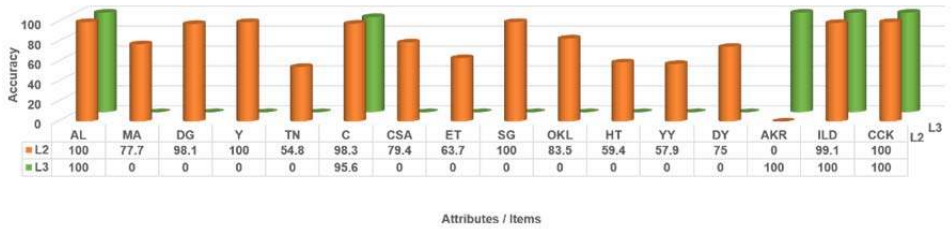


Figure 5.39. Accuracies of our inference models with respect to the user attributes.

The attributes which are inferred with the highest accuracies by the L2 model are AL, SG, ILD, CCK, C, and DG, while the lowest accuracies are obtained for TN, YY, and HT attributes respectively.

In contrast, the L3 model achieves very high accuracies for all of the attributes it employed for. This is because the L3 model uses social connections or information of these connections which are not mined or extracted information. This information may be an edge that is known to be present (due to the bidirectional nature of Facebook) or an attribute revealed by any user in the network.

(2) Results of tie strength calculation: We computed social tie strengths between each pair of befriended users to inject them as edge weights in the next step. In order to do so, we first computed the self-information of tie strength components as their edge weights using Equation 4.69. We obtained our weights based on the satisfaction of our components (see Table 4.14) by each directed edge of 402.3K edges. Table 5.17 presents the results of this experiment which show that the top five most important components of social friending are SGEN, CFRI, SLI, SHT, and CPAGE respectively.

Secondly, we computed tie strength between each pair of users by using Equation 4.70 which is based on the weight and the observed value of each component in interactions of these users. We then injected our computed tie strengths into the graph representation of our crawled network to represent edge weights.

Table 5.17. Computed weights of the tie strength components in ascending order.

Component	Weight	Component	Weight
PREL	0.000577	CMDIR	0.027054
EVNT	0.001148	RPDIR	0.038379
SJOB	0.002426	SEDU	0.044330
FMEM	0.002461	CMIND	0.122535
RPIND	0.002575	CPAGE	0.194213
DRPST	0.007238	SHT	0.211385
PSTAG	0.014422	SLI	0.235391
PATAG	0.014914	CFRI	0.292170
SCOMP	0.018558	SGEN	0.769739

Note that our tie strength computation produces non-symmetric weights for each of the edges between two users due to the unidirectional components given previously in Table 4.14.

(3) Results of privacy scoring: We compute both intrinsic and network-aware risk scores of users. To compute intrinsic risk scores, we created three different R matrices in the dichotomous case by populating revealed or inferred values of attributes. The main reason behind why we use dichotomous case is that our crawler could not know the exact visibility degree of a revealed attribute (see our preliminary task in Section 5.2.3.1 for further details).

We aim to explore how the privacy risks of users were affected by our attribute inference models. For this purpose, we used an incremental way and created three R matrices, namely, R1, R2, and R3 by populating them with attribute values obtained by L1, L1 + L2, and L1 + L2 + L3 respectively. Notice that we do not use the revealed or inferred value of an attribute while building our R matrices. Instead, in dichotomous case, we use 1 for attributes that their values are revealed by users or inferred by our models, whereas we use 0 to represent attributes whose values are not disclosed or inferred (see Section 4.9.1 for further details). Another important thing here is that we ignored attributes that are predicted with less accuracy than ninety percent. We do such filtering to avoid making biased privacy risk scoring.

(A) Results of intrinsic privacy risk scoring: In this phase, we employ IRT (i.e., Pr_IRT) and IPS privacy risk scoring methods which are previously described in Section 4.9.1. Our response matrices (i.e., R1, R2, and R3) store information about how users make their items visible or think that sensitive. However, each of these matrices has different nature due to we populate them by including inferred items (actually kept private in real) in each step. This leads an item to has a different sensitivity and visibility values for each of our R matrices.

For instance, let assume that an arbitrary attribute A1 is kept private by the majority of users and we inferred its value by the L2 model for the most of users. In this case, the sensitivity of A1 would be high in R1 filled by the L1 model, but its sensitivity would be low in R2 filled by L1 + L2 model. This is because item A1 would be shared in R2 by the majority of users in the sub-graph. This gives way to compute biased privacy risks with respect to each of the three models. Therefore, in this phase, we obtained sensitivity and visibility of items based on the R1 and used these values to compute privacy risks from R2 and R3 as well. Figure 5.40a and Figure 5.40b depict privacy risks of randomly selected 100 users due to their shared/inferred items with respect to the R1, R2, and R3 matrices filled by L1, L1 + L2, and L1 + L2 + L3 models respectively.

We would like to note that, in IRT calculation, we scaled the α parameter to an interval (0, 2), while β and θ parameters to the interval [6, 14] as done by Liu and Terzi in their experimental scenario (2010). As seen from these figures, IRT and IPS methods have similar behavior and produce scores at different scales. Privacy risks of users increase after making attribute inference. The highest privacy risk scores are obtained on the R3 matrix which is filled by all of our models.

(B) Result of network-aware privacy risk scoring: In this phase, we obtained NPS values of users by using the page rank algorithm which is formulated in Equation 4.65. However, differently from Pensa et al., (2019), we also injected social tie strengths (TS in short) of users as edge weights which is one of the major contributions of this task. In this phase, we obtained NPS values in different ways

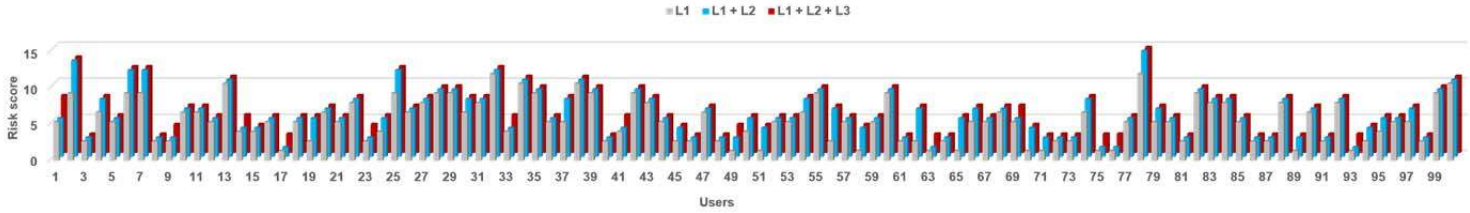
where we fed IRT and IPS scores as initial values of the page rank algorithm. As such, we use NPS_{IPS} and NPS_{IRT} to represent NPS scores obtained by the IRT and IPS scores as initial values respectively.

Additionally, we also included the *"TS"* term as an indicator of whether NPS scores are obtained by injecting social tie strengths or not. To exemplify, $NPS_{IPS\text{-}TS}$ means that NPS scores are obtained by running the page rank algorithm over graph representation of our data that is injected with social tie strengths and uses IPS scores as its initial values.

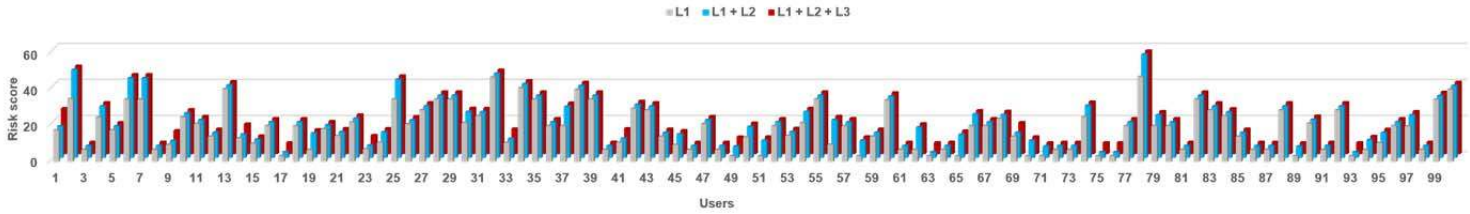
We obtained users' NPS values depending on the initial risk scoring method (i.e., IRT or IPS), R matrix, and injection status of the page rank algorithm. Figure 5.41a and Figure 5.42a show NPS_{IPS} and NPS_{IRT} scores of previously selected 100 users with respect to the attribute models. Figure 5.41b and Figure 5.42b, on the other hand, show results of the same experiments on injected page rank with social tie strengths.

As seen from these figures, intrinsic and network-aware privacy risks are significantly different from each other. Using IRT and IPS values to obtain NPS values only change the scale of risk score as the behavior of the page rank is the same for each case. NPS scores make it enable to observe which users are more central and build more connections in the network. For instance, user numbered with 94 is at more central and build more social connections among previously selected 100 users in the sub-network. Tie strengths are also having a significant effect on the NPS scores of users. As seen from Figure 5.41b and Figure 5.42b, NPS values of user with number 55 increased after injecting the page rank by tie strengths between his/her connections.

After computing privacy risk scores of users, we explored the correlation between risk scoring methods and applied chi-square testing in the next sub-sections to give a clearer insight about behavior and the effect of our suggestion of using tie strength in NPS calculation.

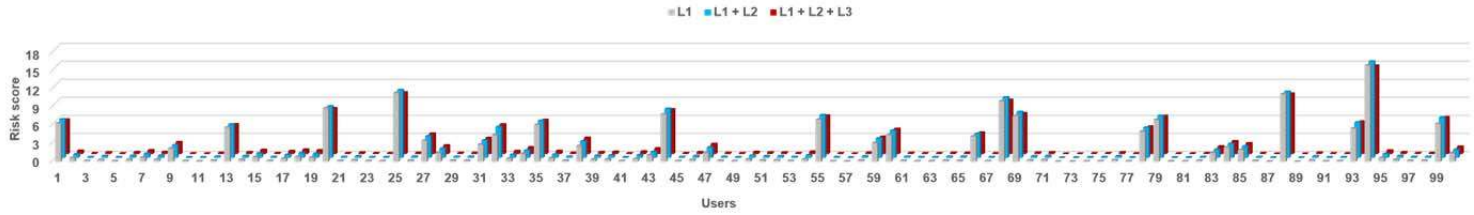


(a)

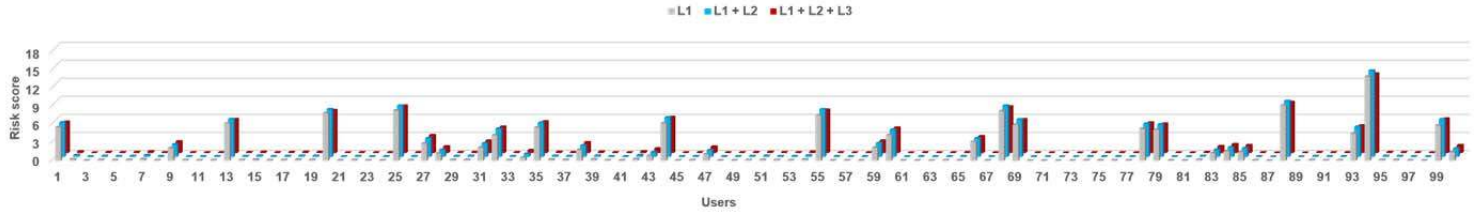


(b)

Figure 5.40. Privacy risk scores of randomly selected 100 users with respect to different models. Risk scores are obtained with: (a) IPS, (b) IRT.

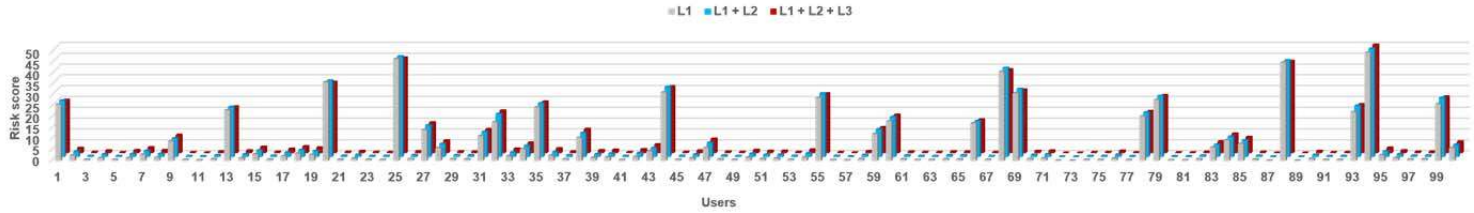


(a)

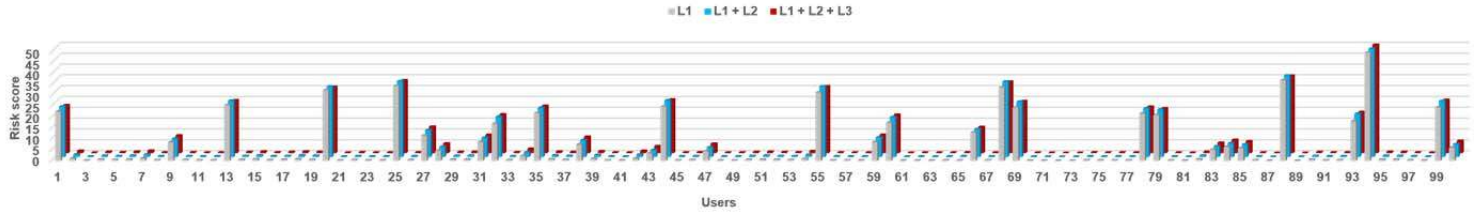


(b)

Figure 5.41. Privacy risk scores of randomly selected 100 users with respect to different models. Risk scores are obtained with: (a) NPS_{IPS}, (b) $NPS_{IPS\text{-}TS}$.



(a)



(b)

Figure 5.42. Privacy risk scores of randomly selected 100 users with respect to different models. Risk scores are obtained with: (a) NPS_{IRT}, and finally, (b) $NPS_{IRT\text{-}TS}$.

(C) Correlation between risk scoring methods: In this step, we computed Pearson correlation (Anonymous, 2020b) between each pair of risk scores obtained by using intrinsic and network-aware risk scoring methods in different cases as mentioned in the previous sub-section. We obtained risk scores on the response matrix R1 and computed pairwise correlations between these methods.

Table 5.18. Pearson correlation scores for privacy scoring methods on the response matrix R1.

Method	**IPS**	**NPS_{IPS}**	**NPS_{IPS-TS}**	**IRT**	**NPS_{IRT}**	**NPS_{IRT-TS}**
IPS	1	0.21	0.24	0.98	0.22	0.25
NPS_{IPS}	0.27	1	0.79	0.20	0.99	0.79
NPS_{IPS-TS}	0.24	0.79	1	0.23	0.80	0.99
IRT	0.98	0.20	0.23	1	0.21	0.23
NPS_{IRT}	0.22	0.99	0.80	0.21	1	0.79
NPS_{IRT-TS}	0.25	0.79	0.99	0.23	0.79	1

As seen in Table 5.18, NPS values have a very low correlation with IPS and IRT values. This means that NPS values have different distribution due to network structure and intense friendship connections have a strong effect on the privacy risk scores of users. IPS and IRT values, on the other hand, have similar behavior as stated in previous subsections. Using tie strengths affects the correlation of NPS values which means that social tie strengths should not be ignored when evaluating network-aware risk scores of OSN users.

(D) Importance of tie strengths in NPS calculation: As seen from Table 5.18, network structure has a strong effect on the privacy risk scores of users. The privacy score of a user depends on his/her location in the network and the number of his/her connections with other users. However, it also depends on the weights of each connection of the users as well. As seen from Figure 5.41b and Figure 5.42b, the privacy scores of users increase or decrease depending on the weights (i.e., tie strength) of their connections. To explore whether there is a strong dependence between NPS score and social tie strength, we use chi-square test statistics (Anonymous, 2020c).

Table 5.19. The number of users with respect to quartiles of average tie strength and NPS values.

Grade (Risk/TS)	VLTS	LTS	MTS	HTS	VHTS
			R1		
VLR	2,114	1,451	386	49	0
LR	1,176	1,392	1,143	281	8
MR	605	945	1,616	281	65
HR	80	232	828	769	912
VHR	0	5	27	953	3,015
			R2		
VLR	2,075	1,452	395	75	2
LR	1,162	1,363	1,150	313	13
MR	648	940	1,588	748	76
HR	90	266	834	1,899	911
VHR	0	4	33	965	2,998
			R3		
VLR	2,045	1,454	414	84	3
LR	1,225	1,325	1,125	313	12
MR	610	972	1,602	742	74
HR	95	270	825	1,891	919
VHR	0	4	34	970	2,992

Our aim here is to explore whether users with high risk have a high social tie strength with others. For this purpose, we obtained the average tie strength value for each user by taking weights of edges coming from other users to the current user at hand. Next, we grouped users by considering the quartiles of NPS and average tie strength values into five different groups. These groups are VLTS, LTS, MTS, HTS, and VHTS for tie strength values, while those are VLR, LR, MR, HR, and VHR for NPS values. Notice that these group names represent tie strength (TS) and risk score (R) in different grades including, low or high (L or H), medium (M), and very low or very high (VL or VH). Afterward, we obtained the number of users for each of these groups and created a contingency table for each model as given in Table 5.19.

We would like to note that Table 5.19 shows three different contingency tables with respect to users' social tie strengths and privacy risk scores obtained from the R1, R2, and R3 matrices respectively. We used *scipy.stats.chisquare*

package to employ chi-square test statistics and obtained a significant p value such that $p < 0.05$ with 95% confidence level for each of three contingency table. These findings show that NPS and social tie strength values are strongly dependent and there is an association between these two variables. In other words, we reject the null hypothesis that these two variables independent from each other.

(4) Overall risk analysis of Turkish Facebook users: In this section, we conduct privacy risk analysis for all users in our crawled snapshot for the purpose of investigating the privacy awareness of Turkish Facebook users and its change by gender and age. Privacy risks due to the revealed and inferred items and social connections have very low correlation (see Table 5.18) and they are at different scales (see Figure 5.40, Figure 5.41, and Figure 5.42). This is because the IPS and IRT methods perform scoring just based on revealed (or inferred) attributes of users, whereas the NPS method considers network structure.

As a result, these two approaches produce different privacy scores for each user. As such, in this analysis, we take the average of privacy risk scores obtained with these two approaches. Even IPS and IRT methods produce highly correlated results, we selected IRT as our intrinsic scoring method in this phase. This is because the IRT approximates the item characteristics curve that best represents the response matrix (Liu and Terzi, 2010). To obtain NPS scores, we fed IRT scores as initial values of the page rank both with the tie strengths ($NPS_{IRT\text{-}TS}$) and without tie strengths (NPS_{IRT}). We aim to make the effect of tie strengths clearer on privacy risk scores. To make a reasonable analysis, we first scaled averaged risk scores in a range of 0 and 1. We then grouped users into ten different groups based on their risk scores. We would like to note that for this analysis we obtained IRT scores on the response matrix R3.

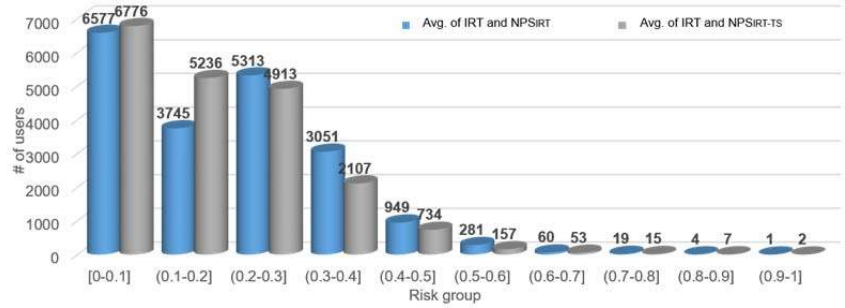


Figure 5.43. The number of users in ten different groups with respect to the average of IRT and NPS values.

We present the results of this experiment in Figure 5.43 which shows that majority of users have less than 0.5 risks of privacy. Using social tie strengths has a strong effect on the privacy risk scores of users. To exemplify, the number of users having risk in a range from 0.1 to 0.2 is 3,745 without tie strengths, whereas it is 5,236 when tie strengths are considered in NPS calculation.

Next, we explored the number of users with respect to their gender attributes and average privacy risk scores. Note that in the rest of our experiments, we use the average of IRT and $NPS_{IRT\text{-}TS}$ values.

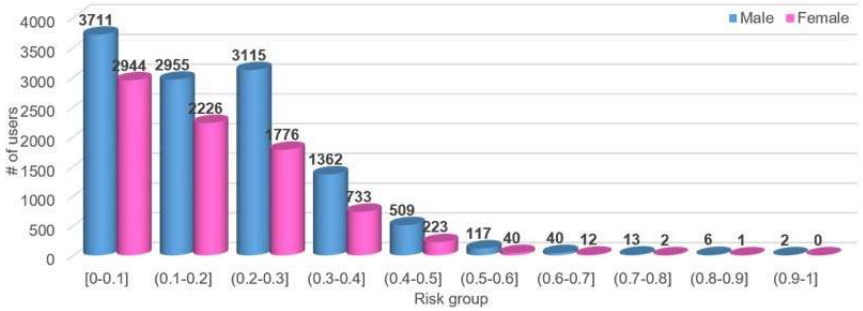


Figure 5.44. The number of users in ten groups with respect to the average privacy risk and gender attribute.

We show the results of this experiment in Figure 5.44, which depicts that male users often have a higher risk of privacy than females in all risk groups. Note

that in this experiment, we considered users who reveal their gender attributes and other users whose gender attributes are inferred by the L2 and/or L3 model(s).

Finally, we investigated the number of users with respect to their age and average privacy risk attributes. In this experiment, we again considered users whose age attributes are obtained from the revealed birth date (i.e., DG) and other users whose age attributes are inferred by the L2 model.

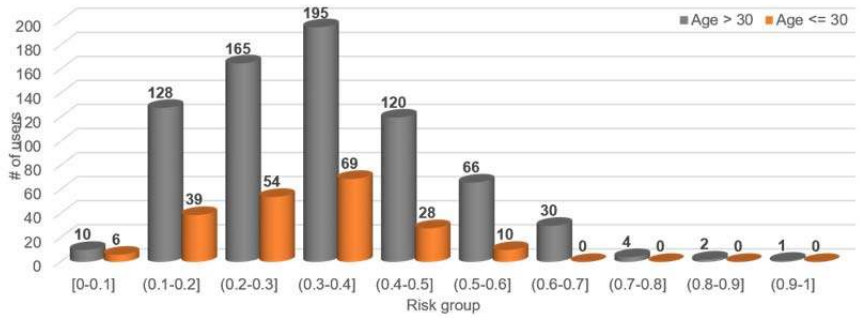


Figure 5.45. The number of users in ten groups with respect to the average privacy risk score and age attribute.

As seen from Figure 5.45, users over the age of 30 years often have a higher risk than those under thirty years of age. We would like to note that we do not name our risk groups in the experiments due to group names are relative and can vary from person to person. However, our privacy analysis of Turkish Facebook users shows that majority of them have less than 0.5 risks of privacy. On the other hand, male users have a higher privacy risk than females, and users over the age of 30 years are at higher risk in general.

In this extended and final task, we perform a comprehensive privacy risk analysis of Turkish Facebook users. For this purpose, we use real-world OSN data unlike most of the existing studies. Besides, we perform attribute inference to fill our R matrices in dichotomous case.

Our attribute inference results show that many attributes of OSN users can be learned using their textual contents and social connections. The majority of

Turkish Facebook users keep their attributes private, but they are still not aware that an attribute can be learned or inferred by using different techniques. Therefore, they often disclose their sensitive information by posting activities on their wall pages or building connections with other users.

However, the risk is so serious that a user may contribute to putting any other user at risk even he/she is not befriended by him/her by only disclosing his gender. This is because an adversary can infer a user's gender by using the popularity of his/her first name among other male and female users in the network. Users disclose their private attributes in their activity contents or other text-based fields of their profiles. Even we used a restricted list of attributes in our inference tasks, the textual contents of users have a great potential for inferring various other attributes (e.g., health status, hobbies, etc.) of users.

Our content-based inference model has provided low accuracies for some of our attributes. However, taking the accuracy metric as a single measure of how an inference task may not be sufficient. This is because the inferred value of an attribute may be true even it does not match with the disclosed value. For instance, a user may have many phone numbers and only disclose one of them in a related field of his/her profile. However, this user can post an activity including his/her other phone number(s) at the same time. In such a case, using accuracy can mislead us because the inferred value may actually be true. Inference based on user connections provide very high accuracies and provides very accurate results. It is also easy to perform such an inference task that requires complete OSN data.

OSN users have a privacy risk score due to their revealed items and social connections in the network. In literature, centrality or network-aware risk scoring methods ignore social tie strengths between users. In this task, we inject and use social tie strengths in network-aware risk computation. We also show that there is a strong association between high-risk score and social tie strength. Therefore, we suggest not just use social connections of users in network-aware risk scoring, but also use weights (i.e., tie strengths) of the social connections of users.

6. CONCLUSION

OSNs are extremely popular services that allow users to interact with each other and share content. Due to the large amounts of data shared by users, OSNs are also rich data sources for research in social network analysis. The first step into OSN research is to access OSN data. Therefore, in this thesis, we first designed and developed a crawler to collect data from Facebook which is one of the most popular OSNs in the world.

We then performed sharing analysis over collected data as analyzing social networks may produce interesting findings and useful knowledge. However, researchers face different challenges to access the data even there are APIs provided by OSN service providers. Therefore, we designed and implemented a crawler that uses HTTP requests and responses to reveal public data and investigated the content-sharing behaviors and awareness of the personal privacy of users in Turkey. Based on our analysis results, we concluded that users generally do not tend to disclose their sensitive attributes either consciously or unconsciously. However, sharing rates of such attributes (e.g., birth date, family members) are too high to be underestimated as by analyzing the OSN data, a lot can be learned about a user.

Users share the books they read, movies they like, places they have been, their comments on other users, their education, and their job status over various OSNs. An adversary that gains to these public data due to insensible use of OSNs may infer high probability private information about a target OSN user who would prefer to keep private. Such attacks may reveal the user's religious belief, race, political opinion, residence address among many others. For instance, if friends of a user post on his/her timeline to celebrate his/her birthday, one can learn this sensitive attribute about the user by considering the dates of the related post. If the attacker obtains personal identifiers such as the mother's maiden name or exact date of birth of an individual, he/she may use these data to impersonate the target

individual over long-distance service channels (e.g., internet or phone). This threat also brings together security risks. Primary causes of concern are identity theft, fraud, and (cyber) stalking.

Motivated by this, we also performed private attribute disclosure for several attributes of users including gender, kinship, and other OSN account(s). To reveal another account of a Facebook user, we performed a user matching task that aims to match users across different OSNs based on solely usernames due to the OSN data is often incomplete. Based on our experimental results, we conclude that the majority of Facebook users do not reveal their other OSN accounts in their Facebook profiles. Users tend to use the same username or a similar one (e.g., ocoban → ondercbn.25) for their connections. Usernames are fundamental and inevitable elements of OSNs and other web sites and often may provide information redundancies. Matching users just based on their usernames may fail in many cases due to users may choose different usernames for their different accounts. On the other hand, using other information of users may not be efficient as OSN data is often incomplete. This task is not an easy task even it is performed on a complete data of OSNs as users may fill their profiles with different (even fake) details.

In the second step of our attribute inference tasks, we performed gender detection of Facebook users to explore which type of public user information produces better results in the gender prediction task. Profile attributes are more preferable than other hand-crafted and learned features in gender detection. However, it may not possible to employ profile attributes to infer the gender of users due to OSN data is often incomplete. In such a case, anyone who is willing to infer the gender attribute of a target user can use the wall owner's wall interactions and wall contents. In light of our extensive results, we conclude that to ensure privacy, wall and other profile pages must be kept private and even selected usernames must consist entirely of numeric values or must not include any part of the user's real name especially the first name. However, this alone is not enough as

any user in your social network may violate your privacy, even if you want to keep related information private. For instance, if any user reveals that he/she is in a relationship with you or you are his/her family member, then anyone can have insight about your basic information including gender. To extend this possibility, if one of the friends types an activity that includes the gender-oriented word(s) and targets you on whether your wall or his/her wall, then he/she reveals your gender.

Finally, we performed kinship detection based on the wall contents of users and used kinship datasets created for this thesis. Based on results and observations in this task, we conclude that users often use many kinship words in their posts and it is quite hard to detect the target user, mentioned user, the topic of the related post, and sense of any kinship word observed in a post. There also other challenges to be handled which stem from the sharing behavior of users. For instance, a user may change its surname on Facebook after when she is married and this will affect any model that considers the surname of users as in this task. Our kinship detection results need to be improved, but we strongly conclude that it is possible to learn/infer kinship relations of Facebook users based on their wall contents. This type of task can easily be achieved by a human. Therefore, we recommend that any user should keep his/her wall page private or avoid using kinship-oriented words in his/her activities when his/her wall page is public.

One of the most interesting things about OSN is that users often communicate with each other via text messages. Motivated by this, we also performed NER and SA task over our collected Facebook data. We first aimed to discover coarse-grained NEs on Facebook posts written in the Turkish language. In order to do so, we developed our NER system that depends on the CRF algorithm and has different capabilities such as automating MUC-CoNLL and tagging (e.g., raw to IOB2) conversions. Based on our results and the challenges we faced; we conclude that NER on Facebook posts falls behind the results that are obtained on well-formed datasets. However, our results are quite successful when compared to the previous studies on users generated contents that are written in Turkish and

collected from different OSNs. To achieve higher results, it is required to use additional features as NER is a completely domain-specific task. Performing NER on OSN content is a challenging task when compared to the news domain and how to increase performance on OSN content is still an open issue. There is a need for large datasets and gazetteers especially for NER on OSN contents.

On the other hand, in our SA task, we performed SA of Facebook data in the context of the Turkish language. Based on extensive experimental results, we conclude that DL methods often outperform traditional ML methods, but it is required to employed a parameter optimization process for the DL side to detect the optimum structure and parameters of the related network. OSN contents often include dirty texts, therefore it is a challenge for text mining tasks over OSNs. To obtain better results for SA over OSNs, it is needed to use corpora from the same problem domain; especially for training the word embedding model and generating lexicon-based features. Facebook reactions often may not provide a clue about the sentiment of the related activity. Users, on the other hand, are more engaged in negatively oriented activities and are willing to put more effort into commenting/replying to the posts/comments of male users.

The widespread use of OSNs has brought forward the issue of privacy as users' sensitive information needs to remain private. This is because users are not often aware of possible privacy risks when they publish information in their accounts. A measure of how much sensitive information users shared with others on an OSN would help users to understand whether they individually share too much. Motivating this, in this thesis, we also performed a privacy risk analysis of Turkish Facebook users.

In our preliminary task, we performed a privacy risk analysis of users by using IPS and NPS methods. Our results show that males are often at higher risk than females and users who are in a range of 21 to 40 often have a higher risk of privacy. The most important output of this task is that it is possible to determine the extent of the appearance of the information on a survey or synthetic data, but it is

not possible when accessing OSN data by a crawler. This case has a direct effect on the R matrix which is used by most of the existing privacy scoring methods to obtain sensitivity and visibility components.

In our extended analysis, finally, we performed a comprehensive privacy risk analysis on the largest snapshot of the crawled Facebook data. In this task, we proposed a framework which is involving our novel aspects along with existing methods. These novel aspects are: (i) populating the R matrix by using attribute inference and (ii) obtaining network aware-risk scores not just by using user connections but also using weights of these connections as well. Based on our results, we conclude that social tie strengths of users' connections have to be taken into consideration when computing their centrality or network-aware risk scores.

It is often possible to infer the private attributes of users by using network structure information. However, the network structure is not enough to perform inference in some cases due to OSN data is often incomplete. Content-based inference, on the other hand, is very challenging compared to the inference based on network structure. This is because OSN textual data is often dirty and written by using informal language. Performing such an inference task without a learning model requires many rules, sources, and so on. Using a learning model, on the other hand, requires a labeled corpus for each related inference task. Using a learning model may improve the accuracy of attribute inference for some attributes, but some others may still be challenging like OKL due to inferring such items require to combine and apply multiple learning tasks such as NER, WSD, and ML.

As future work, we will use both nature-inspired feature selection algorithms and language models like BERT to improve results of our content-based tasks. We also planning to anonymize our crawled data to share it with the OSN research community.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., and Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, USA, 265-283.

Abdesslem, F. B., Parris, I., and Henderson, T. 2012. Reliable online social network data collection (A. Abraham editor). Computational Social Networks, Springer, London, pp. 183-210.

Abid, Y., Imine, A., and Rusinowitch, M. 2018. Sensitive attribute prediction for social networks users. International workshop on data analytics solutions for real-life applications, Vienne, Austria, 28-35.

Aghasian, E., Garg, S., Gao, L., Yu, S., and Montgomery, J. 2017. Scoring users' privacy disclosure across multiple online social networks. IEEE access, 5: 13118-13130.

Aghasian, E., Garg, S., and Montgomery, J. 2018. A privacy-enhanced friending approach for users on multiple online social networks. Computers, 7(3): 1-12.

Agirre, E., and Soroa, A. 2009. Personalizing pagerank for word sense disambiguation. Proceedings of the 12th Conference of the European Chapter of the ACL, Athens, Greece, 33-41.

Ahkter, J. K., and Soria, S. 2010. Sentiment analysis: Facebook status messages. MSc Thesis (Unpublished), Stanford University.

Akaichi, J., Dhouioui, Z., and Pérez, M. J. L. H. 2013. Text mining facebook status updates for sentiment classification. IEEE International conference on system theory, control and computing, Sinaia, Romania, 640-645.

Akcora, C., Carminati, B., and Ferrari, E. 2012. Privacy in social networks: How risky is your social graph?. IEEE International Conference on Data Engineering, Washington, USA, 9-19.

Akın, A. A., and Akın, M. D. 2007. Zemberek, an open source nlp framework for turkic languages. Structure, 10: 1-5.

Aktas, Ö., Birant, Ç. C., Aksu, B., and Çebi, Y. 2013. Automated synonym dictionary generation tool for Turkish (ASDICT). Journal of Social Sciences of the Turkic World, 65: 47-68.

Akter, S., and Aziz, M. T. 2016. Sentiment analysis on facebook group using lexicon-based approach. IEEE International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 1-4.

Alemany, J., del Val, E., Alberola, J., and García-Fornes, A. 2018. Estimation of privacy risk through centrality metrics. Future Generation Computer Systems, 82: 63-76.

Alowibdi, J. S., Buy, U. A., and Yu, P. 2013a. Language independent gender classification on Twitter. Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining, Niagara Falls, Canada, 739-743.

Alowibdi, J. S., Buy, U. A., and Yu, P. 2013b. Empirical evaluation of profile characteristics for gender classification on twitter. IEEE International Conference on Machine Learning and Applications, Miami, Florida, USA, 365-369.

Alpaydin, E. 2014. Introduction to machine learning. MIT press, Cambridge, Massachusetts, USA, 659.

Al-Asmari, H. A., and Saleh, M. S. 2019. A Conceptual Framework for Measuring Personal Privacy Risks in Facebook Online Social Network. IEEE International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia, 1-6.

Al-Lozi, E. 2011. Explaining users' intentions to continue participating in Web 2.0 communities: The case of Facebook in the Hashemite Kingdom of Jordan. PhD Thesis. Brunel University, School of Information Systems, Computing and Mathematics.

Amasyalı, M. F., and Diri, B. 2006. Automatic Turkish text categorization in terms of author, genre and gender. International Conference on Application of Natural Language to Information Systems, Klagenfurt, Austria, 221-226.

Anonymous., 2016. Github Repo: Liseler-Ilkokullar. [Online] Available: https://github.com/SqlHareketi/Liseler-Ilkokullar/blob/master/liseler-ilkokullar.sql

Anonymous., 2019a. The Selenium project and tools. [Online] Available: https://selenium.dev/documentation/en/introduction/the_selenium_project_and_tools/

Anonymous., 2019b. Wikipedia entry: Tarjan's strongly connected components algorithm. [Online] Available:https://en.wikipedia.org/wiki/Tarjan%27s_strongly_connected_components_algorithm#cite_note-1

Anonymous., 2019c. Strongly Connected Components algorithm sample. [Online] Available:https://neo4j.com/docs/graph-algorithms/current/labs algorithms /strongly-connected-components/index.html

Anonymous., 2019d. Wikipedia enrty: Depth-first search. [Online] Available: https://en.wikipedia.org/wiki/Depth-first_search

Anonymous., 2019e. Wikipedia enrty: Breadth-first search. [Online] Available: https://en.wikipedia.org/wiki/Breadth-first_search

Anonymous., 2020a. Facebook First Quarter 2020 Results. [Online] Available: https://investor.fb.com/investor-news/press-release-details/2020/Facebook-Reports-First-Quarter-2020-Results/default.aspx

Anonymous., 2020b. Wikipedia entry: Pearson correlation coefficient [Online] Available: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Anonymous., 2020c. Wikipedia entry: Chi-squared test [Online] Available: https://en.wikipedia.org/wiki/Chi-squared_test

Backstrom, L., Dwork, C., and Kleinberg, J. 2007. Wherefore art thou R3579X? Anonymized social networks, hidden patterns, and structural steganography. Proceedings of the 16th international conference on World Wide Web, Banff Alberta, Canada, 181-190.

Balahur, A., and Jacquet, G. 2015. Sentiment analysis meets social media–Challenges and solutions of the field in view of the current information sharing context. Information Processing & Management, 51(4): 428-432.

Balakrishnan, V. K. 1997. Schaum's Outline of Graph Theory: Including Hundreds of Solved Problems, McGraw Hill Professional, New York, USA, 312p.

Banks, L., and Wu, S. F. 2009. All friends are not created equal: An interaction intensity-based approach to privacy in online social networks. IEEE International Conference on Computational Science and Engineering, Vancouver, Canada, 970-974.

Bassem, B., and Zrigui, M. 2018. Gender Identification: A Comparative Study of Deep Learning Architectures. International Conference on Intelligent Systems Design and Applications, Vellore, India, 792-800.

Bastian, M., Heymann, S., and Jacomy, M. 2009. Gephi: an open source software for exploring and manipulating networks. International AAAI conference on weblogs and social media, California, USA, 361-362.

Bayraktar, O., and Temizel, T. T. 2008. Person name extraction from Turkish financial news text using local grammar-based approach. IEEE International Symposium on Computer and Information Sciences, İstanbul, Turkey, 1-4.

Becker, J. L., and Chen, H. 2009. Measuring privacy risk in online social networks. Proceedings of the Web 2.0 Security and Privacy (W2SP), Oakland, California, USA, 2095-2100.

Bennacer, N., Jipmo, C. N., Penta, A., and Quercini, G. 2014. Matching user profiles across social networks. International Conference on Advanced Information Systems Engineering, Thessaloniki, Greece, 424-438.

Berndt, D. J., and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. Proceedings of the KDD workshop on Knowledge Discovery in Databases, Seattle, Washington, 359-370.

Bewick, V., Cheek, L., and Ball, J. 2005. Statistics review 14: Logistic regression. Critical care, 9(1): 112-118.

Bhat, S. I., Arif, T., and Malik, M. B. 2018. A framework for user identity resolutions across social networks. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 4(1): 307-313.

Biega, J. A., Gummadi, K. P., Mele, I., Milchevski, D., Tryfonopoulos, C., and Weikum, G. 2016. R-susceptibility: An ir-centric approach to assessing privacy risks for users in online communities. Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, 365-374.

Bioglio, L., and Pensa, R. G. 2017. Impact of neighbors on the privacy of individuals in online social networks. International Conference on Computational Science, Zürich, Switzerland, 28-37.

Bioglio, L., Capecchi, S., Peiretti, F., Sayed, D., Torasso, A., and Pensa, R. G. 2018. A social network simulation game to raise awareness of privacy among school children. IEEE Transactions on Learning Technologies, 12(4): 456-469.

Bisong, E. 2019. Google Colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA, 709.

Błachnio, A., Przepiórka, A., and Rudnicka, P., 2013. Psychological determinants of using Facebook: A research review. International Journal of Human-Computer Interaction, 29(11): 775-787.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. 2013. Twitie: An open-source information extraction pipeline for microblog text. Proceedings of the international conference recent advances in natural language processing (RANLP), Hissar, Bulgaria, 83-90.

Braunstein, A., Granka, L., and Staddon, J. 2011. Indirect content privacy surveys: measuring privacy without asking about it. Proceedings of the Seventh Symposium on Usable Privacy and Security, Pittsburgh, Pennsylvania, USA, 1-14.

Breiman, L. 1999. Random forests-random features. Statistics Department, UC Berkeley, Technical Report 576.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems, 30(1): 107-117.

Brinkmeier, M. 2006. PageRank revisited. ACM Transactions on Internet Technology (TOIT), 6(3): 282-301.

Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. 2011. Discriminating gender on Twitter. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, 1301-1309.

Cadwalladr, C., and Graham-Harrison, E. 2018. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. The guardian, [Online]. Available: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.

Calabresi, M., 2017. Inside russia's social media war on America, Online Journal, Time. [Online] Available: https://time.com/4783932/inside-russia-social-media-war-america/

Caliskan Islam, A., Walsh, J., and Greenstadt, R. 2014. Privacy detective: Detecting private information and collective privacy behavior in a large social network. Proceedings of the 13th Workshop on Privacy in the Electronic Society, Scottsdale, Arizona, USA, 35-46.

Canaydın, A., 2017. Github Repo: alpcanaydin/liseler. [Online] Available: https://github.com/alpcanaydin/liseler/blob/master/liseler-web/public/data-.json

Caramujo, J., and Da Silva, A. M. R. 2015. Analyzing privacy policies based on a privacy-aware profile: The Facebook and LinkedIn case studies. IEEE 17th Conference on Business Informatics, Lisbon, Portugal, 77-84.

Catanese, S. A., De Meo, P., Ferrara, E., Fiumara, G., and Provetti, A. 2011. Crawling facebook for social network analysis purposes. Proceedings of the international conference on web intelligence, mining and semantics, Sogndal, Norway, 1-8.

Cetto, A., Netter, M., Pernul, G., Richthammer, C., Riesner, M., Roth, C., and Sänger, J. 2014. Friend inspector: A serious game to enhance privacy awareness in social networks. arXiv preprint arXiv:1402.5878.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16: 321-357.

Chen, H. P., Hsu, K. W., and Chiu, S. I. 2016. Event Detection in an ego Network on Facebook. Pacific Asia Conference on Information Systems (PACIS), Chiayi, Taiwan, 172.

Chollet F., 2015. Keras: Deep learning library for theano and tensorflow. [Online]. Avaiable: https://keras. io/k

Ciot, M., Sonderegger, M., and Ruths, D. 2013. Gender inference of Twitter users in non-English contexts. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, 1136-1145.

Cliche, M. 2017. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. arXiv preprint arXiv:1704.06125.

Conti, M., Poovendran, R., and Secchiero, M. 2012. Fakebook: Detecting fake profiles in on-line social networks. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Istanbul, Turkey, 1071-1078.

Cutillo, L. A., Molva, R., and Strufe, T. 2009. Safebook: A privacy-preserving online social network leveraging on real-life trust. IEEE Communications Magazine, 47(12): 94-101.

Çoban, Ö., and Özel, S. A. 2018. Utilizing Language Model for Term Weighting in Text Categorization. IEEE International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 1-5.

Çoban, Ö., and Özyer, G. T. 2018. Twitter duygu analizinde terim ağırlıklandırma yönteminin etkisi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 24(2): 283-291.

Çoban, Ö., Ozyildirim, B. M., and Ozel, S. A. 2018. An empirical study of the extreme learning machine for Twitter sentiment analysis. International Journal of Intelligent Systems and Applications in Engineering, 6(3): 178-184.

Çoban, Ö., Inan, A., and Ozel, S. A. 2020a. Towards the design and implementation of an OSN crawler: a case of Turkish Facebook users. International Journal of Information Security Science, 9(2): 76-93.

Çoban, Ö., Inan, A., and Ozel, S. A. 2020b. Privacy Risk Analysis for Facebook Users. The 28th IEEE Conference on Signal Processing and Communications Applications, Gaziantep, Turkey.

Çoban, Ö., Ozel, S. A., and Inan, A. 2021. Deep Learning-based Sentiment Analysis of Facebook Data: The Case of Turkish Users. The Computer Journal (in press).

Comer, R., McKelvey, N., and Curran, K., 2012. Privacy and Facebook. International Journal of Engineering and Technology, 2(9): 1626-1630.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., 2009. Introduction to algorithms. MIT press, Cambridge, Massachusetts, USA, 1128.

Corriga, A., Cusimano, S., Malloci, F., Marchesi, L., and Recupero, D. R. 2018. Leveraging Cognitive Computing for Gender and Emotion Detection. Proceedings of 4th Workshop on Sentic Computing, Sentiment Analysis, Opinion Mining, and Emotion Detection Co-located with the 15th Extended Semantic Web Conference 2018, Heraklion, Greece, 47-56.

Cristianini, N., and Shawe-Taylor, J. 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, Cambridge, UK, 187.

Çelik, Ö., and Aslan, A. F. 2019. Gender Prediction from Social Media Comments with Artificial Intelligence. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 23(6): 1256-1264.

Çelikkaya, G., Torunoğlu, D., and Eryiğit, G. 2013. Named entity recognition on real data: a preliminary investigation for Turkish. IEEE 7th International Conference on Application of Information and Communication Technologies, Azerbaijan, Baku, 1-5.

Dalkılıç, F. E., Gelişli, S., and Diri, B. 2010. Named Entity Recognition from Turkish texts. IEEE Signal Processing and Communications Applications, Diyarbakır, Turkey, 918-920.

Dehkharghani, R., Saygin, Y., Yanikoglu, B., and Oflazer, K. 2016. SentiTurkNet: a Turkish polarity lexicon for sentiment analysis. Language Resources and Evaluation, 50(3): 667-685.

Demir, H., and Özgür, A. 2014. Improving named entity recognition for morphologically rich languages using word embeddings. IEEE 13th International Conference on Machine Learning and Applications, Detroit, USA, 117-122.

Deitrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T., and Hu, W. 2012. Gender identification on twitter using the modified balanced winnow. Communications and Network, 4(3): 189-195.

Dey, R., Tang, C., Ross, K., and Saxena, N. 2012. Estimating age privacy leakage in online social networks. Annual IEEE International Conference on Computer Communications, Orlando, Florida, USA, 2836-2840.

Dey, A. 2016. Machine learning algorithms: a review. International Journal of Computer Science and Information Technologies, 7(3): 1174-1179.

Djoudi, A., and Pujolle, G. 2019. Social Privacy Score through Vulnerability Contagion Process. IEEE Fifth Conference on Mobile and Secure Services (MobiSecServ), Miami Beach, Florida, USA, 1-6.

Do, T. N., and Poulet, F. 2015. Parallel multiclass logistic regression for classifying large scale image datasets. Advanced Computational Methods for Knowledge Engineering, Metz, France, 255-266.

Doğan, E. 2011. Türkiye Türkçesinde Cinsiyet Kategorisinin İzleri. Journal of International Social Research, 4(17): 89-98.

Domingo-Ferrer, J. 2010. Rational privacy disclosure in social networks. International conference on modeling decisions for artificial intelligence, Perpignan, France, 255-265.

Dprogrammer, 2019. RNN, LSTM, and GRU. [Online]. Available: http://dprogrammer.org/rnn-lstm-gru

Eken, B. 2015. Kısa Metinlerde Varlık İsmi Tanıma, MSc Thesis, Istanbul Technical University.

Eken, B., and Tantug A.C., 2015. Recognizing named entities in turkish tweets. Proceedings of the Fourth International Conference on Software Engineering and Applications, Dubai, UAE, 450-454.

El Naqa, I., and Murphy, M. J. 2015. What is machine learning? (I. El Naqa, R. Li, M. Murphy editors). Machine learning in radiation oncology, Springer, Cham, 3-11.

Ellison, N. B., Steinfield, C., and Lampe, C., 2007. The benefits of Facebook "friends:" Social capital and college students' use of online social network sites. Journal of computer-mediated communication, 12(4): 1143-1168.

Emekligil, E., Arslan, S., and Agin, O. 2016. A bank information extraction system based on named entity recognition with CRFs from noisy customer order texts in Turkish. In International Conference on Knowledge Engineering and the Semantic Web, Prague, Czech Republic, 93-102.

Erciyes, K., 2018. Guide to Graph Algorithms. Springer, Cham, 471.

Erşahin, B., Aktaş, Ö., Kilinc, D., and Erşahin, M. 2019. A hybrid sentiment analysis method for Turkish. Turkish Journal of Electrical Engineering & Computer Sciences, 27(3): 1780-1793.

Ertopçu, B., Kanburoğlu, A. B., Topsakal, O., Açıkgöz, O., Gürkan, A. T., Özenç, B., and Yıldız, O. T. 2017. A new approach for named entity recognition. In 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 474-479.

Facebook, 2019. Facebook Reports First Quarter 2019 Results. [Online]. Available: https://investor.fb.com/investor-news/press-release-details/2019/Facebook-Reports-First-Quarter-2019-Results/default.aspx

Fang, Q., Sang, J., Xu, C., and Hossain, M. S. 2015. Relational user attribute inference in social media. IEEE Transactions on Multimedia, 17(7): 1031-1044.

Fang, Z., Cao, Y., Liu, Y., Tan, J., Guo, L., and Shang, Y. 2017. A co-training method for identifying the same person across social networks. IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, Canada, 1412-1416.

Farha, I. A., and Magdy, W. 2019. Mazajak: An online Arabic sentiment analyser. Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 192-198.

Fatima, M., Hasan, K., Anwar, S., and Nawab, R. M. A. 2017. Multilingual author profiling on Facebook. Information Processing & Management, 53(4): 886-904.

Fire, M., Kagan, D., Elyashar, A., and Elovici, Y. 2014. Friend or foe? Fake profile identification in online social networks. Social Network Analysis and Mining, 4(1): 1-23.

Flores, G., Lorena, A., Penteado, C., and Kamienski, C., 2013. Can Social Network Influence Voters?. Brazilian Symposium on Computer Networks and Distributed Systems (SBRC), Brasilia, Brazil, 3-8.

Fox, J., Zou, Y., and Qiu, J. 2016. Software frameworks for deep learning at scale. Internal Indiana University, Technical Report.

Gamboa, A. M., and Gonçalves, H. M. 2014. Customer loyalty through social networks: Lessons from Zara on Facebook. Business horizons, 57(6): 709-717.

Garside, J. 2015. Twitter puts trillions of tweets up for sale to data miners. The Guardian. [Online]. Available: https://www.theguardian.com/technology/2015/mar/18/twitter-puts-trillions-tweets-for-sale-data-miners

Gayo Avello, D. 2011. All liaisons are dangerous when all your friends are known to us. Proceedings of the 22nd ACM conference on Hypertext and hypermedia, Eindhoven, Netherlands, 171-180.

Gezici, G., and Yanıkoğlu, B. 2018. Sentiment analysis in Turkish (K. Oflazer, M. Saraçlar editors). Turkish Natural Language Processing, Springer, Cham, 255-271.

Giannakopoulos, O., Kalatzis, N., Roussaki, I., and Papavassiliou, S. 2018. Gender recognition based on social networks for multimedia production. IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), Zagori, Aristi Village, Greece, 1-5.

Gilbert, E., and Karahalios, K. 2009. Predicting tie strength with social media. Proceedings of the SIGCHI conference on human factors in computing systems, Boston, Massachusetts, USA, 211-220.

Giuntini, F. T., Ruiz, L. P., Kirchner, L. D. F., Passarelli, D. A., Dos Reis, M. D. J. D., Campbell, A. T., and Ueyama, J. 2019. How do i feel? identifying emotional expressions on facebook reactions using clustering mechanism. IEEE Access, 7: 53909-53921.

Goga, O., Loiseau, P., Sommer, R., Teixeira, R., and Gummadi, K. P. 2015. On the reliability of profile matching across large online social networks. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 1799-1808.

Golub, G. H., and Van der Vorst, H. A. 2000. Eigenvalue computation in the 20th century. Journal of Computational and Applied Mathematics, 123(1-2): 35-65.

Gomaa, W. H., and Fahmy, A. A. 2013. A survey of text similarity approaches. International Journal of Computer Applications, 68(13): 13-18.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. 2018. Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893.

Gross, R., and Acquisti, A. 2005. Information revelation and privacy in online social networks. Proceedings of the 2005 ACM workshop on Privacy in the electronic society, Virginia, USA, 71-80.

Gundecha, P., Barbier, G., and Liu, H. 2011. Exploiting vulnerability to secure user privacy on a social networking site. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, Halifax, Canada, 511-519.

Gupta, A., and Kaushal, R. 2017. Towards detecting fake user accounts in Facebook. ISEA Asia Security and Privacy Conference (ISEASP), Surat, India, 1-6.

Habernal, I., Ptáček, T., and Steinberger, J. 2013. Sentiment analysis in czech social media using supervised machine learning. Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis, Atlanta, Georgia, USA, 65-74.

Hagberg, A., Swart, P., and Schult, D. 2008. Exploring network structure, dynamics, and function using NetworkX. Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, USA, 11-15.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H., 2009. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1): 10-18.

Han, H., Wang, W. Y., and Mao, B. H. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Advances in Intelligent Computing, International Conference on Intelligent Computing, Hefei, China, 878-887.

Han, K., Jo, Y., Jeon, Y., Kim, B., Song, J., and Kim, S. W. 2018. Photos don't have me, but how do you know me? Analyzing and predicting users on Instagram. Adjunct publication of the 26th conference on user modeling, adaptation and personalization, Singapore, 251-256.

Hangal, S., MacLean, D., Lam, M. S., and Heer, J. 2010. All friends are not equal: Using weights in social graphs to improve search. Proceedings of the SNA-KDD 2010: The 4th International Workshop on Social Network Mining and Analysis, Washington, USA, 1-7.

Hart, P. 1968. The condensed nearest neighbor rule. IEEE transactions on information theory, 14(3): 515-516.

Hasan, K. S., ur Rahman, M. A., and Ng, V. 2009. Learning-based named entity recognition for morphologically-rich, resource-scarce languages. Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, Greece, 354-362.

Hassan, A., and Mahmood, A. 2017. Deep learning approach for sentiment analysis of short texts. IEEE 3rd international conference on control, automation and robotics (ICCAR), Nagoya, Japan, 705-710.

Haveliwala, T. H., 2002. Topic-sensitive pagerank. Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, USA, 517–526.

Hazimeh, H., Mugellini, E., Abou Khaled, O., and Cudré-Mauroux, P. 2017. SocialMatching++: A Novel Approach for Interlinking User Profiles on Social Networks. Proceedings of the 4th International Workshop on Dataset profiling and federated Search for Web Data co-located with The 16th International Semantic Web Conference, Vienna, Austria.

He, H., Bai, Y., Garcia, E. A., and Li, S. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE International Joint Conference on Neural Networks (IJCNN), Hong Kong, China, 1322-1328.

He, H., and Garcia, E. A. 2009. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9): 1263-1284.

Ilić, J., Humski, L., Pintar, D., Vranić, M., and Skočir, Z. 2016. Proof of concept for comparison and classification of online social network friends based on tie strength calculation model. Proceedings of the 6th interantional conference on information society and technology, Kopaonik, Serbia, 159-164.

Illmer, A. 2016. Social Media: A hunting ground for cybercriminals. BBC. [Online] Available: http://www. bbc. co. uk/news/business-36854285

Iram, A. 2018. Sentiment Analysis of Student's Facebook Posts. International Conference on Intelligent Technologies and Applications, Bahawalpur, Pakistan, 86-97.

Islam, M. S., Islam, M. A., Hossain, M. A., and Dey, J. J. 2016. Supervised approach of sentimentality extraction from bengali facebook status. IEEE 19th International Conference on Computer and Information Technology (ICCIT), Istanbul, Turkey, 383-387.

Jajodia, S., and Meadows, C. 1995. Inference problems in multilevel secure database management systems. Information Security: An integrated collection of essays, 1: 570-584.

Jansen, B. J., Sobel, K., and Cook, G. 2011. Classifying ecommerce information sharing behaviour by youths on social networking sites. Journal of Information Science, 37(2): 120-136.

Jeh, G., and Widom, J. 2003. Scaling personalized web search. Proceedings of the 12th international conference on World Wide Web, New York, USA, 271-279.

Jernigan, C., and Mistree, B. F. 2009. Gaydar: Facebook friendships expose sexual orientation. First Monday, 14(10).

Jiang, F., Leung, C. K. S., and Tanbeer, S. K. 2012. Finding popular friends in social networks. IEEE Second International Conference on Cloud and Green Computing, Xiangtan, China, 501-508.

Jiang, R., Banchs, R. E., and Li, H. 2016. Evaluating and combining name entity recognition systems. Proceedings of the Sixth Named Entity Workshop, Berlin, Germany, 21-27.

Jin, C., Ma, T., Hou, R., Tang, M., Tian, Y., Al-Dhelaan, A., and Al-Rodhaan, M. 2015. Chi-square statistics feature selection based on term frequency and distribution for text categorization. IETE journal of research, 61(4): 351-362.

Jordan, M. I., and Mitchell, T. M. 2015. Machine learning: Trends, perspectives, and prospects. Science, 349(6245): 255-260.

Kahanda, I., and Neville, J. 2009. Using transactional information to predict link strength in online social networks. Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media, San Jose, California, 74-81.

Kalender, M., and Korkmaz, E. E. 2017. Turkish entity discovery with word embeddings. Turkish Journal of Electrical Engineering & Computer Sciences, 25(3): 2388-2398.

Kanaris, I., Kanaris, K., Houvardas, I., and Stamatatos, E. 2007. Words versus character n-grams for anti-spam filtering. International Journal on Artificial Intelligence Tools, 16(6): 1047-1067.

Karpathy, A., Johnson, J., and Fei-Fei, L. 2015. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.

Kastrati, Z., Imran, A. S., Yildirim-Yayilgan, S., and Dalipi, F. 2015. Analysis of online social networks posts to investigate suspects using SEMCON. International Conference on Social Computing and Social Media, Los Angeles, California, USA, 148-157.

Keeshin, J., Galant, Z., and Kravitz, D. 2010. Machine Learning and Feature Based Approaches to Gender Classification of Facebook Statuses. Eprint Arxiv.

Keogh, E. J., and Pazzani, M. J. 2000. Scaling up dynamic time warping for datamining applications. Proceedings of the sixth SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts, USA, 285-289.

Khan, A., Baharudin, B., Lee, L. H., and Khan, K. 2010. A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1): 4-20.

Khandelwal, A. 2019. Towards Identifying Humor and Author's Gender in Code-mixed Social Media Content. PhD Thesis. International Institute of Information Technology Hyderabad.

Khoshgoftaar, T. M., Golawala, M., and Van Hulse, J. 2007. An empirical study of learning from imbalanced data using random forest. IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Patras, Greece, 310-317.

Kılıç, Y., and Inan, A. 2019a. Privacy Scoring Over Professional OSNs: More Central Users are Under Higher Risk. International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES2019), Alanya, Turkey, 157-161.

Kılıç, Y., and Inan, A. 2019b. Implementing A Web Crawler with An Attacker Perspective on A Professional Purpose Online Social. Proceedings of the International Conference on All Aspects of Cyber Security, Adana, Turkey, 27-32.

Kılıç, Y., and Inan A. 2021. QPR-EVAL: A Quantitative Framework For Privacy Risk Score Evaluation. Unpublished manuscript.

Kibriya, A. M., Frank, E., Pfahringer, B., and Holmes, G. 2004. Multinomial naive bayes for text categorization revisited. Australasian Joint Conference on Artificial Intelligence, Cairns, Queensland, Australia, 488-499.

Kilimci, Z. H., and Akyokus, S. 2018. Deep learning-and word embedding-based heterogeneous classifier ensembles for text classification. Complexity, 2018: 1-10.

Kim, Y. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., and Klein, M. 2002. Logistic regression. Springer-Verlag, New York, USA, 701.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th international joint conference on Artificial intelligence, San Francisco, California, USA, 1137-1145.

Kokkos, A., and Tzouramanis, T. 2014. A robust gender inference model for online social networks and its application to linkedin and twitter. First Monday, 19(9-1): 1-34.

Konkol, M., and Konopík, M. 2015. Segment representations in named entity recognition. International Conference on Text, Speech, and Dialogue, Pilsen, Czech Republic, 61-70.

Korolova, A., Motwani, R., Nabar, S. U., and Xu, Y. 2008. Link privacy in social networks. Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, 289-298.

Kosinski, M., Stillwell, D., and Graepel, T. 2013. Private traits and attributes are predictable from digital records of human behavior. Proceedings of the National Academy of Sciences, 110(15): 5802-5805.

Köksal, A., 2018. Turkish-Word2Vec. [Online]. Available: https://github.com/akoksal/Turkish-Word2Vec/wiki.

Krakan, S., Humski, L., and Skočir, Z. 2018. Determination of friendship intensity between online social network users based on their interaction. Tehnički vjesnik, 25(3): 655-662.

Krawczyk, B. 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4): 221-232.

Krebs, F., Lubascher, B., Moers, T., Schaap, P., and Spanakis, G. 2017. Social emotion mining techniques for Facebook posts reaction prediction. arXiv preprint arXiv:1712.03249.

Kridalukmana, R. 2015. Generic social network data crawler using attributed graph. IEEE 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 138-142.

Krishnamurthy, B., and Wills, C. E. 2009. On the leakage of personally identifiable information via online social networks. Proceedings of the 2nd ACM workshop on Online social networks, Barcelona, Spain, 7-12.

Kuru, O., Can, O. A., and Yuret, D. 2016. Charner: Character-level named entity recognition. Proceedings of the 26th International Conference on Computational Linguistics (COLING), Osaka, Japan, 911-921.

Kucukyilmaz, T., Cambazoglu, B. B., Aykanat, C., and Can, F. 2006. Chat mining for gender prediction. International conference on advances in information systems, Izmir, Turkey, 274-283.

Küçük, D., and Steinberger, R. 2014. Experiments to improve named entity recognition on Turkish tweets. arXiv preprint arXiv:1410.8668.

Küçük, D., Jacquet, G., and Steinberger, R. 2014. Named entity recognition on Turkish tweets. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 450-454.

Küçük, D., 2015. Automatic compilation of language resources for named entity recognition in Turkish by utilizing Wikipedia article titles. Computer Standards & Interfaces, 41: 1-9.

Küçük, D., Küçük, D., and Arıcı, N. 2016a. A named entity recognition dataset for Turkish. IEEE 24th Signal Processing and Communication Application Conference, Zonguldak, Turkey, 329-332.

Küçük, D., and Arıcı, N. 2016b. Türkçe için Wikipedia Tabanlı Varlık İsmi Tanıma Sistemi. Politeknik Dergisi, 19(3): 325-332.

Küçük, D., Arıcı, N., and Küçük, D. 2017. Named entity recognition in Turkish: Approaches and issues. International Conference on Applications of Natural Language to Information Systems, Liège, Belgium, 176-181.

Küçük, D., and Yazici, A. 2011. A hybrid named entity recognizer for Turkish with applications to different text genres (E. Gelenbe, R. Lent, G. Sakellari, A. Sacan, H. Toroslu, A. Yazici editors). Computer and Information Sciences, Springer, Dordrecht, 113-116.

Lafferty, J., McCallum, A., and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the Eighteenth International Conference on Machine Learning (ICML), Williamstown, MA, USA, 282-289.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. International conference on machine learning, Beijing, China, 1188-1196.

Lemaître, G., Nogueira, F., and Aridas, C. K. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research, 18(1): 559-563.

Leskovec, J., and Krevl, A. 2014. SNAP datasets: stanford large network dataset collection. [Online]. Available: http://snap.stanford.edu/data.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady, 10(8): 707-710.

Li, X., Yang, Y., Chen, Y., and Niu, X. 2018. A privacy measurement framework for multiple online social networks against social identity linkage. Applied Sciences, 8(10): 1790.

Li, Y., Peng, Y., Ji, W., Zhang, Z., and Xu, Q. 2017. User identification based on display names across online social networks. IEEE Access, 5: 17342-17353.

Li, Y., Peng, Y., Zhang, Z., Wu, M., Xu, Q., and Yin, H. 2018. A deep dive into user display names across social networks. Information Sciences, 447: 186-204.

Li, Y., Peng, Y., Zhang, Z., Yin, H., and Xu, Q. 2019. Matching user accounts across social networks based on username and display name. World Wide Web, 22(3): 1075-1097.

Liang, J., Hu, Q., Dang, C., and Zuo, W. 2018. Weighted graph embedding-based metric learning for kinship verification. IEEE Transactions on Image Processing, 28(3): 1149-1162.

Liang, T., Yang, X., Wang, L., and Han, Z. 2018. Kinship Determination in Mobile Social Networks. IEEE 23rd International Conference on Engineering of Complex Computer Systems (ICECCS), Melbourne, Australia, 205-208.

Lindamood, J., Heatherly, R., Kantarcioglu, M., and Thuraisingham, B. 2009. Inferring private information using social network data. Proceedings of the 18th international conference on World wide web, Madrid, Spain, 1145-1146.

Lipton, Z. C., Berkowitz, J., and Elkan, C. 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

Liu, K., and Terzi, E. 2010. A framework for computing the privacy scores of users in online social networks. ACM Transactions on Knowledge Discovery from Data, 5(1): 1-30.

Liu, Q., Li, J., Wang, Y., Xing, G., and Ren, Y. 2014. Account matching across heterogeneous networks. IEEE 5th International Conference on Game Theory for Networks, Beijing, China, 1-5.

Liu, Z., Li, H., and Wang, C. 2020. NEW: A Generic Learning Model for Tie Strength Prediction in Networks. Neurocomputing, 406: 282-292.

Lobur, M., Romanyuk, A., and Romanyshyn, M. 2011. Using NLTK for educational and scientific purposes. IEEE 11th international conference the experience of designing and application of CAD systems in microelectronics (CADSM), Polyana-Svalyava, Ukraine, 426-428.

Loper, E., and Bird, S. 2002. NLTK: the natural language toolkit. Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Philadelphia, Pennsylvania, USA, 63-70.

Louppe, G. 2017. Bayesian Optimisation with Scikit-Optimize. [Online]. Available: https://orbi.uliege.be/bitstream/2268/226433/1/PyData%202017_%20Bayesian%20optimization%20with%20Scikit-Optimize.pdf

Ljubesic, N., Boras, D., Bakaric, N., and Njavro, J. 2008. Comparing measures of semantic similarity. IEEE 30th International Conference on Information Technology Interfaces, Dubrovnik, Croatia, 675-682.

Malakasiotis, P., and Androutsopoulos, I. 2007. Learning textual entailment using SVMs and string similarity measures. Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing, Prague, Czech Republic, 42-47.

Malhotra, A., Totti, L., Meira Jr, W., Kumaraguru, P., and Almeida, V. 2012. Studying user footprints in different online social networks. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Istanbul, Turkey, 1065-1070.

Mao, H., Shuai, X., and Kapadia, A. 2011. Loose tweets: an analysis of privacy leaks on twitter. Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society, London, UK, 1-12.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. [Online]. Available: http://mallet.cs.umass.edu.

Meire, M., Ballings, M., and Van den Poel, D. 2016. The added value of auxiliary data in sentiment analysis of Facebook posts. Decision Support Systems, 89: 98-112.

Metin, S. K., Kisla, T., and Karaoglan, B. 2012. Named entity recognition in Turkish using association measures. Advanced Computing, 3(4): 43-49.

Mfenyana, S. I., Moorosi, N., and Thinyane, M. 2014. Facebook Crawler Architecture for Opinion Monitoring and Trend Analysis Purposes. Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Eastern Cape, South Africa, 191-196.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 26 (NIPS 2013), Lake Tahoe, Nevada, USA, 3111-3119.

Minkus, T., and Memon, N. 2014. On a scale from 1 to 10, how private are you? scoring facebook privacy settings. Proceedings of the Workshop on Usable Security (USEC 2014), San Diego, California, USA, 1-6.

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B., 2007. Measurement and analysis of online social networks. Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, California, USA, 29-42.

Mislove, A., Viswanath, B., Gummadi, K. P., and Druschel, P. 2010. You are who you know: inferring user profiles in online social networks. Proceedings of the third ACM international conference on Web search and data mining, New York, USA, 251-260.

Mittal, S., and Sahu, G. 2017. Twitter Crawler with MultilingualText Classification. International Journal of Innovations & Advancement in Computer Science, 6(6): 77-83.

Modi, Y. A., and Gandhi, I. S. 2014. Internet sociology: Impact of Facebook addiction on the lifestyle and other recreational activities of the Indian youth. In SHS Web of Conferences, 5(00001): 1-4.

Mohammed, R., Rawashdeh, J., and Abdullah, M. 2020. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. IEEE 11th International Conference on Information and Communication Systems (ICICS), Copenhagen, Denmark, 243-248.

Moh'd A Mesleh, A. 2007. Chi square feature extraction based svms arabic language text categorization system. Journal of Computer Science, 3(6): 430-435.

Motamedi, R., Gonzalez, R., Farahbakhsh, R., Cuevas, A., Cuevas, R., and Rejaie, R. 2013. What osn should i use? characterizing user engagement in major osns. [Online]. Available: http://www.it.uc3m.es/~rgonza1/pubs/whatOSN.pdf.

Nadeau, D., and Sekine, S. 2007. A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1): 3-26.

Narayanan, A., and Shmatikov, V. 2009. De-anonymizing social networks. IEEE 30th symposium on security and privacy, Oakland, California, USA, 173-187.

Navarro, G. 2001. A guided tour to approximate string matching. ACM computing surveys, 33(1): 31-88.

Nepali, R. K., and Wang, Y. 2013. Sonet: A social network model for privacy monitoring and ranking. IEEE 33rd International Conference on Distributed Computing Systems Workshops, Philadelphia, USA, 162-166.

Nguyen, H. M., Cooper, E. W., and Kamei, K. 2011. Borderline over-sampling for imbalanced data classification. International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1): 4-21.

Nick, S., 2018. Maker of popular quiz apps on Facebook exposed data of 120 million users. [Online]. Available: https://www.theverge.com/2018/6/28/17514822/facebook-data-leak-quiz-app-nametests-social-sweetheart-exposed-user-info.

Nicvist, S., Bogatireva, D., and Bobicev, V. 2018. Tweet Author Gender Identification. PAN 2016 Task. The 6th International Conference Telecommunications Electronics and Informatics, Chisinau, Moldova, 344-347.

Novak, P. K., Smailović, J., Sluban, B., and Mozetič, I. 2015. Sentiment of emojis. PloS one, 10(12): 1-22.

Nuutila, E., and Soisalon-Soininen, E., 1994. On finding the strongly connected components in a directed graph. Information Processing Letters, 49(1): 9-14.

Okur, E., Demir, H., and Özgür, A. 2018. Named entity recognition on twitter for turkish using semi-supervised learning with word embeddings. arXiv preprint arXiv:1810.08732.

Ortigosa, A., Martín, J. M., and Carro, R. M. 2014. Sentiment analysis in Facebook and its application to e-learning. Computers in human behavior, 31: 527-541.

Oukemeni, S., Rifà Pous, H., and Marquès Puig, J. M. 2018. Privacy in microblogging online social networks: issues and metrics. Proceedings of the Spanish Meeting on Cryptology and Information Security (RECSI), Granada, Spain, 107-112.

Oukemeni, S., Rifà-Pous, H., and Puig, J. M. M. 2019. IPAM: Information Privacy Assessment Metric in Microblogging Online Social Networks. IEEE Access, 7: 114817-114836.

Özkaya, S., and Diri, B. 2011. Named entity recognition by conditional random fields from Turkish informal texts. IEEE 19th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 662-665.

Pamay, T., Sulubacak, U., Torunoğlu-Selamet, D., and Eryiğit, G. 2015. The annotation process of the ITU Web Treebank. Proceedings of the 9th Linguistic Annotation Workshop, Colorado, USA, 95-101.

Passaro, L. C., Bondielli, A., and Lenci, A. 2016. Fb-news15: A topic-annotated facebook corpus for emotion detection and sentiment analysis. Proceedings of the Third Italian Conference on Computational Linguistics, Napoli, Italy, 228-232.

Pathak, M. J., Patel, R. L., and Rami, S. P., 2018. Comparative Analysis of Search Algorithms. International Journal of Computer Applications, 179(50): 40-43.

Patil, R., and Temkar, R., 2017. Intelligent Testing Tool: Selenium Web Driver. International Research Journal of Engineering and Technology, 4(6): 1920-1923.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12: 2825-2830.

Peersman, C., Daelemans, W., and Van Vaerenbergh, L. 2011. Predicting age and gender in online social networks. Proceedings of the 3rd international workshop on Search and mining user-generated contents, Glasgow, UK, 37-44.

Peled, O., Fire, M., Rokach, L., and Elovici, Y. 2013. Entity matching in online social networks. IEEE International Conference on Social Computing, Atlanta, Georgia, USA, 339-344.

Peng, H., Long, F., and Ding, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence, 27:8, 1226-1238.

Pensa, R. G., and Di Blasi, G. 2016a. A semi-supervised approach to measuring user privacy in online social networks. International Conference on Discovery Science, Bari, Italy, 392-407.

Pensa, R. G., and Di Blasi, G. 2016b. A centrality-based measure of user privacy in online social networks. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, California, USA, 1438-1439.

Pensa, R. G., and Di Blasi, G. 2017. A privacy self-assessment framework for online social networks. Expert Systems with Applications, 86: 18-31.

Pensa, R. G., Di Blasi, G., and Bioglio, L. 2019. Network-aware privacy risk estimation in online social networks. Social Network Analysis and Mining, 9: 1-15.

Petkos, G., Papadopoulos, S., and Kompatsiaris, Y. 2015. PScore: a framework for enhancing privacy awareness in online social networks. IEEE 10th International Conference on Availability, Reliability and Security, Toulouse, France, 592-600.

Ponnappa, R. 2019. Google Colab: Using GPU for Deep Learning. [Online]. Available: https://python.gotrained.com/google-colab-gpu-deep-learning/

Pool, C., and Nissim, M. 2016. Distant supervision for emotion detection using Facebook reactions. arXiv preprint arXiv:1611.02988.

Quan-Haase, A., and Young, A. L., 2010. Uses and gratifications of social media: A comparison of Facebook and instant messaging. Bulletin of Science, Technology & Society, 30(5): 350-361.

Raad, E., Chbeir, R., and Dipanda, A. 2010. User profile matching in social networks. IEEE 13th International Conference on Network-Based Information Systems, Takayama, Japan, 297-304.

Rahim, R., Abdullah, D., Nurarif, S., Ramadhan, M., Anwar, B., Dahria, M., and Khairani, M., 2018. Breadth First Search Approach for Shortest Path Solution in Cartesian Area. In Journal of Physics: Conference Series, 1019(2018).

Rangel, F., and Rosso, P. 2013. On the identification of emotions and authors' gender in facebook comments on the basis of their writing style. Proceedings of the First International Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM), Turin, Italy, 34-46.

Rehurek, R., and Sojka, P. 2010. Software framework for topic modelling with large corpora. Proceedings of the LREC Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 46-50.

Reif, J. H., 1985. Depth-first search is inherently sequential. Information Processing Letters, 20(5): 229-234.

Rodrigues, G. P. W., Costa, L. H. M., Farias, G. M., and de Castro, M. A. H., 2019. A Depth-First Search Algorithm for Optimizing the Gravity Pipe Networks Layout. Water Resources Management, 33(13): 4583-4598.

Rodrıguez, S. S., Redondo, R. P. D., Vilas, A. F., and Arias, J. J. P. 2012. Using Facebook activity to infer social ties. Proceedings of the 2nd International Conference on Cloud Computing and Services Science, Porto, Portugal, 325-333.

Rysbek, D. 2019. Sentiment Analysis with Recurrent Neural Networks on Turkish Reviews Domain. PhD Thesis. Middle East Technical University.

Sahin, H. B., Tirkaz, C., Yildiz, E., Eren, M. T., and Sonmez, O. 2017. Automatically annotated Turkish corpus for named entity recognition and text categorization using large-scale gazetteers. arXiv preprint arXiv:1702.02363.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5): 513-523.

Salur, M. U., Aydin, İ., and Alghrsi, S. A. 2019. SmartSenti: A Twitter-Based Sentiment Analysis System for the Smart Tourism in Turkey. IEEE International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 1-5.

Sang, E. F., and De Meulder, F. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.

Santur, Y. 2019. Sentiment Analysis Based on Gated Recurrent Unit. IEEE International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 1-5.

Sap, M., Park, G., Eichstaedt, J., Kern, M., Stillwell, D., Kosinski, M., and Schwartz, H. A. 2014. Developing age and gender predictive lexica over social media. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 1146-1151.

Sarhan, F., 2017. Archived Github Repo: ButunUniversiteListesiCrawler. [Online] Available: https://github.com/f9n/ButunUniversiteListesiCrawler/blob/master/universities.txt

Say, B., Zeyrek, D., Oflazer, K., and Özge, U. 2002. Development of a corpus and a treebank for present-day written Turkish. In Proceedings of the eleventh international conference of Turkish linguistics, Gazimağusa, Northern Cyprus, 183-192.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., and Ungar, L. H. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. PloS one, 8(9): e73791.

Seo, Y. D., Kim, Y. G., Lee, E., and Baik, D. K. 2017. Personalized recommender system based on friendship strength in social network services. Expert Systems with Applications, 69: 135-148.

Seok, M., Song, H. J., Park, C. Y., Kim, J. D., and Kim, Y. S. 2016. Named entity recognition using word embedding as a feature. International Journal of Software Engineering and Its Applications, 10(2): 93-104.

Sezerer, E., Polatbilek, O., and Tekir, S. 2019a. Gender Prediction from Turkish Tweets with Neural Networks. IEEE 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 1-4.

Sezerer, E., Polatbilek, O., and Tekir, S. 2019b. A Turkish Dataset for Gender Identification of Twitter Users. Proceedings of the 13th Linguistic Annotation Workshop, Florence, Italy, 203-207.

Sha, Y., Liang, Q., and Zheng, K. 2016. Matching user accounts across social networks based on users message. Procedia Computer Science, 80: 2423-2427.

Shao, M., Xia, S., and Fu, Y. 2014. Identity and kinship relations in group pictures (F. Yun editor). Human-centered social media analytics, first edition, Springer International Publishing, Switzerland, 175-190.

Shi, N., Lee, M. K., Cheung, C. M., and Chen, H., 2010. The continuance of online social networks: how to keep people using Facebook?. IEEE 43rd Hawaii International Conference on System Sciences, Kauai, Hawaii, 1-10.

Siddula, M., Li, L., and Li, Y. 2018. An empirical study on the privacy preservation of online social networks. IEEE Access, 6: 19912-19922.

Siwag, T., Sirohi, P., and Singhal, N. 2014. Novel Architecture of a focused crawler for social websites. International Journal of Computer Engineering and Applications, 7(3): 132-144.

Snedecor, G. W., and Cochran, W. G. 1989. Statistical Methods. Iowa state University press, Ames, Iowa, 491.

Song, X., Wang, X., Nie, L., He, X., Chen, Z., and Liu, W. 2018. A personal privacy preserving framework: I let you know who can see what. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, USA, 295-304.

Sosa, P. M. 2017. Twitter sentiment analysis using combined lstm-cnn models. Eprint Arxiv.

Spiliotopoulos, T., and Oakley, I. 2013. Understanding motivations for Facebook use: Usage metrics, network structure, and privacy. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 3287-3296.

Sramka, M. 2015. Evaluating Privacy Risks in Social Networks from the User's Perspective (G. Navarro-Arribas, V. Torra). Advanced Research in Data Privacy, Springer, Cham. 251-267.

Srivastava, A., and Geethakumari, G. 2013. Measuring privacy leaks in online social networks. IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 2095-2100.

Srivastava, A., and Geethakumari, G. 2014. A privacy settings recommender system for online social networks. IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 1-6.

Srivastava, A., and Geethakumari, G. 2015. Privacy landscape in online social networks. International Journal of Trust Management in Computing and Communications, 3(1): 19-39.

Stojanovski, D., Strezoski, G., Madjarov, G., and Dimitrovski, I. 2016. Finki at semeval-2016 task 4: Deep learning architecture for twitter sentiment analysis. Proceedings of the 10th International workshop on semantic evaluation, San Diego, California, USA, 149-154.

Su, J., Shirab, J. S., and Matwin, S. 2011. Large scale text classification using semi-supervised multinomial naive bayes. Proceedings of the 28th international conference on machine learning, Bellevue, Washington, USA, 97-104.

Sundheim, B. 1995. MUC-6 Named Entity Task Definition (v2. 1). In Proceedings of the Sixth Message Understanding Conference, Columbia, Maryland, USA, 317-332.

Sutton, C., and McCallum, A. 2012. An introduction to conditional random fields. Foundations and Trends in Machine Learning, 4(4): 267-373.

Süerdem, A. K., and Kaya, E. 2015. Using Sentiment Analysis to Detect Customer Attitudes in Social Media Comments. Research in Computing Science, 90: 207-215.

Symeonidis, I., Beato, F., Tsormpatzoudi, P., and Preneel, B. 2015. Collateral damage of Facebook Apps: an enhanced privacy scoring model. IACR Cryptology ePrint Archive, 2015: 456-473.

Şeker, G. A., and Eryiğit, G. 2012. Initial explorations on using CRFs for Turkish named entity recognition. Proceedings of the 24th International Conference on Computational Linguistics (COLING), Mumbai, India, 2459-2474.

Şeker, G. A., and Eryigit, G. 2017. Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content. Semantic Web, 8(5): 625-642.

Talebi, M., and Köse, C. 2013. Identifying gender, age and education level by analyzing comments on Facebook. IEEE 21st Signal Processing and Communications Applications Conference (SIU), Haspolat, North Cyprus, 1-4.

Talukder, N., Ouzzani, M., Elmagarmid, A. K., Elmeleegy, H., and Yakout, M. 2010. Privometer: Privacy protection in social networks. IEEE 26th International Conference on Data Engineering Workshops (ICDEW), Long Beach, California, USA, 266-269.

Tang, C., Ross, K., Saxena, N., and Chen, R. 2011. What's in a name: A study of names, gender inference, and gender behavior in facebook. International Conference on Database Systems for Advanced Applications, Busan, Korea, 344-356.

Tarjan, R.,1972. Depth-first search and linear graph algorithms. SIAM journal on computing, 1(2): 146-160.

Tatar, S., and Cicekli, I. 2011. Automatic rule learning exploiting morphological features for named entity recognition in Turkish. Journal of Information Science, 37(2): 137-151.

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., and Belopolsky, A., 2016. Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.

Tellez, E. S., Miranda-Jiménez, S., Moctezuma, D., Graff, M., Salgado, V., and Ortiz-Bejar, J. 2018. Gender identification through multi-modal tweet analysis using microtc and bag of visual words. Proceedings of the Ninth International Conference of the CLEF Association, Avignon, France.

Terrana, D., Augello, A., and Pilato, G. 2014. Facebook users relationships analysis based on sentiment classification. IEEE International Conference on Semantic Computing, California, USA, 290-296.

Terrana, D., Augello, A., and Pilato, G. 2015. A system for analysis and comparison of social network profiles. Proceedings of the IEEE 9th International Conference on Semantic Computing (ICSC), Anaheim, California, USA, 109-115.

Tezgider, M., Yıldız, B., and Aydın, G. 2018. Improving Word Representation by Tuning Word2Vec Parameters with Deep Learning Model. IEEE International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 1-7.

Thabtah, F., Eljinini, M., Zamzeer, M., and Hadi, W. 2009. Naïve Bayesian based on Chi Square to categorize Arabic data. Proceedings of the 11th International Business Information Management Association Conference (IBIMA) on Innovation and Knowledge Management in Twin Track Economies, Cairo, Egypt, 4-6.

Thomas, K., Grier, C., and Nicol, D. M. 2010. unfriendly: Multi-party privacy risks in social networks. International Symposium on Privacy Enhancing Technologies Symposium, Berlin, Germany, 236-252.

Tian, Y., Galery, T., Dulcinati, G., Molimpakis, E., and Sun, C. 2017. Facebook sentiment: Reactions and emojis. Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Valencia, Spain, 11-16.

Tomek, I. 2010. Two modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, 6(11): 769-772.

Trinh, S., Nguyen, L., Vo, M., and Do, P. 2016. Lexicon-based sentiment analysis of Facebook comments in Vietnamese language (D. Król, L. Madeyski, N Nguyen editors). Recent developments in intelligent information and database systems, Springer, Cham, 263-276.

Troussas, C., Virvou, M., Espinosa, K. J., Llaguno, K., and Caro, J. 2013. Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning. IEEE Fourth International Conference on Information, Intelligence, Systems and Applications (IISA), Piraeus, Greece, 1-6.

Tuna, T., Akbas, E., Aksoy, A., Canbaz, M. A., Karabiyik, U., Gonen, B., and Aygun, R. 2016. User characterization for online social networks. Social Network Analysis and Mining, 6(1): 104.

Turan, M. 2012. Kişisel Verileri Koruma Kurumu. Türkiye Kalkınma Bankası Yayını, 63: 1-4.

Tur, G., Hakkani-Tur, D., and Oflazer, K. 2003. A statistical information extraction system for Turkish. Natural Language Engineering, 9(2): 181-210.

Umair, A., Nanda, P., He, X., and Choo, K. K. R. 2018. User relationship classification of Facebook messenger mobile data using weka. International Conference on Network and System Security, Hong Kong, China, 337-348.

Uysal, A. K., and Murphey, Y. L. 2017. Sentiment classification: Feature selection based approaches versus deep learning. IEEE International Conference on Computer and Information Technology (CIT), Helsinki, Finland, 23-30.

Uzel, V. N., Eşsiz, E. S., and Özel, S. A. 2018. Using Fuzzy Sets for Detecting Cyber Terrorism and Extremism in the Text. IEEE Innovations in Intelligent Systems and Applications Conference (ASYU), Adana, Turkiye, 1-4.

Van Asch, V. 2013. Macro-and micro-averaged evaluation measures, CLiPS, Belgium, 49.

Vapnik, V. N. 1995. The Nature of Statistical Learning Theory. New York: Springer, Verlag, 188.

Vashisht, G., and Thakur, S. 2014. Facebook as a corpus for emoticons-based sentiment analysis. International Journal of Emerging Technology and Advanced Engineering, 4(5): 904-908.

Veiga, M. H., and Eickhoff, C. 2016. Privacy leakage through innocent content sharing in online social networks. arXiv preprint arXiv:1607.02714.

Vidyalakshmi, B. S., Wong, R. K., Ghanavati, M., and Chi, C. H. 2014. Privacy as a service in social network communications. IEEE International Conference on Services Computing, Anchorage, Alaska, USA, 456-463.

Vidyalakshmi, B. S., Wong, R. K., and Chi, C. H. 2015a. Privacy scoring of social network users as a service. IEEE International Conference on Services Computing, New York, USA, 218-225.

Vidyalakshmi, B. S., Wong, R. K., and Chi, C. H. 2015b. Privacy preserving information dispersal in social networks based on disposition to privacy. IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 372-377.

Villota, E. J., and Yoo, S. G. 2018. An experiment of influences of Facebook posts in other users. IEEE International Conference on eDemocracy & eGovernment (ICEDEG), Buenos Aires, Argentina, 83-88.

Viswanath, B., Mislove, A., Cha, M., and Gummadi, K. P. 2009. On the evolution of user interaction in facebook. Proceedings of the 2nd ACM workshop on Online social networks, Barcelona, Spain, 37-42.

Voigt, P., and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). A Practical Guide, Springer, Cham, 383.

Vosecky, J., Hong, D., and Shen, V. Y. 2009. User identification across multiple social networks. IEEE first international conference on networked digital technologies, Ostrava, Czech Republic, 360-365.

Wani, M. A., and Jabin, S. 2018. Mutual clustering coefficient-based suspicious-link detection approach for online social networks. Journal of King Saud University-Computer and Information Sciences.

Wani, M. A., Agarwal, N., Jabin, S., and Hussai, S. Z., 2018. Design and Implementation of iMacros-based Data Crawler for Behavioral Analysis of Facebook Users. arXiv preprint arXiv:1802.09566.

Wang, G., Gallagher, A., Luo, J., and Forsyth, D. 2010. Seeing people in social context: Recognizing people and social relationships. Proceedings of the European conference on computer vision (ECCV), Crete, Greece, 169-182.

Wang, Y., Liu, T., Tan, Q., Shi, J., and Guo, L. 2016. Identifying Users across Different Sites using Usernames. Proceedings of the International Conference on Computational Science (ICCS), San Diego, California, USA, 376-385.

Wang, Q., Xue, H., Li, F., Lee, D., and Luo, B. 2019. #DontTweetThis: Scoring Private Information in Social Networks. Proceedings on Privacy Enhancing Technologies, 2019(4): 72-92.

Wearesocial., 2019. Digital 2019 Turkey. [Online]. Available: https://wearesocial.com/global-digital-report-2019.

Wilson, C., Boe, B., Sala, A., Puttaswamy, K. P., and Zhao, B. Y. 2009. User interactions in social networks and their implications. Proceedings of the 4th ACM European conference on Computer systems, Nuremberg, Germany, 205-218.

Wilson, C., Sala, A., Puttaswamy, K. P., and Zhao, B. Y., 2012. Beyond social graphs: User interactions in online social networks and their implications. ACM Transactions on the Web, 6(4): 1-31.

Wilson, D. L. 1972. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, (3): 408-421.

Witten, I. H., Frank, E., and Hall, M. A. 2005. Data Mining: Practical machine learning tools and techniques. Elsevier, San Francisco, USA, 578.

Wondracek, G., Holz, T., Kirda, E., and Kruegel, C. 2010. A practical attack to de-anonymize social network users. IEEE symposium on security and privacy, Oakland, California, USA, 223-238.

Wong, C. I., Wong, K. Y., Ng, K. W., Fan, W., and Yeung, K. H. 2014. Design of a crawler for online social networks analysis. WSEAS Transactions on Communications, 13: 263-274.

Xiang, R., Neville, J., and Rogati, M. 2010. Modeling relationship strength in online social networks. Proceedings of the 19th international conference on world wide web, Raleigh, North Carolina, USA, 981-990.

Xiao, Z., Liu, B., Hu, H., and Zhang, T. 2012. Design and implementation of facebook crawler based on interaction simulation. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, England, 1109-1112.

Xu, H., Tao, W., and Guo, F. 2019. A novel estimation model for user relationship intensity in social network. International Journal of Wireless and Mobile Computing, 16(3): 272-280.

Yang, B., Wen, D., Qin, L., Zhang, Y., Wang, X., and Lin, X., 2019. Fully dynamic depth-first search in directed graphs. Proceedings of the VLDB Endowment, 13(2): 142-154.

Yang, Y., Lutes, J., Li, F., Luo, B., and Liu, P. 2012. Stalking online: on user privacy in social networks. Proceedings of the second ACM conference on Data and Application Security and Privacy, San Antonio, Texas, USA, 37-48.

Yeniterzi, R. 2011. Exploiting morphology in Turkish named entity recognition system. Proceedings of the ACL Student Session, Portland, Oregon, USA, 105-110.

Yeniterzi, R., Tür, G., and Oflazer, K. 2018. Turkish Named-Entity Recognition (M. Saraçlar editor). Turkish Natural Language Processing, Springer, Cham, 115-132.

Ypma, T. J. 1995. Historical development of the Newton–Raphson method. SIAM review, 37(4): 531-551.

Zafarani, R., Abbasi, M. A., and Liu, H., 2014. Social media mining: an introduction. Cambridge University Press, UK, 380.

Zafarani, R., and Liu, H. 2013. Connecting users across social media sites: a behavioral-modeling approach. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, Chicago, Illinois, USA, 41-49.

Zamani, N. A. M., Abidin, S. Z., Omar, N, and Abiden, M. Z. Z. 2014. Sentiment analysis: determining people's emotions in Facebook. Proceedings of the 13th international conference on applied computer and applied computational science, Kuala Lumpur, Malaysia, 111-116.

Zeng, Y., Sun, Y., Xing, L., and Vokkarane, V. 2015. A study of online social network privacy via the tape framework. IEEE Journal of Selected Topics in Signal Processing, 9(7): 1270-1284.

Zhao, X., Yuan, J., Li, G., Chen, X., and Li, Z. 2012. Relationship strength estimation for online social networks with the study on Facebook. Neurocomputing, 95: 89-97.

Zheleva, E., and Getoor, L. 2011. Privacy in social networks: A survey (C. Aggarwal editor). Social network data analytics, Springer, Boston, MA, 277-306.

Zhang, C., Sun, J., Zhu, X., and Fang, Y., 2010. Privacy and security for online social networks: challenges and opportunities. IEEE network, 24(4): 13-18.

Zhang, Y., Rahman, M. M., Braylan, A., Dang, B., Chang, H. L., Kim, H., and McDonnell, T. 2016. Neural information retrieval: A literature review. arXiv preprint arXiv:1611.06792.

Zhang, L., Wang, S., and Liu, B. 2018. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4): e1253.

Zhang, J., and Philip, S. Y. 2019. Broad Learning Through Fusions An Application on Social Networks, Springer International Publishing, 419.

Zhou, X., Liang, X., Zhang, H., and Ma, Y. 2015. Cross-platform identification of anonymous identical users in multiple social media networks. IEEE transactions on knowledge and data engineering, 28(2): 411-424.

CURRICULUM VITAE

Önder ÇOBAN was born in Erzurum. He received a BSc degree from Süleyman Demirel University, Isparta in 2013 and an MSc degree from Atatürk University, Erzurum in 2016. Since 2014, he has been working as a research assistant and is currently working at Adıyaman University.

APPENDICES

APPENDIX A: Statistical Description of Three Snapshots of the Crawled Data

Table A.1. Descriptive properties of directed (DIR) and undirected (UND) graphs of crawled public Facebook data

Attribute	Snapshot (S)					
	5K		10K		20K	
Graph type	DIR	UND	UND	DIR	DIR	UND
Nodes	5,000	5,000	10,000	10,000	20,000	20,000
Edges	59,754	45,449	113,489	88,159	256,979	201,151
Avg. degree	11.951	18.180	11.349	17.632	12.849	20.115
Network diameter	6	5	7	5	7	6
Graph density	0.002	0.004	0.001	0.002	0.001	0.001
Avg. clustering coefficient	0.422	0.520	0.387	0.475	0.376	0.460
Avg. path length	3.522	3.536	3.678	3.661	3.784	3.751
# of triangles	101,497	198,345	148,823	315,842	384,952	886,674

Table A.2. User interactions in crawled data

Attribute	Snapshot (S)		
	5K	10K	20K
# of visited user	5,000	10,000	20,000
# of users whose first neighborhoods are visited	76	139	261
# of discovered friends	1,109,383	2,003,085	3,980,270
# of unique friends	734,971	1,263,148	2,350,454
# of filtered users	756	959	1079
# of posts	794,873	1,408,496	2,762,023
# of comments	933,187	1,589,310	2,743,513
# of replies	144,342	262,362	466,995
# of discovered kinships	2,082	3,808	7,581
# of discovered private relationships	839	2,136	4,660
Crawling period	20/12/18-28/02/19		

Table A.3. Graph size by hop count for the largest snapshot of the crawled data

# of users	Hop count				
(nodes)	1	2	3	4	5
Unique	6	260	19,843	2,243,043	2,350,454
Total	6	268	22,549	3,701,567	3,980,270

Table A.4. Links that follow from symmetry (for friendship links)

Links (Edges)	Snapshot		
	5K	10K	20K
A → B / B → A (Unidirectional)	31,142	62,827	145,321
A ↔ B (Bidirectional / Symmetric)	14,306	25,331	55,829
Discovered unidirectional	59,754	113,489	256,979
Unidirectional known to be present (i.e., implicit links)	90,896	176,316	402,300
Percent of inferred (B → A)	34.26	35.63	36.12

Table A.5. SCC/WCC analysis in three snapshots for friendship links

Property (# of)	Snapshot		
	5K	10K	20K
Edges	59,754	113,489	256,979
Nodes in LSCC	2,336	4,242	8,283
Singleton SCCs (WCCs)	2,664	5,758	3,279
Edges in LSCC	30,389	53,458	117,340
Nodes in WCE	2,664	5,758	11,717
Edges in WCE	29,365	60,001	139,610

Table A.6. Distribution of revealed interest, religious view, and gender attributes in three snapshots (S) of the crawled data

S	User Interest			Religious View		Gender		
	Males	Females	Males and Females	Islam	Other	Male	Female	NA
5K	28	97	72	141	6	2,694	1,506	798
10K	51	194	140	266	14	5,067	3,105	1,827
20K	108	415	286	472	40	9,800	6,486	3,714

Table A.7. The number of users in three snapshots (S) who make their wall and friends pages public, private, and partially public

S	Wall (Timeline)		Friends		
	Public	Private	Public	Private	Partially Public
5K	4,505	495	2,153	2,667	180
10K	9,077	923	4,074	5,745	181
20K	18,579	1,421	8,109	11,675	216

Table A.8. Top 10 languages that are most frequently disclosed by users

S	Disclosed Languages									
	TR (Turkish)	EN (English)	G (German)	AR (Arabic)	RU (Russian)	FR (French)	SP (Spanish)	KR (Kurdish)	ZA (Zaza)	AZ (Azeri)
5K	235	289	73	26	18	19	8	23	34	15
10K	430	485	118	54	30	39	13	46	55	22
20K	919	1,069	267	99	67	85	31	95	87	45

Table A.9. Distribution of disclosed private relationships in snapshots (S)

S	Private Relationships					Total
	Married	In a Relation	Single	Engaged	Other	
5K	702	25	91	16	3	837
10K	1,734	64	292	38	8	2,136
20K	3,689	149	740	69	13	4,660

Table A.10. Total number of reactions for posts and comments on users' wall pages in different snapshots (S)

S	Facebook Reactions					
	Love	Haha	Wow	Sad	Angry	Like
5K	130,307	18,201	5,463	76,007	14,305	13,904,484
10K	268,769	38,558	10,150	136,489	24,940	25,123,756
20K	534,228	96,755	21,610	245,607	49,995	42,110,090

Table A.11. The number of users at different age-ranges by gender who revealed their date of birth

Gender	Age Range						
	[0-10]	[11-20]	[21-30]	[31-40]	[41-50]	[51-60]	[61+]
				5K			
Male	0	4	27	60	29	10	3
Female	0	2	37	36	9	4	1
NA	0	0	3	15	3	0	1
				10K			
Male	0	7	55	109	54	19	5
Female	0	2	65	64	14	10	1
NA	0	0	12	27	6	1	2
				20K			
Male	0	13	107	216	92	38	12
Female	0	6	108	131	31	14	3
NA	0	1	30	58	10	5	4

Table A.12. The number of users by gender who disclose their working status

Gender	Working Status			
	Continues	Leaved	Unknown	NA
		5K		
Male	481	57	699	1,456
Female	160	18	276	1,052
NA	91	15	148	544
		10K		
Male	810	98	1,353	2,806
Female	299	29	523	2,254
NA	165	28	273	1,361
		20K		
Male	1,451	186	2,475	5,688
Female	606	77	1,081	4,722
NA	355	46	511	2,800

Table A.13. Distribution of crawled post types in three snapshots (S)

S	Post Types									
	GR (Greeting)	E (Event)	T (Text)	P (Photo)	V (Video)	L (Link)	M (Memory)	G (Game)	LO (Location)	O (Other)
5K	23K	4,7K	236K	256K	59K	56K	4,8K	7,4K	14K	108K
10K	41K	6,9K	414K	459K	112K	104K	11K	11K	27K	219K
20K	65K	13K	828K	838K	229K	208K	18K	19K	54K	484K

Table A.14. Average number of posts and friends of users by gender in three snapshots (S)

S	Average Post			Average Friends		
	Male	Female	NA	Male	Female	NA
5K	190.876	107.955	118.252	303.856	115.669	137.038
10K	169.055	112.157	111.465	286.576	120.139	97.409
20K	170.579	109.382	102.613	276.641	137.153	102.263

Table A.15. The number of revealed user links in snapshots (S)

S	User Links					
	Facebook	Instagram	Myspace	Linked-In	Twitter	Other
5K	336	55	2	2	23	102
10K	1,085	147	2	4	48	210
20K	2,836	295	9	9	105	509

Table A.16. The number of users in three snapshots (S) whose usernames are composed of completely numeric values or any string

S	Username	
	Completely Numeric	Any String
5K	947 (19%)	4,053 (81%)
10K	2,044 (20%)	7,956 (80%)
20K	3,943 (20%)	16,057 (80%)

Table A.17. The number of users who disclose at least one education information by different education levels in three snapshots (S)

S	Education Level				
	Primary School	Secondary School	High School	University	Other
5K	34	14	1,363	1,704	449
10K	63	30	2,529	3,163	811
20K	114	53	4.985	6,021	1,626

Table A.18. Total number of posts, comments, replies, and number of responses to the posts and comments in three snapshots (S)

S	Activity Type			Total # of Responses	
	Post	Comment	Reply	Commented Posts	Replied Comments
5K	771,165	892,764	136,143	188,899	98,272
10K	1,408,496	1,589,310	262,362	339,315	188,289
20K	2,762,023	2,743,513	466,995	591,116	343,017

Table A.19. The most frequently disclosed eight cities among users in terms of hometown and place of lived-in for each snapshot (S)

Type	City							
	EL (Elazığ)	IS (İstabul)	BR (Bursa)	ANK (Ankara)	IZ (İzmir)	TU (Tunceli)	MA (Malatya)	KO (Konya)
				5K				
Lived-in	648	700	320	168	137	5	35	83
Hometown	801	174	164	68	88	7	72	122
				10K				
Lived-in	1,336	1,146	380	336	253	240	107	132
Hometown	1,622	263	191	122	131	290	133	179
				20K				
Lived-in	2,710	2,570	703	630	581	251	173	154
Hometown	3,304	676	400	258	355	349	229	223

Table A.20. The number of users with respect to the current graduation level and gender attributes.

Gender	Graduation Level					
	University	High School	Secondary School	Primary School	Other	NA
			5K			
Male	752	277	3	12	264	1,393
Female	304	96	2	3	83	1,012
NA	189	36	1	0	33	538
			10K			
Male	1,377	502	9	18	449	2,712
Female	626	231	3	5	174	2,066
NA	324	71	1	2	66	1,363
			20K			
Male	2,507	1,022	15	31	872	5,353
Female	1,224	494	4	10	371	4,383
NA	653	155	1	4	157	2,742

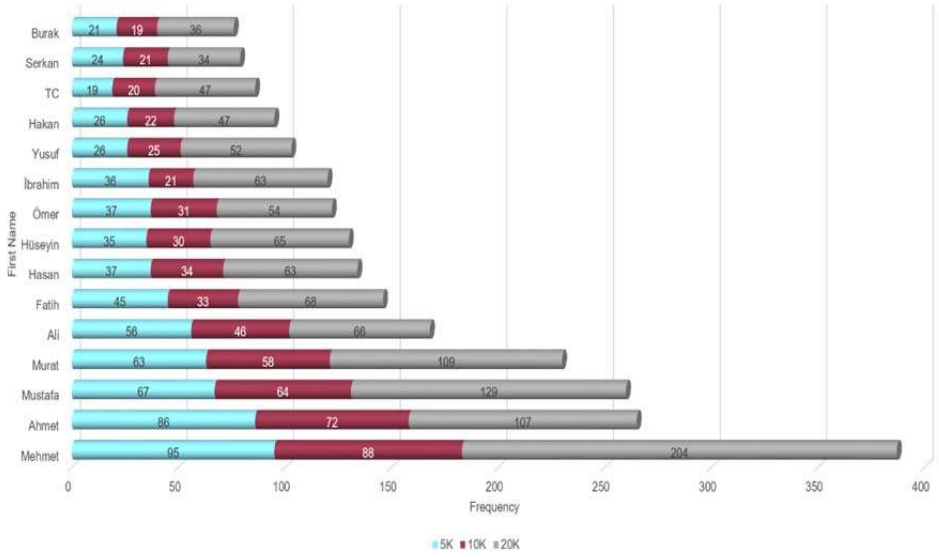


Figure A.1. Stacked frequencies of most frequent fifteen first names among male users in three snapshots

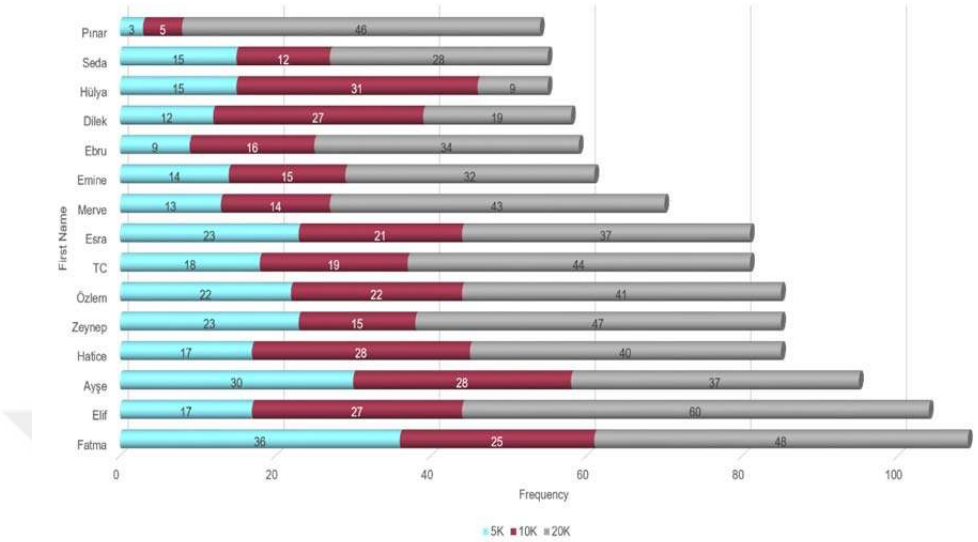


Figure A.2. Stacked frequencies of most frequent fifteen first names among female users in three snapshots

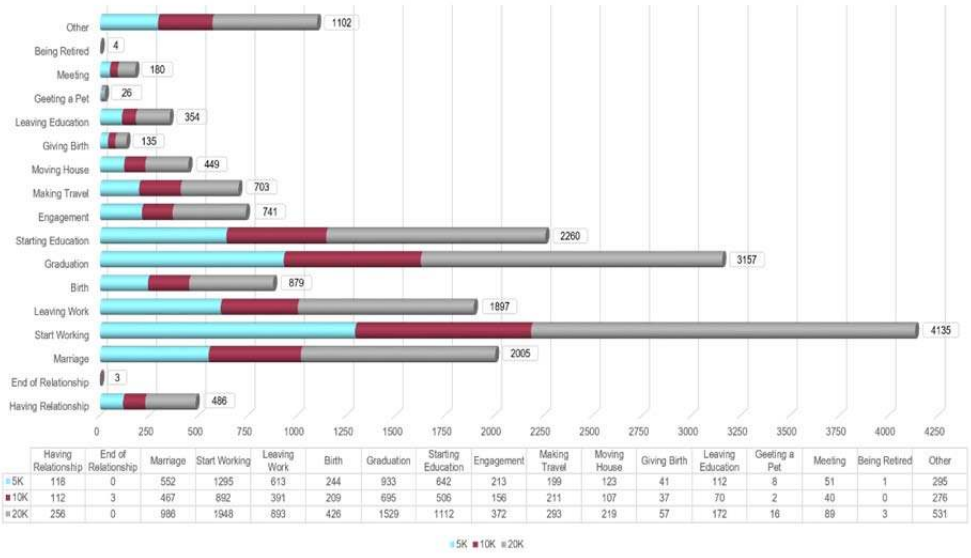


Figure A.3. Stacked distribution of the number of disclosed user events in three snapshots

Table A.21. The number of all posts by months in three snapshots (S)

S	Month											
	JAN (January)	FEB (February)	MAR (March)	APR (April)	MAY (May)	JUN (June)	JUL (July)	AUG (August)	SEP (September)	OCT (October)	NOV (November)	DEC (December)
5K	70,631	66,834	67,501	59,724	62,340	60,824	66,314	63,565	58,555	60,187	59,687	73,851
10K	123,600	115,533	125,073	112,064	121,232	119,607	122,083	120,121	111,609	107,102	104,002	125,253
20K	247,778	227,898	237,700	217,332	235,102	232,576	236,763	232,320	217,355	213,543	215,074	247,213

Table A.22. The number of all posts by years in three snapshots (S)

S	Year												
	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
5K	347	613	4,243	22,163	35,372	59,514	78,259	106,794	126,665	155,283	139,626	39,379	NA
10K	503	1,135	7,283	40,269	64,541	101,453	137,257	185,485	218,035	269,397	246,686	132,381	NA
20K	1,043	2,800	16,812	86,359	131,102	191,038	254,569	342,571	398,306	490,047	449,010	353,645	38,434

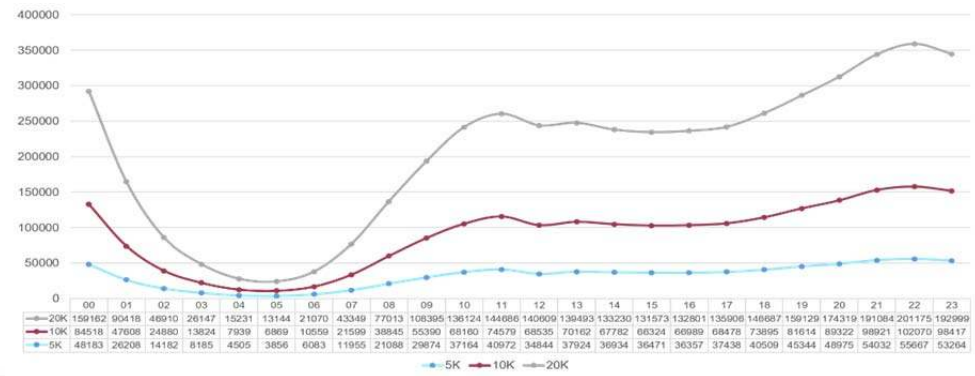


Figure A.4. Distribution of the number of all posts in three snapshots (S) by hours

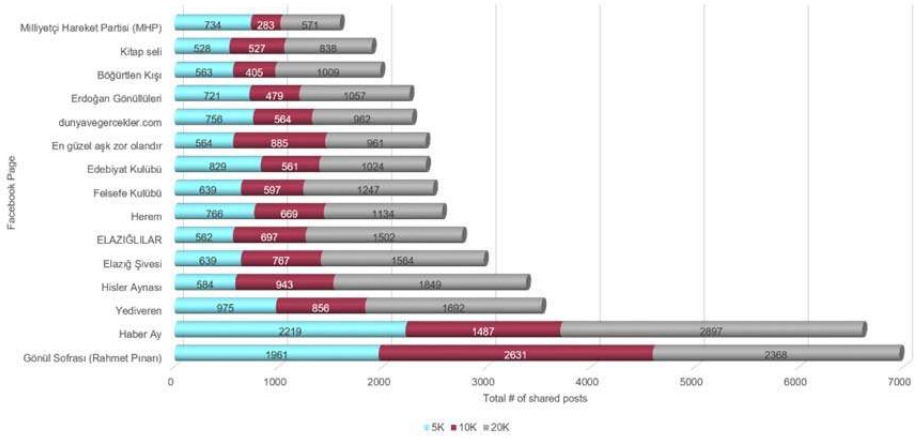
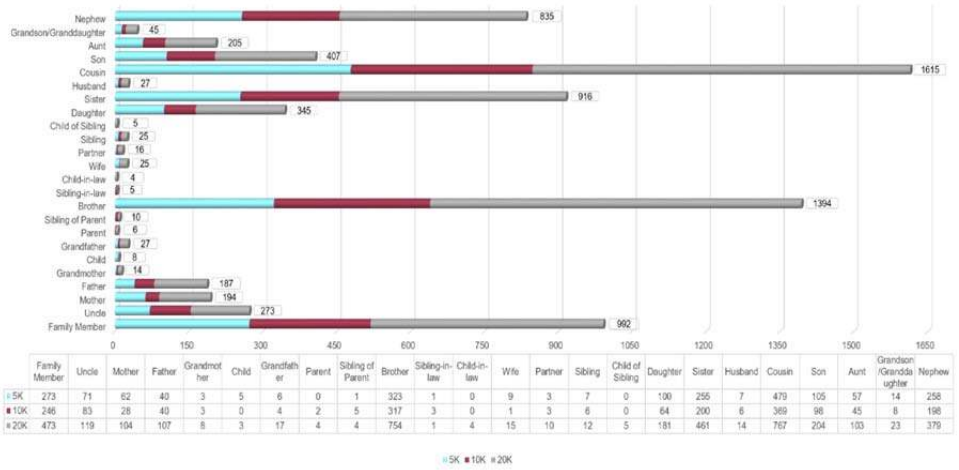


Figure A.5. Stacked distribution of the number of shared posts from top fifteen Facebook pages in three snapshots



	Family Member	Uncle	Mother	Father	Grandmother	Child	Grandfather	Parent	Sibling of Parent	Brother	Sibling-in-law	Child-in-law	Wife	Partner	Sibling	Child of Sibling	Daughter	Sister	Husband	Cousin	Son	Aunt	Grandson/Granddaughter	Nephew
5K	273	71	62	40	3	5	6	0	1	323	1	0	9	3	7	0	100	255	7	479	105	57	14	258
10K	246	83	28	40	3	0	4	2	5	317	3	0	1	3	6	0	64	200	6	369	98	45	8	198
20K	473	119	104	107	8	3	17	4	4	754	1	4	15	10	12	5	181	461	14	767	204	103	23	379

Figure A.6. Stacked distribution of the number of kinship types disclosed by users in three snapshots

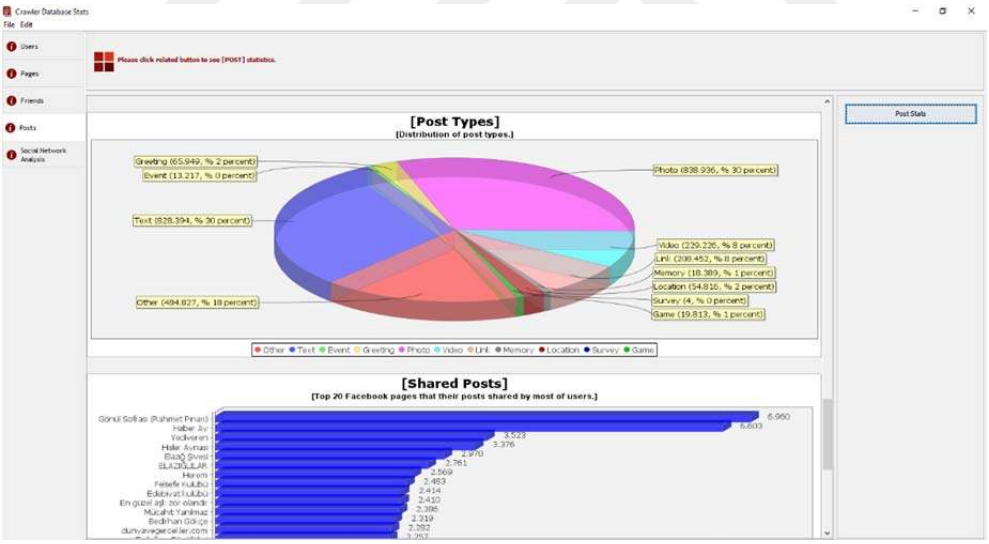


Figure A.7. Screenshot of our tool that is developed to extract statistical properties of public OSN data.

APPENDIX B: Developed NER Tools

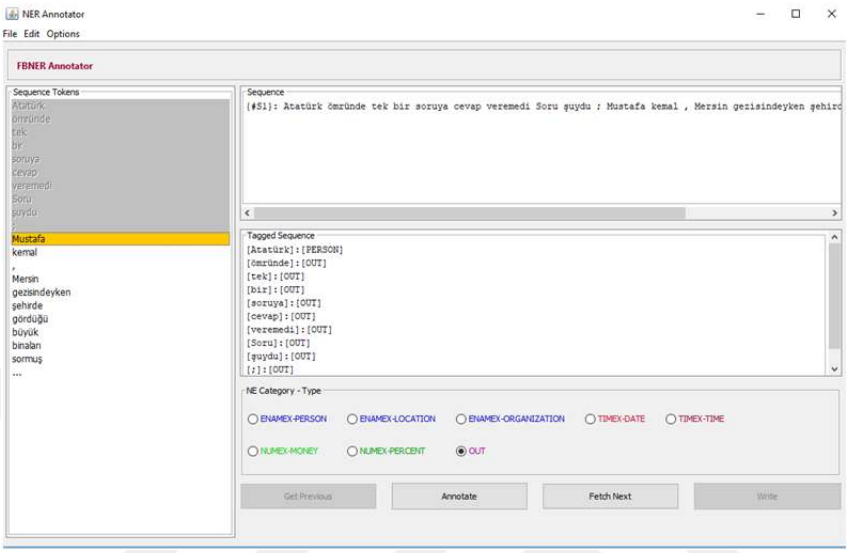


Figure B.1. Annotation tool

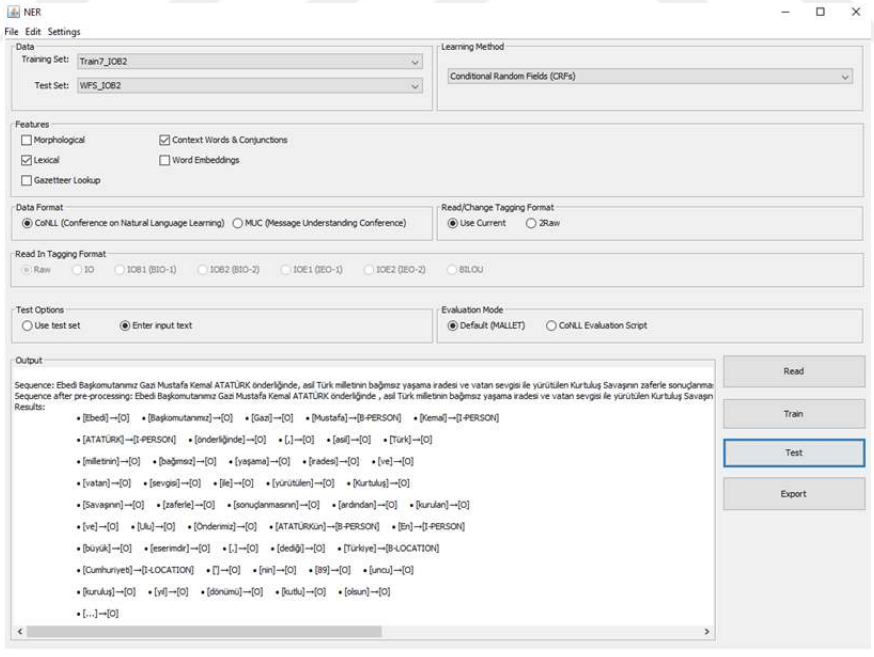


Figure B.2. CRF-based NER system

Table B.1. True predictions of a NER system in terms of MUC and CoNLL

Sequence		Predicted Type	MUC		CoNLL (TYPE + TEXT)
Token	Type		TYPE	TEXT	
Robert	PER	O	✗	✗	✗
and	O	LOC	✗	✗	✗
John	PER	ORG			
Briggs	PER	ORG	✓	✓	
Jr	PER	ORG			
contacted	O	O			
Wonderful	ORG	O			
Stockbrokers	ORG	ORG	✓	✓	
Inc	ORG	O			
in	O	PER			
New	LOC	PER	✓	✓	
York	LOC	PER			
to	O	O			
sell	O	O			
their	O	O			
shares	O	O			
in	O	O			
Acme	ORG	ORG	✓	✓	✓

APPENDIX C: HTML Source of Facebook

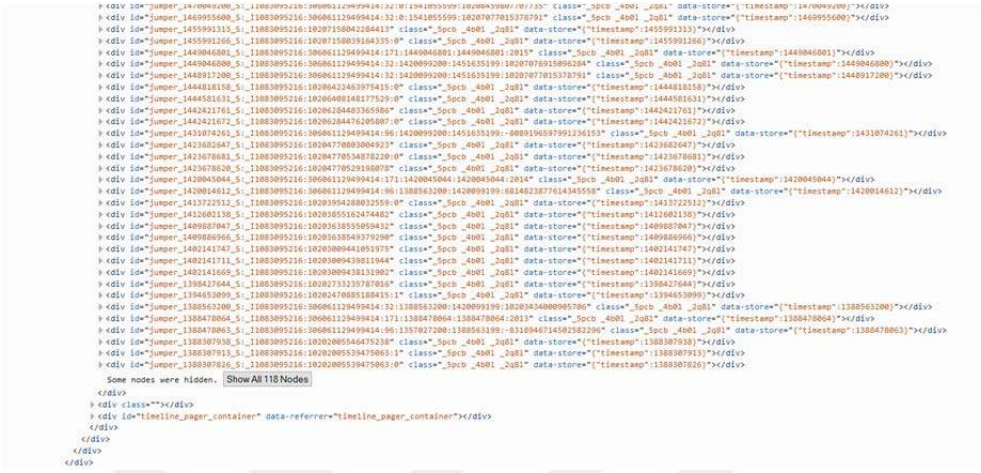


Figure C.1. HTML source of an arbitrary user's profile page before a regular update.

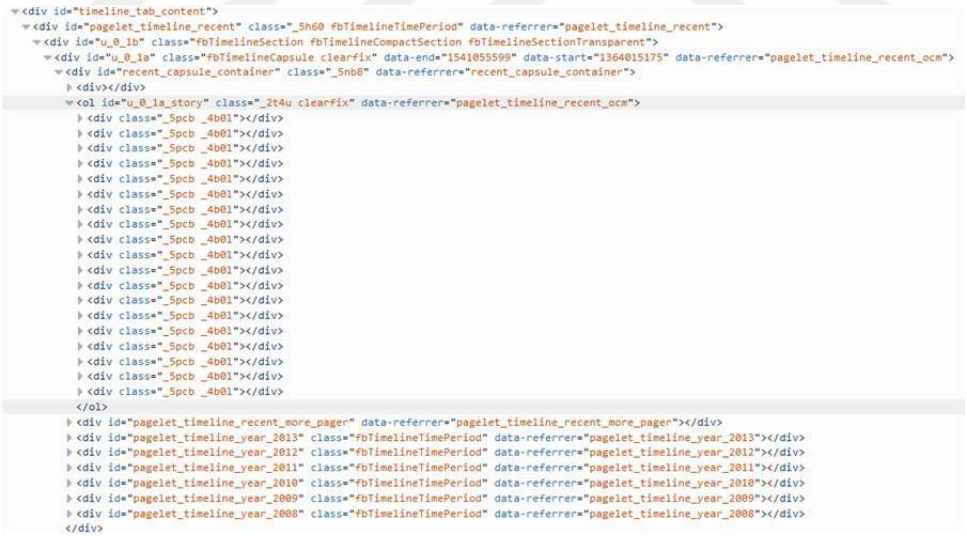


Figure C.2. HTML source of an arbitrary user's profile page after a regular update.

APPENDIX D: Designed Database

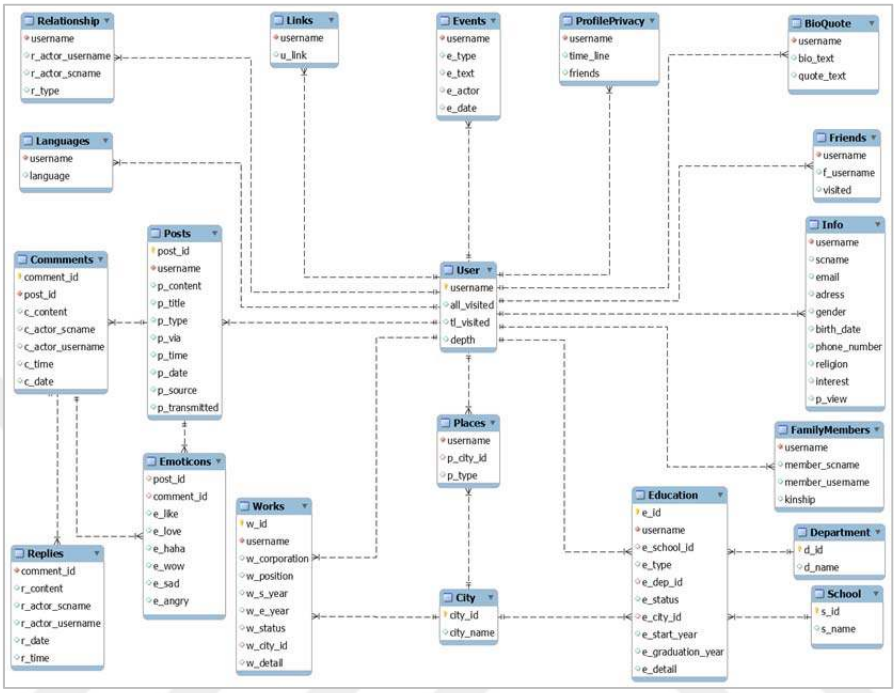


Figure D.1. Star schema of the designed relational database

APPENDIX E: Structures of DL algorithms used in our SA task

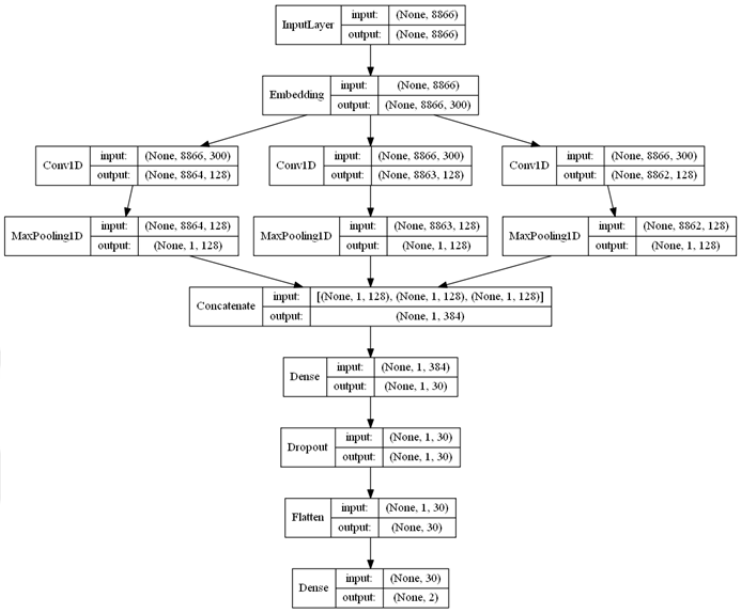


Figure E.1. Summary of CNN structure

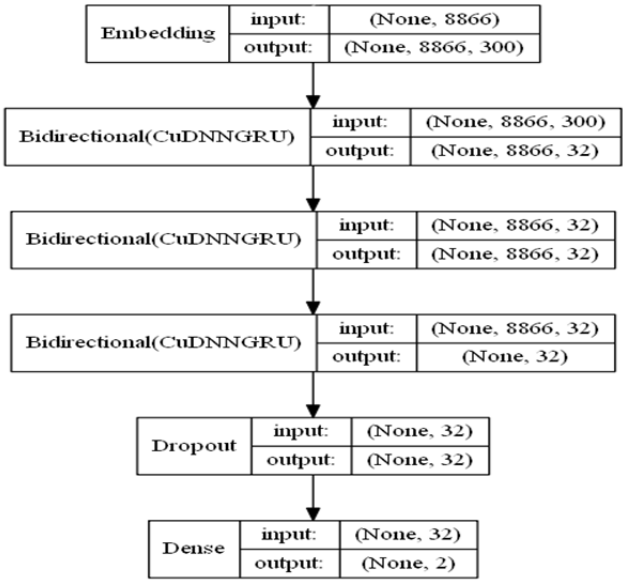


Figure E.2. Summary of our optimum stacked BRNN structure with GRU cells

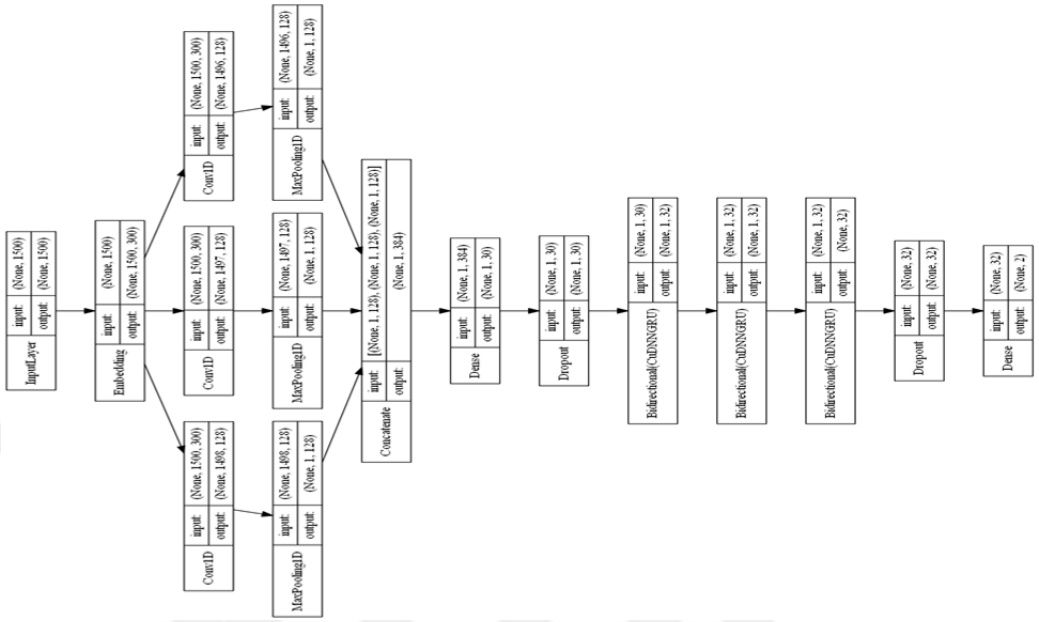


Figure E.3. Summary of our combined CNN-GRU structure

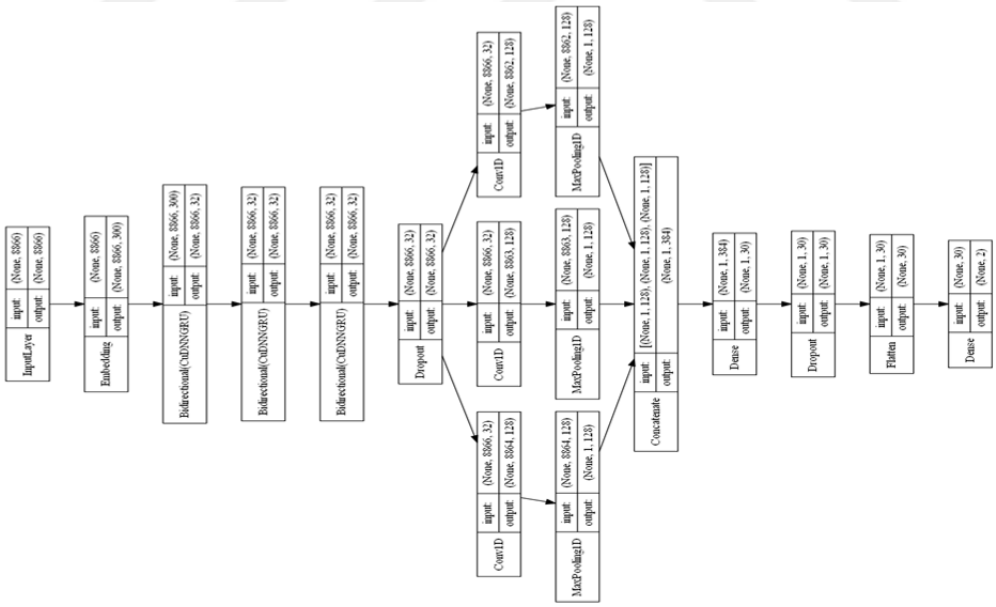


Figure E.4. Summary of our combined GRU-CNN structure

Table E.1. A sample classified list of activities with age and gender attributes of their writers

User Attributes		Activity	Predicted Polarity
Age	Gender		
19	M	Ben çare nedir bilmem Ama çaresizliği iyi bilirim. .Değerli abim :* *(I don't know what the way out is, but I know the despair. Dear brother: *)*	Negative
20	M	Belkide insan sevmeyi bilmediğinden değil, sevgisine layık biri olmadığından yalnızdır	Negative
21	M	Kendini beğenmiş insanları severim. Hiç kimsenin beğenmediği bir şeyi beğenmek, Ayrıcalıktır	Positive
27	M	doğum günümü kutlayan herkeze teşekkür ediyorum.sizi seviyorum yaaaaa:::))))))	Positive
55	M	Her bayramımız böyle olsun iyi bayramlar	Positive
37	M	İstiklal marşın üstadı doğum günün kutlu olsun..M.A.ERSOY	Positive
38	M	Beyler ağlayacaksanız oynamayalım. 4. yıldızı FB kaybediyor. Eziktaş kadar ağlamıyor…	Negative
58	M	Bugün Aylemizin bir dönemi daha kapandı Babamın Amcası Hakın rahmetine kavuştu.	Negative
27	M	Ah be kardeşim.. bilmez miydin kimsenin gidişi senin kadar acıtmazdı canımı..	Negative
20	F	unutulur gidersin kaldırırlar rafa	Negative
20	F	yafff senn yerin ayrı askm ama seda	Positive
27	F	tesekkur eederiimmm ablacimm sende benim canimsinnn	Positive
28	F	Ben yaşarken beni ağlatanlar, ben öldüğüm de arkamdan ağlamasınlar ...	Negative
37	F	bu neeeeeeeeeeeeee fotoya bile bakamıyom çok korkunç!!!!!	Negative
32	F	Saolll coook tesekkur ederım darısı basınaaa	Positive
43	F	off be ne yerim şimdi offff	Negative
54	F	İyiki doğdun yavrum iyiki benim oğlumsun seni cok seviyorum *(Happy birthday, baby, I'm glad you are my son, I love you so much)*	Positive

APPENDIX F: Produced publications from this thesis

Table F.1. A list of publications produced from this thesis

Title	Status	Year
Towards the design and implementation of an OSN crawler: a case of Turkish Facebook users *(International Journal of Information Security Science)*	Published	2020
Privacy Risk Analysis for Facebook Users *(The 28th IEEE Conference on Signal Processing and Communications Applications)*	Published	2020
*Deep Learning-based Sentiment Analysis of Facebook Data: The Case of Turkish Users *(The Computer Journal)*	Accepted	2020
*Facebook Tells Me Your Gender: An Exploratory Study of Gender Prediction for Turkish Facebook Users *(ACM Transactions on Asian and Low-Resource Language Information Processing)*	Submitted	2020
*Your Username Can Give You Away: Matching Turkish OSN Users with Usernames *(International Journal of Information Security Science)*	Submitted	2021

*Journal name and title of the study are subject to change until acceptance of the related study. Due to the review process, the study may also include additional analyses and results to meet possible request(s) of reviewers and editors of the related journal.