EMBEDDINGS-BASED CLUSTERING FOR TARGET SPECIFIC STANCES

A Thesis

by

Ammar Raşid

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
August 2020

Copyright © 2020 by Ammar Raşid

EMBEDDINGS-BASED CLUSTERING FOR TARGET SPECIFIC STANCES

Approved by:

Professor Tanju Erdem, Advisor
Department of Computer Science
Özyeğin University

Assoc. Professor Hasan Sözer
Department of Computer Science
Özyeğin University

Asst. Professor Reyyan Yeniterzi,
Co-Advisor
Department of Computer Science
Sabanci University

Assoc. Professor Cenk Demiroğlu
Department of Computer Science
Özyeğin University

Date Approved: August 2020

Assoc. Professor Walid Magdy
School of Informatics
The University of Edinburgh

To my people...

ABSTRACT

We propose an unsupervised user stance detection method to capture fine grained divergences in a community across various topics. We employ pre-trained universal sentence encoders to represent users based on the content of their tweets on a particular topic. User vectors are projected onto a lower dimensional space using UMAP, then clustered using HDBSCAN. Our method performs better than previous approaches on two datasets in different domains, achieving precision and recall scores ranging between 0.89 and 0.97. We compiled a dataset of more than 300k tweets about UEFA Super Cup's 2019 final, and tagged 12k users as Liverpool FC or Chelsea FC fans. We utilized our method to analyze the stances of Twitter users noting a correlation between user stances towards various polarizing issues. We used the resultant clusters to quantify the polarization in various topics, and analyze the semantic divergence between clusters.

ÖZETÇE

Çeşitli konularda bir topluluktaki hassas taneli sapmaları yakalamak için denetimsiz bir kullanıcı duruş algılama yöntemi öneriyoruz. Kullanıcıları belirli bir konudaki tweet'lerinin içeriğine göre temsil etmek için önceden eğitilmiş evrensel cümle kodlayıcıları kullanıyoruz. Kullanıcı vektörleri UMAP kullanılarak daha düşük boyutlu bir alana yansıtılır, daha sonra HDBSCAN kullanılarak kümelenir. Yöntemimiz, farklı alanlardaki iki veri kümesinde önceki yaklaşımlardan daha iyi performans göstererek 0,89 ile 0,97 arasında değişen hassasiyet ve hatırlama puanları elde eder. UEFA Süper Kupa'nın 2019 finali hakkında 300k'dan fazla tweet'in bir veri kümesini derledik ve 12k kullanıcıyı Liverpool FC veya Chelsea FC hayranları olarak etiketledik. Twitter kullanıcılarının tutumlarını analiz etmek için yöntemimizi kullandık ve çeşitli kutuplaşma sorunlarına karşı kullanıcı tutumları arasında bir korelasyon olduğunu fark ettik. Elde edilen kümeleri, çeşitli konularda polarizasyonu ölçmek ve kümeler arasındaki anlamsal sapmayı analiz etmek için kullandık.

ACKNOWLEDGEMENTS

I would like to thank my dear wife for her profound insights, deep appreciation, and, especially, heartfelt support. I would also like to express my gratitude to Professor Kareem Darwish of Qatar Computing Research Institute in Doha, and Professor Mucahid Kutlu of TOBB University of Economics and Technology in Ankara, for their help and support throughout the research process. I would not be here if it were not for my family, and Şehir University's CS professors were family to me. I am greatly obliged to my professors; Prof. Ahmet Bulut, Prof. Shervin Shirmohammadi, Prof. Ali Çakmak, Prof. Barış Arslan, and Prof. Onur Guzey. Special gratitute to my philosophy professor, Prof. Sinem Elkatip Hatipoğlu. Extended gratefulness to my pre-college teachers. And before all, *Alhamdulillah*.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ABSTRACT . . . . . **iv**

ÖZETÇE . . . . . **v**

ACKNOWLEDGEMENTS . . . . . **vi**

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xi**

DEDICATION . . . . . **xii**

I INTRODUCTION **1**

II BACKGROUND **4**

2.1 Social background 4

2.1.1 Homophily 4

2.1.2 Polarization 5

2.2 Technical Background 7

2.2.1 Word Embeddings 7

2.2.2 Sentence Embeddings 11

2.2.3 Dimensionality Reduction 13

2.2.4 Clustering 18

III RELATED WORK **25**

3.1 Stance detection 25

3.1.1 Supervised text-based stance classification 25

3.1.2 Beyond textual features 26

3.1.3 User stance detection 27

3.2 Polarization on social media 29

3.2.1 Communal polarization 29

3.2.2 Semantic based polarization 30

3.2.3 Quantifying polarization 31

IV METHODOLOGY **33**

4.1 Data . . . 34
4.1.1 Data Collection . . . 34
4.1.2 Data Pre-processing . . . 34
4.2 Features . . . 35
4.2.1 Encoding . . . 35
4.2.2 Averaging . . . 36
4.3 Model . . . 36
4.3.1 Projection . . . 37
4.3.2 Clustering . . . 37
V EXPERIMENTS . . . 40
5.1 Datasets . . . 40
5.1.1 UEFA Dataset . . . 40
5.1.2 Trump Dataset . . . 41
5.2 Validation . . . 42
5.2.1 Baselines . . . 42
5.2.2 Results . . . 43
5.3 Model Analysis . . . 44
5.3.1 Data . . . 44
5.3.2 Features . . . 47
5.3.3 Model . . . 48
VI ANALYSIS AND DISCUSSION . . . 54
6.1 Topics . . . 54
6.2 Quantitative Analysis . . . 55
6.2.1 Stance Alignment . . . 55
6.2.2 Polarization Quantification . . . 56
6.3 Qualitative analysis . . . 58
6.3.1 Semantic Difference in Clusters . . . 59
6.3.2 Limitation Analysis . . . 61
VII CONCLUSION . . . 64
APPENDIX A — DATA COLLECTION KEYWORDS AND DATES . . . 65

REFERENCES **66**

VITA **74**

LIST OF TABLES

1 The taxonomy of clustering algorithms with examples. 19

2 UEFA dataset. 41

3 Trump dataset. 41

4 Benchmark results for Trump and UEFA datasets. PRO and ANTI are for Trump dataset, while LFC and CFC are for UEFA dataset. 44

5 Model's performance with different sample sizes. 45

6 Resulting datasets with different thresholds for the minimum number of unique tweets per user. 46

7 Model's performance with different minimum thresholds of the number of unique tweets per user. 47

8 Model's performance with different sentence embedding models. 48

9 Model's performance with different dimensionality reduction techniques. 49

10 The effect of dimensionality reduction on the model's performance. . . 51

11 Model's performance with different clustering methods. 53

12 UEFA topics used in the analysis. 55

13 RWC polarization measure across topics. 58

14 Possible reasons of misclustering with examples. 62

LIST OF FIGURES

1 Two-dimensional PCA projection of word embeddings returned by the skip-gram algorithm [1]. 8

2 Graph representations of the CBOW and Skip-gram variations of the Word2Vec model [2]. 9

3 Network structure of BERT [3]. 11

4 The architecture of the Deep Average Network (DAN) [4]. 13

5 An illustrative framework of using Universal Sentence Encoder for Semantic Textual Similarity (STS) [5]. 13

6 A comparison of UMAP performance against t-SNE and PCA on three different datasets. UMAP captures the global structure well represented by PCA, as well as the local structure captured by t-SNE [6]. 16

7 The output of a variation of UMAP hyperparameters on a uniformly distributed random sample from a three-dimensional color cube [6]. . . 17

8 Points positions on different steps of Mean-shift. The sequence order is from left to right, top to bottom [7]. 20

9 Comparing different clustering algorithms on toy datasets [8]. 23

10 The convex hull polygons of the output clusters returned by DBSCAN, OPTICS, and HDBSCAN on groundwater aquifer data [9]. Each polygon is considered to represent one groundwater aquifer. 24

11 The pipeline of the framework for a specific topic 33

12 Performance of our model vs UnSup with different sample sizes averaged over Trump and UEFA datasets. 46

13 Model's performance with different UMAP parameters. 50

14 Inter-dependence of stances on different topics measured in (AMI). . . . 57

15 Word Clouds Generated by prominent terms in each cluster for different topics. 61

“ When all think alike, no one thinks very much. ”

Walter Lippmann, The Stakes of Diplomacy

CHAPTER I

INTRODUCTION

Social media platforms have enabled users to discover, meet, and interact with others around the world. The facilitated communication has fostered closely knitted communities to form through gradual flocking of like-minded individuals. Each controversial discussion stirs more polarization among the online users. With enough polarization in the digital atmosphere, each community develops an identity shaped by a set of stances towards various topics. More importantly, polarized communities grow more critical of disconfirming evidence of their beliefs or stances, while uncritically accepting confirming evidence. Depending on the size and the way a group of users are identified as an online community, stances towards multiple topics can be consistent in some communities and diverse in others. Studying which stances are consistent in which communities, and vice versa, provides a more thorough understanding of the factors composing the cyber-demographic polarization.

Polarization can be conceptualized as the divergence of stances towards a set of topics among groups with homogeneous stances. Communities can be polarized on a topic when they have divergent stances towards the same issues. This divergence is driven by the social psychology of people. Individuals tend to congregate more around those who share their social content. As mentioned in the proverb, "birds of a feather, flock together". Once in social groups, people experience and exert social pressure to stick to certain ideas. This in turn fosters the formation of homogeneous social groups that are exposed to similar information and are critical to ideas disconfirming their owns. Dissecting a polarization instance into its composing stance targets, not only provides a deeper understanding of the polarization subjects, but also exposes the correlation between different polarizing issues. We propose a novel approach to automatically

detect the stances of social media users.

Stance detection is the computational task of determining the position of a user towards an issue or an entity. Stance detection has been relying on supervised or semi-supervised learning methods that required annotating the stances of all or some of the users. Previous, unsupervised stance detection approaches, however, were limited to platform-specific features, such as Twitter's retweets. In this work, we propose an unsupervised user stance detection approach that requires just the stance-bearing text of users. Specifically, we employ transformer sentence embeddings to map users based on the text of their tweets about a specific topic, onto an n-dimensional embeddings space. Next, we project users into a lower dimensional space in a manner that brings similar users closer together and dissimilar users further apart, where similarity is computed based on their embeddings vectors. Lastly, the projected users are clustered.

We validate our approach on two different datasets in two different domains, namely, politics, and sports. Furthermore, we utilize our method to extensively analyze a real-life case study of our approach. The novelty of our approach is in mapping the hierarchy of stances, in unsupervised manner, using sentence embeddings pre-trained on unlabeled corpus, and providing a framework for extensive quantitative and qualitative analysis of the results. Our contributions in this work are four-fold:

- We provide an embedding-based unsupervised user stance detection approach that outperforms previous approaches, without requiring platform-specific features, or fine-tuning language models.

- We compiled a large dataset of tweets related to the UEFA Super Cup 2019's final, containing more than 300k tweets, and accurately tagged nearly 12k users.

- Using embeddings with projection and clustering, we provide a comprehensive framework to analyze fine-grained polarization between groups over various topics, and we show the efficacy of this approach in performing unsupervised stance detection in general.

- Given embeddings representations of user tweets on specific topics, we project and cluster users to determine if they are polarized on those topics. This allows us to measure cross-topic mutual information to determine the alignment of polarization across multiple topics.

CHAPTER II

BACKGROUND

This is an interdisciplinary work of computational science. Thus, in this chapter, we lay down the sociological foundation pertinent to the studies presented in this work. The social theories and phenomena addressed in this work are an integral part of the story told in our findings and the potential impact we hope to make.

2.1 Social background

2.1.1 Homophily

In this section we review one of the strongest social phenomena shaping communication spaces, and influencing the flow of information diffusion. We provide an overview about the definition, the driving factors, and the significant consequences of homophily in a sociological context. This section is inspired by the work of [10]. Homophily is the tendency of individuals to connect with those similar to them. Proverbially, "birds of a feather flock together". In a social system, homophily is defined in terms of relationships and similarity. Homogeneity in personal networks is the sociological projection of homophily onto interpersonal relationships. Whether they are hang-out friendships, professional acquaintanceships, or marriage, relationships vary in types and degrees. Similarity is defined by informal, formal and ascribed status in *status homophily*. Whereas, *value homophily* is based on values, attitudes and beliefs similarities. The demography of potential tie pool in a social network lay the *baseline homophily*. Explicit homophilial effects measured above the opportunity set are distinguished as *inbreeding homophily*. Empirical studies of homophily have been limited due to the scarcity

of recorded social networks and insufficient computational power. However, the increasing popularity of online social networks and accelerating computational technology advancements have introduced new potentials for empirical social studies allowing researchers to conduct large-scale studies on homophily simultaneously on multiple characteristics.

2.1.2 Polarization

In this section we provide an overview of the social phenomenon of polarization. We review the definition of polarization in the context of opinion mining. Furthermore, we summarize some of the social effects associated with polarization that have inspired this work.

Polarization is the phenomenon of opinion divergence among close-knit clusters of homogeneous individuals. Polarization in social media can be detected in various scenarios, ranging from social issues (e.g. abortion and gun control) [11], political issues (e.g. political orientation) [12], or sports fans. Polarization can play a direct role in shaping the atmosphere of its context. For example, increased political polarization in the US congress, in the form of the dominance of intra-group coordination over inter-group coordination, led to a decline in legislative effectiveness [13]. Paradoxically, prolonged political polarization led to the increased partisanship of the largest coalition, which in turn facilitated passing more bills [13]. Studying polarization is motivated by:

- Measuring the degree by which a society is getting more (or less) polarized.
- Analyzing the effect of polarization on various societal functions.
- Derive strategies to control polarization, whether it is viewed as a state of a society, or a property of opinion formation dynamics [14].

Segregation Structural cleavages in social networks reflect segregation among ethnic, gender, religious and social status groups. The strength of ties segregation occurs among reflects various sociological issues. Although, online social networks are typically richer in weak ties than strong ones, small circles of strong ties have been the focus of the majority of research on segregation in social networks [15]. Weak ties reflecting, even superficial, cross-ethnic contact can reduce inter-group prejudice. Whereas, homophilous network selection and social influence potentially result in an *"echo chambers"* effect. The overlap of off- and online social networks provide opportunities to study the generalization of social phenomena.

Echo-chamber Effect It has been argued that polarization is fostered by greater interaction between like-minded individuals [16]. With the advent of internet, individuals could find more of their like-minded counterparts, undermining communication barriers such as language or location. As Col. Robert Bateman puts it:

> " Once, every village had an idiot. It took the internet to bring them all together. [17] "

The abundance of diverse information sources allows information to be fed through narrow channels tailored, manually, or algorithmically, through recommendation systems [18, 14] or web search engines [19], to personalized tastes. Polarization has been found to be reinforced by "echo-chamber" or "filter bubble" effect, where individuals are exposed only or mostly to information they agree with.

Biased Assimilation Polarized groups process information in a biased manner, referred to in social psychology as "Biased Assimilation", where individuals readily accept confirming evidence, but are rather critical when provided with disconfirming evidence. Coupled with echo-chamber effect, biased assimilation has been found to push individuals to the extreme versions of their opinions [20]. Interestingly, homophily alone, without biased assimilation, does not suffice to cause significant divergence of

opinions to polarize a society [14].

2.2 Technical Background

Any computational analysis of a social phenomenon falls under the name of *Social Computing*, also known as *Computational Social Science*. In order to understand the impact, limitation, and scope of our interdisciplinary story, it is important to understand the techniques we used and the motivation behind choosing them. In this chapter we go over some of the technical concepts and algorithms that eventually led to our choice of the algorithms we utilized to address our research problem of stance detection.

2.2.1 Word Embeddings

In this subsection, we review the idea of word embeddings and visit some of the common word embeddings algorithms.

What are Word Embeddings? Word embeddings are a way to represent text in a way that similar words have similar representation. Whereas word embeddings techniques map words to vectors, learned in a neural network, or what may resemble it. The benefit of word embeddings as a means of text representation culminated in the ability to represent different aspects of each word, while maintaining a much smaller size of features, as opposed to classic sparse word representation techniques, such as one-hot encoding [21]. In fact, one of the key breakthroughs in natural language processing, and deep learning, was the introduction of low-dimensional vector representation of words. The importance of this breakthrough is best described in the following quote.

> "One of the benefits of using dense and low-dimensional vectors is computational: the majority of neural network toolkits do not play well with very high-dimensional, sparse vectors. ... The main benefit of the dense representations is generalization power: if we believe some features may

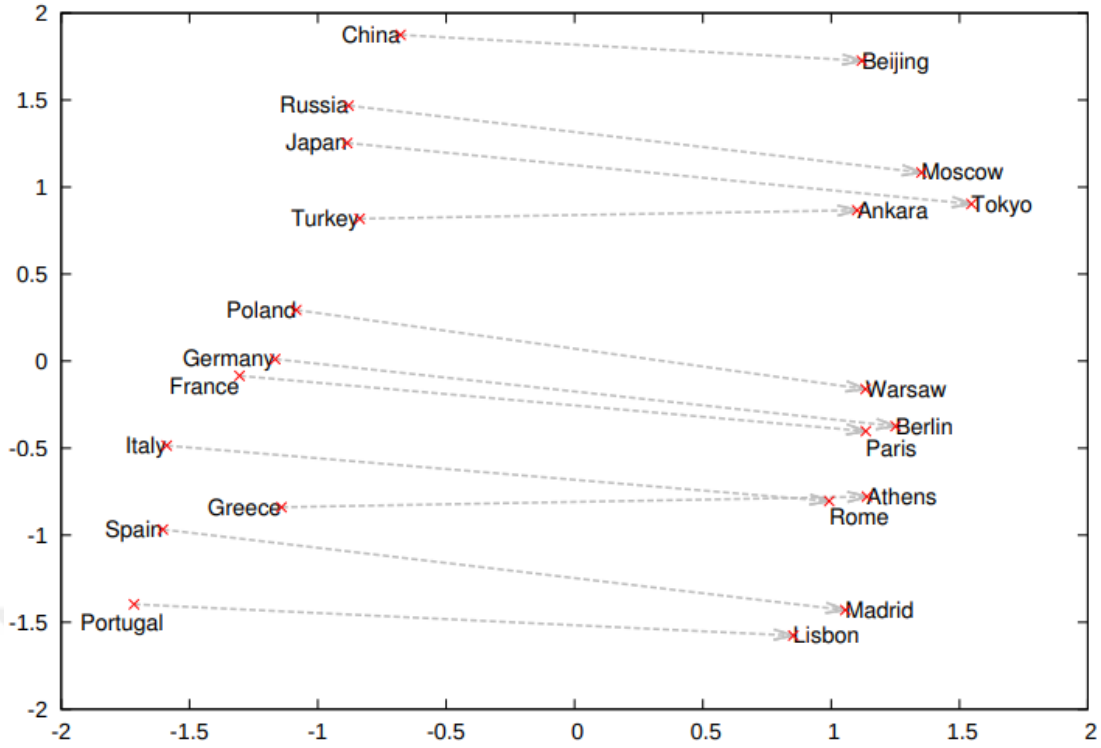


Figure 1: Two-dimensional PCA projection of word embeddings returned by the skip-gram algorithm [1].

> provide similar clues, it is worthwhile to provide a representation that is able to capture these similarities." [22, p. 92]

The main assumption upon which word embeddings techniques were developed is that words with similar meanings, share similar contexts, or are used similarly: "You shall know a word by the company it keeps!"[23, p. 11]. Figure 1 shows a 2-dimensional visualization of word embeddings, learned with the algorithm of Skip-Gram [1], which we will visit shortly after. The figure shows the ability of word embeddings to organize concepts and learn the relationships between them without any supervision.

Algorithms Word embeddings approach can be categorized by how words are representation into static and dynamic word embeddings [3]. Static word embeddings refer to approaches where each word has exactly one vector representing it, thus failing to capture polysemy. Whereas dynamic, or often called, Deep Contextualized, approaches of word embeddings incorporate the context of each word into its embedding. That means,

that the same word, in different contexts, can have different vector representations.

Static Word Embeddings The idea about static word embeddings is to learn stand-alone vector representation of words from a text corpus. The goal was to estimate a vector space such that words that co-occur are embedded closer to each other in an n-dimensional euclidean space. Through the probabilistic modelling of contextual information, this goal was achieved to a certain extent using what is called 'Word2Vec' [24, 1]. Word2Vec approaches provide a dense low-dimensional representation of words. The authors proposed two variations of their model; using a contextual window to predict a word in what is known as Continuous Bag Of Words (CBOW), and using a word to predict its context, in what is known as Skip-gram. Figure 2 shows the graphical representation of CBOW and Skip-gram models, as depicted in [2].

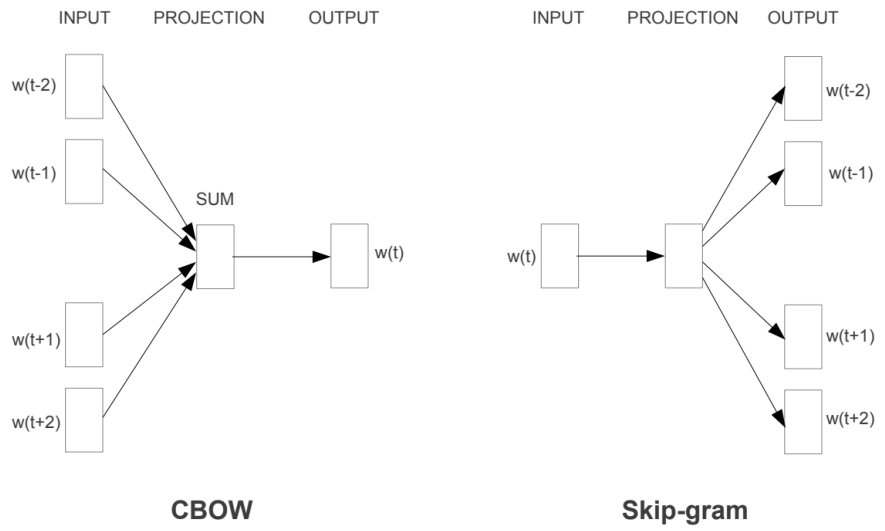


Figure 2: Graph representations of the CBOW and Skip-gram variations of the Word2Vec model [2].

One limitation of the word2vec model was not using the global information of word co-occurrence statistics in a corpus. This information was captured in the Global Vector (GloVe) model [25]. Moreover, the fact that Word2vec assigns a single vector for each word introduced several limitations to the approach. First, word2vec model

ignores the morphology of words. This limitation was addressed in fastText, an extension of the Skip-gram model to learn representations for character, n-grams, instead of word n-grams [26, 27]. However, static word embeddings fail to capture polysemy. For example, the English word 'open' has a single vector representation in a static word embedding model, while it can mean 'available', 'honest', or 'not covered'. Furthermore, static word embeddings do not work well in transfer learning, as they are not designed to be fine-tuned for downstream specific tasks.

Dynamic Word Embeddings Accounting for the context-dependent nature of the words, dynamic embeddings provided an alternative word representation that changes with the context. This approach was invigorated by the deep Context Vector (CoVe) [28]. CoVe utilized long short-term memory (LSTM) encoders trained in a sequence-to-sequence machine translation (MT) model to generate contextualized embeddings for words. However, CoVe suffered from two main limitations. First, acquiring cross-lingual data for a machine translation model is very expensive. This is where Embeddings from Language Models (ELMo) [29] overcame this limitation by extracting the internal layers of a deep bidirectional language model (biLM) pre-trained on large unlabeled corpus. Second, relying on LSTM networks deprives the model from the ability to capture long-range dependencies. This is where Generative Pre-Training (GPT) model [30] employed transformer networks [31] to learn a linguistic structure with longer range.

BERT While GPT achieved great results in a variety of natural language processing tasks, the model was restricted to left or right context only, due to the unidirectional objective of the language model. This is where Bidirectional Encoder Representations from Transformers (BERT) [32] was designed to capture the context on both directions using a deep bidirectional transformer. Moreover, BERT was trained to guess a randomly masked word, in the novel task of Masked Language Model (MLM). This allowed BERT to learn deeper and more contextualized representations. Furthermore,

BERT uses WordPiece embeddings [33] as opposed to word embeddings. This increases the generalization power of BERT by equipping it with the balance between the flexibility of ”character”-delimited models and the efficiency of ”word”-delimited models. Figure 3 shows the network structure of BERT as illustrated in [3]. BERT's framework involves three main blocks. First, converting words to vectors; token embeddings, segment embeddings, position embeddings. Second, they chain a series of encoder sub-blocks, where each encoder's output is fed into the next one. The role of each encoder block is to learn and encode the relationship between the input representations. Third, the decoder which takes the output of the encoder as input and predicts the next word. One caveat is that when predicting an output word, only a part of the context has most of the relevant information. Encoder and decoder blocks involve *attention* [31] operations, which is basically using only the part of the input with the most relevant information. Attention mechanisms help improve the spatial understanding of the model.

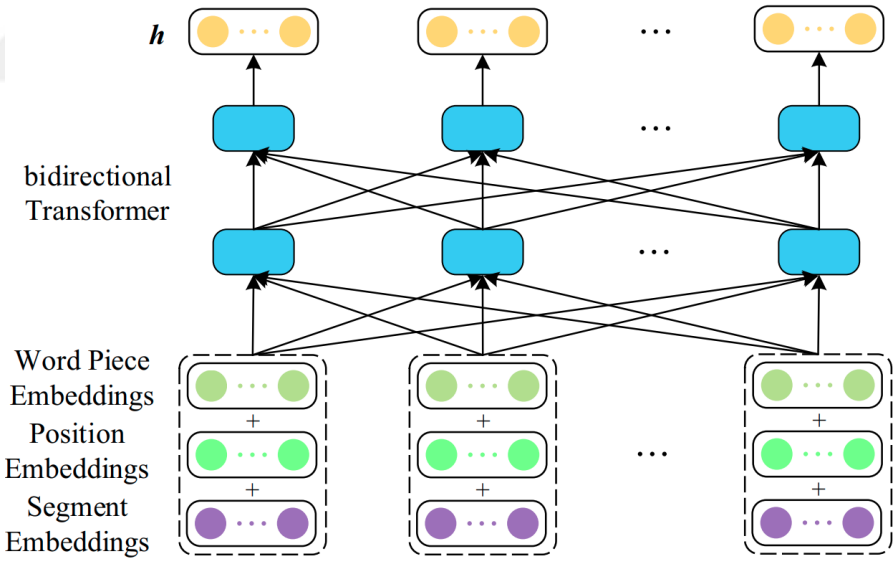


Figure 3: Network structure of BERT [3].

2.2.2 Sentence Embeddings

Many natural language tasks deal with sentences, such as tweets, reviews, or comments, rather than words or tokens. To get the embedding of a sentence, the common naive

approach is to take the average of a sentence's word vectors. This approach, so called the Bag-of-Words (BOW) approach, despite ignoring the order of the words, provides a reasonable baseline for many downstream tasks that depend more on the signal of words than their order, such as text classification, sentiment analysis, and document clustering.

Universal Sentence Encoder Taking context aware word embeddings a step further, researchers in [34] introduced two variations of Universal Sentence Encoder (USE). For the first variation, they use a transformer based sentence encoding model to obtain context aware word representations using the encoding sub-graph of the transformer architecture [31]. The context awareness of word representation is the result of the attention mechanism used in the transformer model, which captures the order and the identity of a word in the embedded sentence. Word representations are aggregated by element-wise summation to obtain a fixed length sentence embedding vector. The other variation is using Deep Averaging Networks (DAN) [4] (figure 4), where input word and bi-gram embeddings are averaged together before passed through a feed-forward deep neural network (DNN) to produce the sentence embeddings. Although the transformer based model showed more promising results, the DAN based model requires less computational resources.

Sentence-BERT Averaging the BERT embeddings of words in a sentence results in sentence embeddings that perform rather poorly in semantic similarity tasks. Moreover, finding similar pairs of BERT embeddings is very computationally expensive; takes up to 65 hours. However, [35] proposed modifying BERT to use twin and triplet neural networks, also known as "Siamese" networks [36], to derive semantically meaningful sentence embeddings, in what they called Sentence-BERT, or SBERT. The sentence embeddings generated by SBERT can be compared using cosine-distance. Furthermore, finding the most similar pairs of SBERT embeddings takes only 5 seconds, without compromising accuracy.

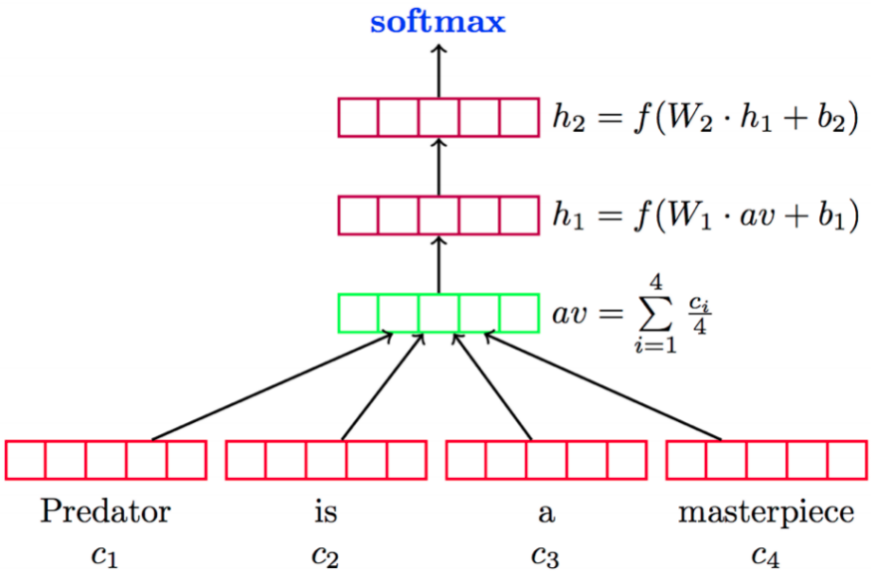


Figure 4: The architecture of the Deep Average Network (DAN) [4].

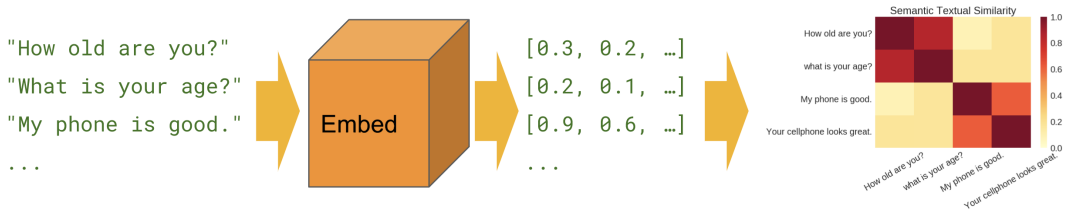


Figure 5: An illustrative framework of using Universal Sentence Encoder for Semantic Textual Similarity (STS) [5].

Universal Sentence Encoders (USE) and SBERT excel at embedding the semantic similarity between sentences. USE performs significantly better in Semantic Textual Similarity (STS) benchmark than BOW approaches. Moreover, they preserve the advantages of using transformer-based embeddings; USE and SBERT return a pre-trained model that can be fine tuned for downstream tasks in a transfer learning manner.

2.2.3 Dimensionality Reduction

Dimensionality reduction is an important concept in information theory. Reducing the dimensionality of the representation of data implies the elimination of multiple variables explaining the data, while preserving as much information as possible. In this section we go over the motivation behind the process of dimensionality reduction. Furthermore,

we review three comparable dimensionality reduction algorithms, concluding with a discussion about the details and distinguishing advantages of the algorithm we chose for our framework.

Motivation Visualizing any embedding space is possible only up to three-dimensional spaces. Additionally, the performance of any clustering algorithm often depends on the distance metric used to group clusters in an embedding space, which in turn depends on the dimensionality of the space. For example, the euclidean distance between two n-dimensional data points requires n subtractions; one for each dimension in each of the two data points. Since clustering is particularly needed in analyzing high-dimensional sparse data, it became increasingly needed to reduce the dimensionality of data without losing much of the information. Moreover, processing large high-dimensional datasets can be computationally cumbersome. Thus, it is a valuable solution for machine learning applications to compress the representation of data by minimizing the number of variables needed to explain the data.

Principal Component Analysis (PCA) One of the most common approaches for dimensionality reduction is Principal Component Analysis (PCA), invented by Karl Pearson in 1901 [37]. The objective of PCA is maximize variance while preserving long pairwise distances. PCA assumes that there is a latent correlation between the variables of a set of observations and tries to project them onto a lower dimensional space of uncorrelated variables. The interpretability of PCA shows how much information can be encoded in how many dimensions. Furthermore, PCA can be reversed; original vectors can be approximately reconstructed from their PCA projections. Despite PCA being a powerful dimensionality reduction method, it is inherently linear; PCA maps a high-dimensional space to a lower-dimensional space by multiplying the source vectors with the PCA eigenvectors, and matrix multiplication is inherently linear. However, PCA is computationally expensive; for n data points represented with k features, the complexity of computing the co-variance matrix is $O(P^2n)$, and the eigenvalue decomposition

complexity is $O(P^3)$, making PCA's total time complexity $O(P^2n + P^3)$.

t-distributed Stochastic Neighbor Embedding (t-SNE) Unlike PCA, t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm preserves neighbor similarities; only small pairwise distances are preserved [38]. Preserving large pairwise distances can lead to poor visualization especially with non-linear manifold structures, such as cylindrical or spherical geometric shapes. Whereas t-SNE centers a Gaussian distribution P_i about a data point X_i and measures the density of all other data points X_j under that distribution P_i. Normalizing densities give us a set of probabilities P_{ij} proportional to the similarities between data points X_{ij}. While we could control the number of principal components to preserve in PCA, in t-SNE, the main hyper-parameters are for the Gaussian distribution used to center around each data point, which is called *perplexity*. Despite the power of t-SNE, it suffers from crucial cons that make PCA preferred in some scenarios. Unlike the deterministic solution of PCA, the method of t-SNE is stochastic, as there is no analytical solution for t-SNE's optimization criterion. Furthermore, the stochastic nature of t-SNE makes the method less explicable while not necessarily representing global trends in the data with accuracy, as large pairwise similarities are ignored. Therefore, t-SNE is typically used for visualization rather than pre-processing.

The original t-SNE proposed by [38] has a time and memory complexity of $O(N^2)$ for N data points. However, the implementation of the method evolved by [39] by using vantage-point trees to compute sparse pairwise similarities, making the final time complexity $O(N \log N)$ and memory complexity $O(N)$.

Uniform Manifold Approximation and Projection (UMAP) Assuming that the input data is a locally connected, uniformly distributed Riemannian manifold, [6] showed that it is possible to model the manifold in a fuzzy topological space [40]. Unlike t-SNE, UMAP is able to model global trends in the data as large pairwise similarities are

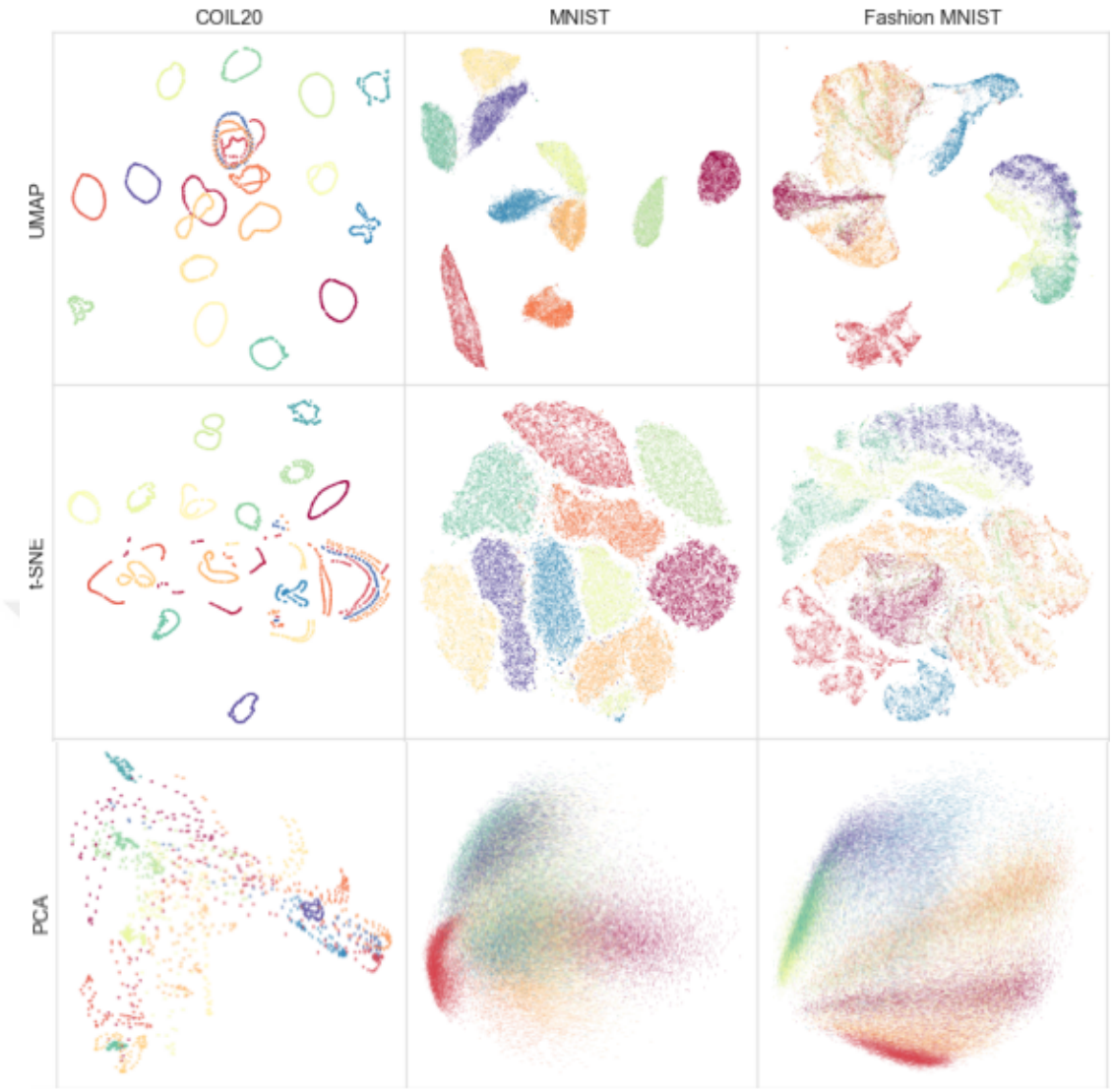


Figure 6: A comparison of UMAP performance against t-SNE and PCA on three different datasets. UMAP captures the global structure well represented by PCA, as well as the local structure captured by t-SNE [6].

captured, while preserving a visualization quality competitive with t-SNE. Moreover, UMAP has no computational restrictions on the dimensionality size of the embeddings, making it computationally more efficient than t-SNE. In other words, clusters relatively closer to each other are projected closer to each other than dissimilar clusters. This concludes a major advantage for UMAP over t-SNE as UMAP is viable for general-purpose dimensionality reduction, not just high-quality visualization like t-SNE. The output of UMAP is compared to the output of t-SNE and PCA on different data sets and shown in figure 6.

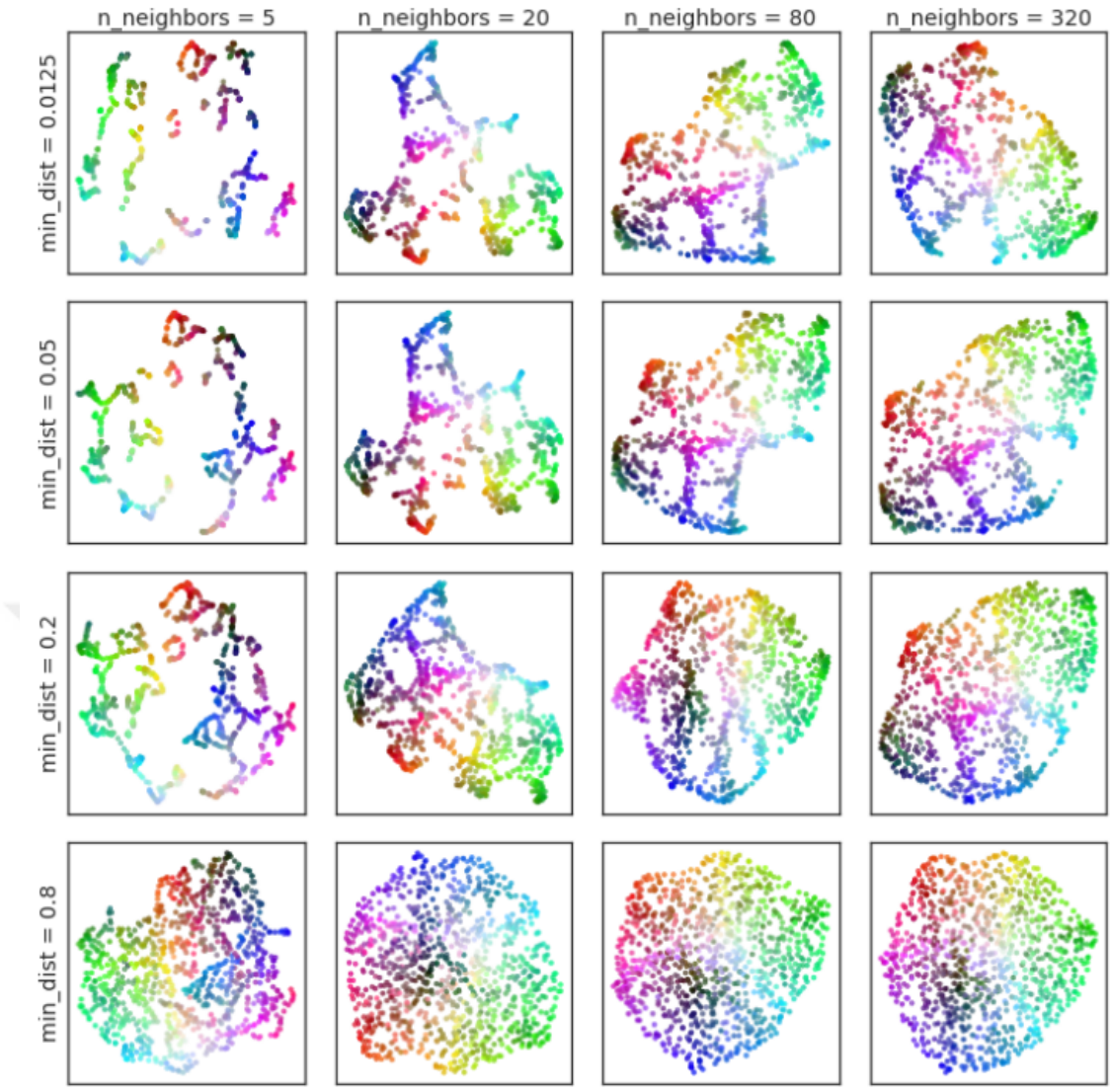


Figure 7: The output of a variation of UMAP hyperparameters on a uniformly distributed random sample from a three-dimensional color cube [6].

Although powerful, UMAP is no silver bullet; the solution generated by UMAP has to be read considering a couple of underlying parameters. The solution of t-SNE requires specifying the number of components or output clusters. Whereas, The solution generated by UMAP is a subject of two main parameters;

- *min_dist:* the minimum distance between two elements to be considered neighbors. This parameter controls how tightly can UMAP pack elements together.

- *n_neighbors:* the number of neighbors in a local structure. This parameter controls the balance between local and global structures in the data.

Increasing the minimum number of neighbors in a cluster results in a generally fewer but larger clusters, and vice versa. The lower the minimum distance between neighbor points, the further output clusters are from each other. These two hyper-parameters, despite giving an arguably comprehensive control over UMAP's results, necessitate the need to read UMAP results with careful consideration to the model's parameters. The effect of these parameters on the output of UMAP on a uniformly distributed random sample from a three-dimensional color cube is shown in figure 7.

2.2.4 Clustering

In this section we go over the machine learning concept of clustering. We provide an overview of the different types of clustering approaches, eventually concluding with the discussion of the clustering algorithm we used in our framework.

Clustering Taxonomy While embedding models organizes concepts in such a way that similar concepts are embedded closer to each other (e.g. Moscow with Tokyo and Russia with Japan 1), there could be a latent hierarchical structure of concepts with different granularity levels implicitly embedded in the embedding space. Clustering is the process of grouping similar concepts together in a *cluster* of concepts (e.g. Cluster of capital cities: Moscow and Tokyo, countries: Russia and Japan). The main advantage of clustering over classification learning approaches is that they are unsupervised; clustering does not require any sort of supervision, such as labels, for training. There are numerous applications of clustering in market research, pattern recognition, bioinformatics, and many other fields. For example, hierarchical clustering, where clusters are organized in a hierarchy, are used to capture the hierarchical map of biological pathways [41]. Clustering is typically needed when the problem studied suffer from data sparsity and scalability, such as building a recommendation system of millions of items based on millions of ratings. E-commerce conglomerates can have their recommendation systems target users in scale of millions, with significantly sparse ratings

information; each user can rate only an infinitesimal portion of the items set. A clustering approach has been used to partition users in clusters, and apply recommendation algorithms separately on each partition [42].

Clustering algorithms can be divided, by how the shape of clusters are defined, as centroid-based / parametric clustering, and density-based / non-parametric clustering. In centroid-based clustering algorithms assume the shape of clusters, which allows such algorithms to work well on small amounts of data. However, density-based clustering algorithms let the data define the shape of clusters. In both centroid and density based clustering algorithms, elements break apart naturally according to the shape of the clusters; elements are organized around centroids or high-density regions. Clustering algorithms can also be divided, by the complexity of the clustering resolution, into flat and hierarchical clustering. In flat clustering, clusters are partitions of similar data elements. Whereas hierarchical clustering algorithms organize elements into a set of nested relationships between clusters. The taxonomy of clustering algorithms with examples for each type is shown in table 1.

	Flat	Hierarchical
Centroid/ Parametric	K-means GMM	Ward Complete-Linkage
Density/ Non-parametric	DBSCAN Mean-shift	HDBSCAN

Table 1: The taxonomy of clustering algorithms with examples.

Density-Based Clustering The work presented in this study is mainly concerned with density-based clustering. Therefore, we opted not to inundate the reader with irrelevant details about other clustering classes.

The general idea about density-based clustering is that points within a *dense* region are considered a cluster. The density of a region in an embedding space is the probability distribution of the set of data in that region, also known as probability density function (PDF), which can be estimated using a method like Kernel Density Function (KDE).

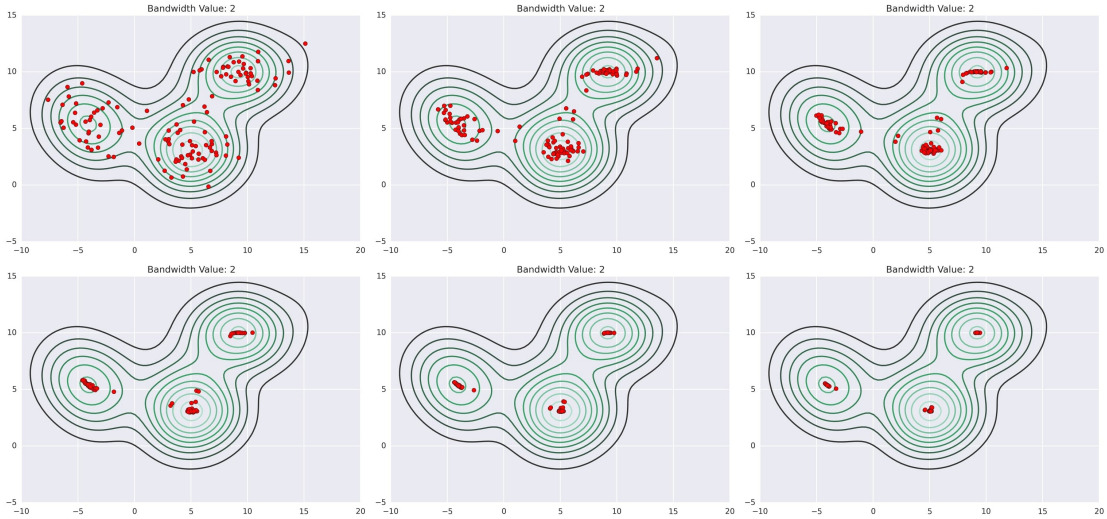


Figure 8: Points positions on different steps of Mean-shift. The sequence order is from left to right, top to bottom [7].

The density function is estimated by placing a kernel; a weighting function such as Gaussian or linear kernels, on each point in the set of data in a region and adding all individual kernels to generate a probability surface.

Mean-shift The KDE idea has been leveraged in clustering. By repetitively moving each point in an embedding space towards the nearest peak on the KDE surface and re-estimating the density function, data points are going to be organized in separate dense regions, or clusters. This clustering method was introduced by Fukunaga and Hostetler in 1975 and referred to as mean-shift algorithm [43]. One caveat though, the KDE surface varies depending on the kernel bandwidth; if the bandwidth is very small; the kernel is too thin and tall, each data point will be on a separate peak by itself, resulting in a organization of the data where each point is a cluster by itself. Whereas, if the kernel bandwidth is too large; the kernel is too wide, all data points will be in the same density region, resulting in a single cluster for all points. The runtime complexity of mean-shift algorithm is $O(N^2)$, where N is the number of data points.

A sequence of data positions in consequent iterations of mean-shift algorithm is shown in figure 8.

DBSCAN The structure of data points in an embedding space can be conceptually modelled as a graph, where each data point is a vertex in that graph, and vertices are connected only if the distance between them is at most ϵ. A vertex is considered a core object if it is connected to a minimum number of vertices m_{pts}. A subset of core objects that are directly and transitively ϵ-reachable, also known as *density-connected*, form a cluster. This method is rooted in the formalization of the definition of clusters by by R.F.Ling [44], who proposed a related method with an estimated runtime complexity of $O(N^3)$, where N is the number of data points. However, it was until 1996 that this method was re-formalized as density-based spatial clustering of applications with noise (DBSCAN) [45] with a significant performance improvement; the runtime complexity of DBSCAN is $O(N^2)$, and the space complexity is $O(N)$.

The nature of this method implies that some data points might end up being non-core objects; a non-core object is a an object that is more than ϵ-far from any other object. In other words, the ratio of data points clustered by DBSCAN depends on ϵ and m_{pts}. Moreover, clusters regions can intersect; an object x can be reachable from more than one cluster, in which case the cluster x is assigned to is indeterministic.

OPTICS DBSCAN suffers from a major weakness when used to detect clusters in data of varying density. Moreover, the DBSCAN dependency on the reachability parameter ϵ is problematic as it is hard to determine which ϵ value produces the most accurate clustering structure. To address this problem, researchers in [46] improved over DBSCAN by adding a couple of concepts. Firstly, while a core point in DBSCAN is a point that has at least m_{pts} points within ϵ distance, a point is also assigned a *core-distance*; the distance to the m_{pts}-nearest neighbour. The core-distance is meant to capture the density information of points assignments to clusters. Secondly, the *reachability-distance* of a point x_i from another point x_j is the maximum between $dist(x_i, x_j)$; the distance between x_i and x_j, and the core-distance of x_j. Additionally,

a priority queue, such as indexed heap data structure, is used to create an augmented ordering of the data that represents the density-base clustering structure, hence the name of the algorithm; ordering points to identify the clustering structure (OPTICS). The nature of OPTICS algorithm undermines the importance of the parameter ϵ, as ϵ can simply be set to the maximum possible value. However, sufficiently large ϵ would consider the entire database as one big interesting cluster, in which case core-distance and reachability-distance are always defined, increasing the runtime of the algorithm significantly; $O(N^2)$, where N is the number of points in the data set. Whereas, setting ϵ smaller than the maximum distance between any two points in the data set allows querying a neighbour hood from the spatial index to be $O(\log N)$, in which case querying the total runtime complexity, which is $O(N)$ queries in the worst scenario, adds up to $O(N \log N)$. In other words, the parameter ϵ, strictly speaking, is not required, but setting it carefully has a heavy influence on the runtime of the OPTICS algorithm.

A comparison between the output clusters of Mean-shift, DBSCAN, and OPTICS is shown in figure 9.

HDBSCAN A conceptual and algorithmic improvement over OPTICS was introduced as a hierarchical density-based clustering method (HDBSCAN) [47]. While it is difficult to choose the reachability parameter in DBSCAN or OPTICS when the scale of the data is not well understood, HDBSCAN generates DBSCAN solutions for varying ϵ values and selects clusters with the best stability over ϵ. Therefore, the first major improvement of HDBSCAN is that it requires only the m_{pts} parameter, getting rid of the reachability parameter ϵ altogether. The second major improvement HDBSCAN offers over its predecessors; DBSCAN and OPTICS, is the ability to generate a tree of significant density-based hierarchical clusters for an infinite range of density thresholds. To understand HDBSCAN, we must first mention their mutual-reachability distance; the mutual-reachability distance between two points x_i and x_j is the maximum between the core-distance of x_i, the core-distance of x_j and the distance between x_i and x_j, whether

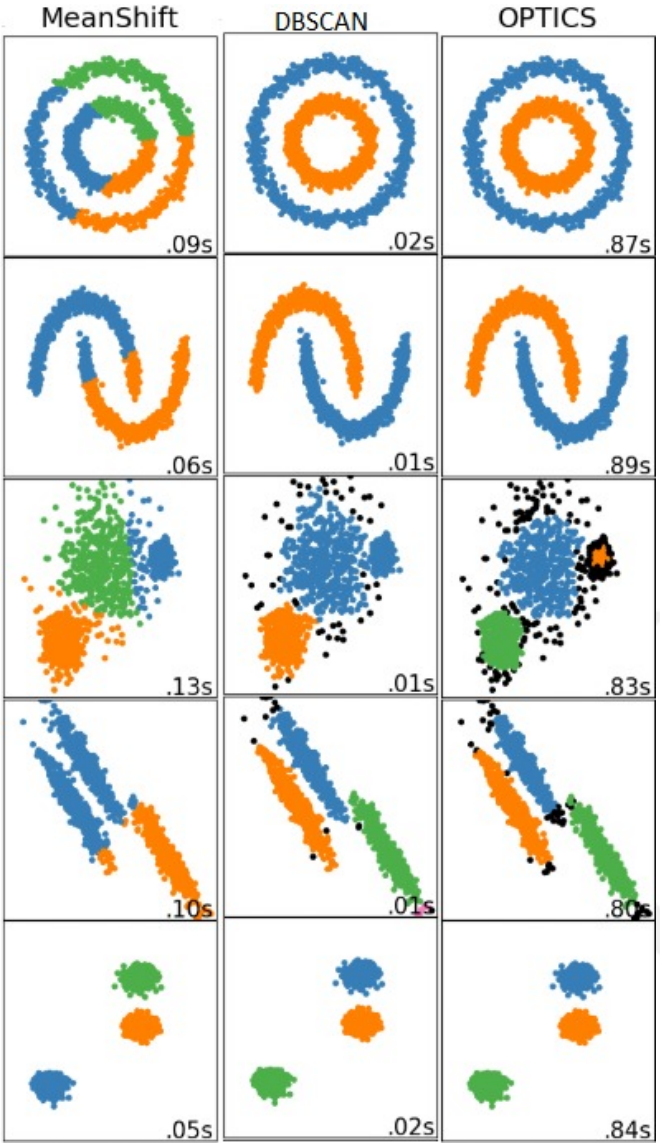
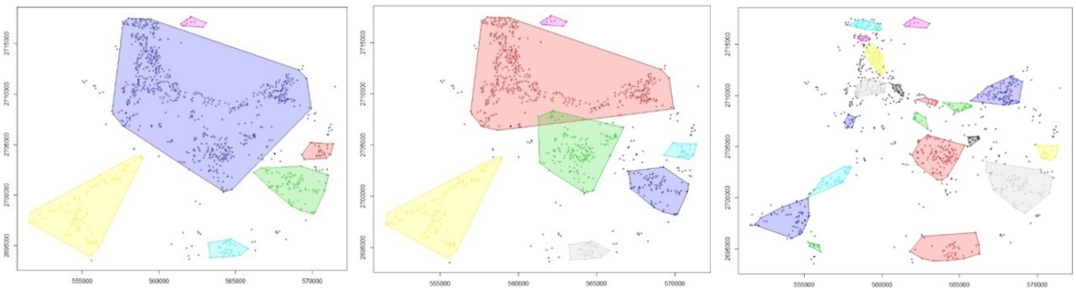


Figure 9: Comparing different clustering algorithms on toy datasets [8].

it is euclidean, or cosine, or any other distance measure. Mutual reachability distance is used to make sure that in order to connect components during the clustering process, they not only have to be close in the euclidean space, but they also have to be dense.

Since it can be incomprehensible to read the hierarchical cluster tree generated by HDBSCAN for the infinite range of density thresholds, this tree is simplified; As a cluster shrinks with an increasing minimum density threshold, it either disappears or gets divided into smaller clusters, without allowing singleton clusters. The authors of HDBSCAN paper [47] not just introduced for hierarchical density-based clustering, but they also proposed a novel cluster stability. They adapt the notion of excess of mass [48] is



(a) DBSCAN (b) OPTICS (c) HDBSCAN

Figure 10: The convex hull polygons of the output clusters returned by DBSCAN, OPTICS, and HDBSCAN on groundwater aquifer data [9]. Each polygon is considered to represent one groundwater aquifer.

adapted to measure the persistence of clusters against increasing the density threshold. They extract the most significant clusters by optimizing on the persistence of clusters; clusters maintaining a large amount of elements for longer time before disappearing due to the increasing density threshold. The convex hull polygons of the output clusters returned by DBSCAN, OPTICS, and HDBSCAN are shown for comparison in figure 10.

CHAPTER III

RELATED WORK

In this chapter we review previous works related to our study. Our framework consists of multiple blocks of sub-tasks. Therefore, our literature review discusses each block separately, as well as interdisciplinary works that have addressed the same problem from different angles. Although each block in our framework is studied separately in previous works, our framework connect all these blocks to tell a comprehensive story from the data.

3.1 Stance detection

In this section we provide an overview of the computational task of detecting the stances of users. We go over multiple studies that defined stance detection in different ways, and consequently approach the problem from various angles.

3.1.1 Supervised text-based stance classification

Stance detection was featured as task-6 in the international workshop on semantic evaluation (SamEval 2016)[1]. The task provided a set of tweets about 5 different topics labeled as FAVOR, AGAINST, or NEITHER. Stance can be formulated in different ways. In SamEval's task, stance was formulated as classifying a piece of text as in favor of or against a given target [49]. The MITRE corporation submitted the best performing model in the workshop [50]. They projected a one-hot encoding representation of the data using a 256-dimensional pre-trained SkipGram embedding layer. The projected vectors were fed to a recurrent layer of long short-term memory (LSTM) units [51].

[1] http://alt.qcri.org/semeval2016/task6/

The LSTM output was then passed to a rectified linear unit activation function to make a classification architecture which they trained on a several hundred labeled examples. This early work on stance detection showed the ability of pre-trained word embeddings, even if static like SkipGram, to capture stance information.

Other researchers used pre-trained word embeddings in many different ways for the same task. [52] proposed the use of bidirectional conditional encoders to learn target-dependant tweet representation. Extending the work with bidirectional encoders and pretrained word embeddings, [53] focused on exploiting the latent cross-target information in stances. They introduced the CrossNet architecture which used self-attention mechanism [31, 54] to highlight the stance-bearing parts of a sentence. The CrossNet showed that there can mutual information between different stance targets. [55] took stance classification from cross-target; applying what has been learnt from a source target to a destination target, to multi-target; jointly learning stances towards multiple related targets. Introducing dynamic memory-augmented network DMAN, [55] combined bidirectional encoders and target-based attention mechanisms with memory modules [56, 57] to capture stance-indicative information. Using end-to-end memory networks had been used to predict whether a document agrees, disagrees, discusses, or unrelated with respect to a given target [58]. Moreover, memory networks provides the ability to extract snippets of evidence for the predictions they generate. The association degrees with the memory module can be visualized to qualitatively analyze the relation between stances towards multiple targets. This idea has been leveraged later by [55] as they showed that the association degree with memory for the target Donald Trump and the stance FAVOR is similar to association degree with memory for the target Hillary Clinton and the stance AGAINST, and vice versa.

3.1.2 Beyond textual features

Although stance is fundamentally different from sentiment [59], sentiment information can still be leveraged to augment a stance detection model. [60] proposed a neural model

to jointly learn sentiment and stance data. Sentiment information is fed to the stance classifier through an auxiliary LSTM layer shared between both stance and sentiment data.

[61] ran a battery of experiments on different stance targets to analyze four types of possible factors for stance detection on social media, particularly on Twitter:

- TXT: text-based factors such as word and character n-grams of the tweet text.
- IN: Interaction network of the tweet's author, which includes the list of users the author retweets, replies to, or mentions, as well as the web domains of the author's tweets. These features are collected from the *Home* timeline[2].
- PN: Preference network: IN features extracted from the *Likes* timeline; the tweets the user likes.
- CN: Connection network: the list of the followers and friends of the user.

The authors fed their features to a support vector machine (SVM) classifier [62] and evaluated their model on the SamEval dataset. Interestingly, connection network features captured stance information. In other words, even if a user does not tweet at all, the user's connections is sufficient to detect the user's stance with a reasonable performance (i.e. macro average of F1 over 5 stance targets is 71.39).

3.1.3 User stance detection

Social media platforms can simulate a social network by creating communities of like-minded users. The connections between users can be based on what they talk about, how they express their stances, or who they follow or like. [63] provided a sequential community-based stance detection SCSD model that propagates the known stances of a set of seed users to their significantly similar users.

[2] `https://twitter.com/home`

User-generated text on social media can differ arguably from rather more formal platforms, such as newspapers, or academic articles. The characteristics of social media text, such as adjectives frequency, adverbs frequency, or verb frequency, can potentially carry stance-related signals. [64] proposed a text representation based on twenty-one different text characteristics and stance features, such as contrariety, hypotheticality, and necessity constructions. They projected their stance-related text characteristics vectors onto a two-dimensional space with PCA. The projected vectors were then organized into six clusters. Each cluster represent a different style of writing, since the clustering is based on stance-related characteristics features. Each cluster is associated with the corresponding stance category, providing an unsupervised approach for stance detection. Defining the similarity between users can be a subject to the specific features provided by the social media platform. [65] studied the user similarity feature space for stance detection. They experimented with various Twitter-based features; retweets, hashtags, mentions, and web domains. Users are projected onto a lower dimensional user similarity space prior to classification [65].

Unsupervised Approaches The idea of projecting user vectors onto a lower dimensional space was reintroduced later in [66]. They projected user vectors using UMAP and clustering the projected vectors using mean-shift algorithm. However, the flat nature of mean-shift clustering fail to capture the latent hierarchical structure of stances. Moreover, these models suffer from rigidity as they are grounded on specific social media platforms features. Other methods for user stance detection include collective classification [67], where users in a network are jointly labeled. In this work, we use semantic features, which makes our framework applicable to a much wider range of social media platforms. Furthermore, we use density-based hierarchical clustering, which we found to perform better on semantic features than mean-shift and other clustering algorithms.

To sum up, Stance detection can be performed using supervised classification with a

variety of features such as text-level features (e.g., words or hashtags), user-interaction features (e.g., user mentions and retweets), and profile-level features (e.g., name and location) [68, 69, 70]. The use of retweets seems to yield competitive results [69, 71, 72]. Label propagation is also an effective semi-supervised method that propagates labels in a network based on user similarity, which can be simulated by follow or retweet relationships [68, 73] or the sharing of identical tweets [69, 74, 75].

The novelty of our approach is in mapping the hierarchy of stances, in unsupervised manner, using sentence embeddings pre-trained on unlabeled corpus, and providing a framework for extensive quantitative and qualitative analysis of the results.

3.2 Polarization on social media

In this section we summarize important and recent works on the study of polarization on social media. Furthermore, we review various context and approaches of polarization studies.

3.2.1 Communal polarization

Social media is a fertile ground for polarization, due to two social phenomena, namely: homophily (see 2.1.1), which is the tendency of similar users to congregate together, and biased-assimilation 2.1.2, where individuals readily accept evidence confirming their group's view, but are rather critical when provided with disconfirming evidence. Both phenomena are amplified on social network platforms [14, 10, 76]. Moreover, echo chambers (see 2.1.2) have been observed in various social media platforms [77]. Online social networks facilitate the discovery of and communication between like-minded users, and hence the creation of large homophilous communities [76]. Biased-assimilation has been shown to play a crucial role in the dynamics of polarization, as it makes community members more entrenched in their views, particularly for controversial topics [14].

The dynamics of *intra-community* and *inter-community* interactions provide predictive information about potential conflicts [73]. [78] introduced a method that employs graph embeddings, where the graph captures user interactions on Reddit[3], in conjunction with user, community, and text features to predict potential conflict and subsequent community mobilization. Unlike Twitter, the pseudo-anonymous nature of Reddit users may affect the types of interactions between communities. [79] analyzed discussions on Reddit within and across homogeneous communication spaces. The similarity of stances of a group of individuals can be reflected on the semantic difference of the same contexts. [79] used semantic difference to analyze interaction patterns in and between homogeneous groups. They quantified semantic difference as the cosine distance between words' vector representation between different but aligned word embeddings trained on each groups' vocabulary. While [79] used semantic distance to analyze communication between stance-labeled polarized groups, we use semantic similarity to detect the stances in a polarized space.

3.2.2 Semantic based polarization

Semantic features, word embeddings in particular, were shown to capture implicit human biases [80]. Several studies have trained word embeddings, based on the content generated by different communities, to contrast them through the usage of similar words and words associated with different concepts [81, 82]. [82] trained a polarity classification model using word embeddings with a seed lexicon of polarity-labeled words. Whereas, [81] trained temporal word embeddings that span 100 years to measure shifts in racial and gender attitudes. They also correlated key concepts with positive and negative adjectives over time. Moreover, [83] used word2vec embeddings and clustering metrics to quantify the polarization. However In this work, we propose the usage of sentence embeddings instead of word embeddings to capture more contextual information from the stance-bearing user generated text. Sentence embeddings have been shown to

[3] `https://www.reddit.com/`

perform significantly better than aggregated word embeddings in document semantic similarity tasks [34]. Dynamic transformer-based sentence encoders focuses on word sub-segments, which is typically useful in modelling agglutinative and morphologically rich languages, such as Turkish.

3.2.3 Quantifying polarization

Other work has focused on quantifying polarization [11, 84, 85, 86]. Several methods were used including random graph walks, network betweenness, distances in embedding spaces for different groups [85], inter-group and intra-group distances [84], and popularity of boundary nodes between communities [11]. We use the graph-based random walk controversy (RWC)[85] measure to quantify the polarization between stance-bearing users. RWC is defined as:

> "Consider two random walks, one ending in partition X and one ending in partition Y, RWC is the difference of the probabilities of two events: (i) both random walks started from the partition they ended in and (ii) both random walks started in a partition other than the one they ended in."

The similarity between users is taken from the UMAP projection of user vectors onto a two-dimensional space. Therefore, the degree of polarization in our context models the polarization between the semantic and interaction features of users, as a proxy to the polarization between their stances. Given polarized communities, several studies looked at identifying distinguishing features, such as hashtags, between such communities. One method uses the so-called valence score[12] that measures the relative probability of a feature appearing in one community compared to another [12, 74, 73]. More recent frameworks focus on mining the phrases that best summarize user stances through modelling topic-viewpoint relations[87]. We use also use the valence score to extract the top terms distinguishing polarized groups from each other. This is particularly useful as the features used to assign polarized labels to users are semantic based; top distinguishing

terms reflect the underlying factors of our stance detection model.

In conclusion, previous stance detection works have mainly been supervised approaches. Platform-specific features, such as retweets, have been shown to yield consistently competitive results. The best unsupervised user stance detection to our knowledge [66] utilized these platform-specific features to cluster users. This in turn, made it possible to label the stance a few number of users to predict the stances of the rest. This work outperforms previous unsupervised approaches without relying on platform-specific features. We employ our approach to analyze the polarization of a case study in the sports domain. Our polarization analysis is based on previous works of quantifying the polarization and qualitatively inspecting the semantic divergence between clusters based on the top distinguishing terms of each cluster.

CHAPTER IV

METHODOLOGY

In this chapter we dissect the building blocks of our framework, the underlying algorithms, and the overall pipeline. We aim to perform unsupervised fine-grained user-level stance detection based on the content of user tweets. Specifically, our proposed method represents users in an embeddings space using their topically-relevant tweets, projects user representations to a lower dimensional space, and then clusters users. Our pipeline consists of three main blocks; *Data*, *Features*, and *Model*. In the *Data* block, we explain the steps we take to collect and pre-process the data used in our framework. The *Features* block is about processing the data to extract the necessary features for our model. The output of the *Features* block is the vector representation of the users to be fed to the *Model* block. The third and final block is the *Model* block, where we project then cluster user vectors to detect their stances. The flow of the pipeline is illustrated in **Figure 11**

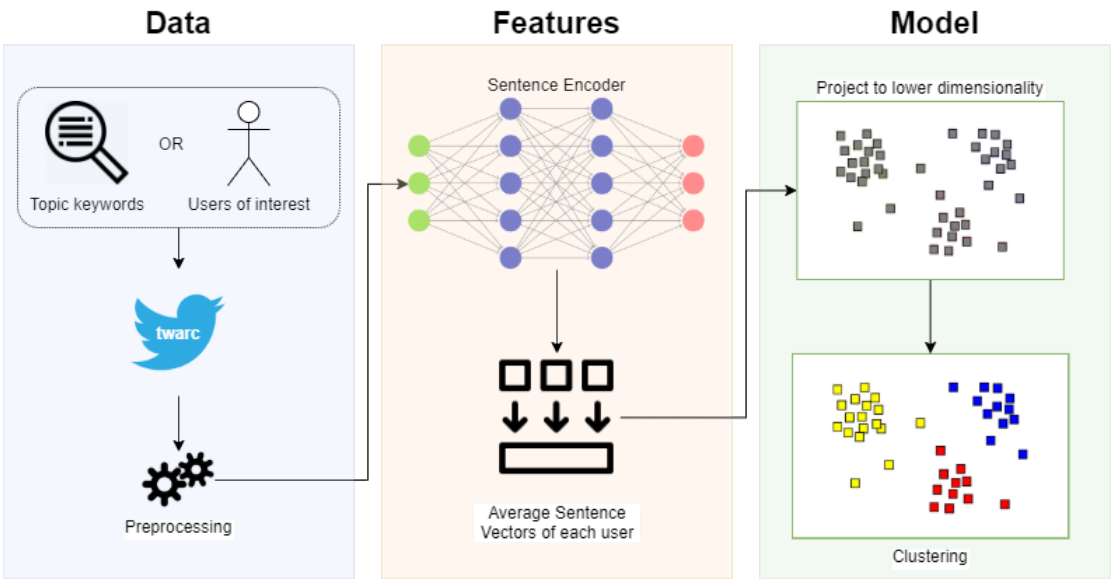


Figure 11: The pipeline of the framework for a specific topic

4.1 Data

In this section we explain how the data is collected. Furthermore, we explain the steps taken to pre-process the data before it is fed for the next block in the pipeline.

4.1.1 Data Collection

Twitter allows the free collection of tweets using keyword-based or user-based filters within a window of nearly one week. We used a mixed list of usernames and keywords pertaining to the topic of interest to collect the tweets needed for our analysis. We utilized the command line tool of twarc[1] to collect the tweets during a specific time period.

4.1.2 Data Pre-processing

In this subsection we elaborate on the way the raw data fed to our framework is processed before passed to the next analysis blocks.

Data Cleaning Due to the informal nature of Twitter, tweets commonly have grammatical and spelling errors. We filter out tweets that have less than three tokens to put more emphasis on the usability of semantic features. We also filter out users that has less than three unique tweets, to eliminate rare and outlier patterns. Furthermore, Twitter users frequently use emojis, emoticons, hashtags, media links, and other non-alphabetic characters. We used the `tweet-preprocessor` [2] Python library to remove links, and user mentions from the tweets. Overall, we performed the following pre-processing steps:

– Converting camel-cased hashtags to space-separated phrase.

– Removal of all links, and user mentions.

[1] `https://github.com/DocNow/twarc`

[2] `https://pypi.org/project/tweet-preprocessor/`

– Removal of all non-letter characters and punctuation.

– Replacement of all numbers to the word “number”.

– Case folding, where we lowercase all letters.

Bots filter In order to filter out unauthentic accounts, such as bots and fake accounts, we checked the account status of the users in our data after a few months from the original collection date. We assumed that if an account has been suspended, that implies that this account was a bot, and therefore the tweets generated by that bot can be misleading. Twitter’s usage policy suspends fake and bot accounts. Therefore, we considered only the accounts that have not been suspended by the time we checked the account status of the users.

4.2 Features

In this section we explain the steps of processing the data into feature vectors to be fed to our model. Given a target of interest, such as the name of a club or the name of a player, we filter user tweets to obtain all tweets that mention the target. Next, we pass all the filtered tweets to a sentence encoder to obtain vector representations of each. To represent individual users, we take the average of all the vectors of the filtered tweets that were posted by the user. The novelty of this work focuses on the usability of sentence embeddings for user representation in a stance detection framework.

4.2.1 Encoding

Filtered and pre-processed tweets are encoded using a sentence embedding model pre-trained on unlabeled corpus, such as, Google’s Universal Sentence Encoder [34], and Sentence-BERT [35]. Although using transfer learning let the framework work on a relatively smaller amount of data, a pre-trained sentence embedding model might not be available. In this case, if the data is sufficiently large, an off-the-shelf sentence

embedding model is trained on the data itself. Alternatively, it is typically easier to obtain an unlabelled corpus for training a sentence embedding mode. We noticed that both of the following sentence encoders produced similar results.

Universal Sentence Encoder We use Google's Universal Sentence Encoder (USE) with pre-trained embeddings [34] to represent tweets. The embedding was trained on a variety of English data sources and natural language understanding tasks. The text was tokenized using SentencePiece (a.k.a. BP), which produces sub-word tokens [88]. USE takes a sentence as an input sequence, tokenizes it, and produces a 512-dimensional output vector. USE has been shown to produce competitive results for a variety of natural language processing and retrieval tasks [34][3].

4.2.2 Averaging

The vectors of a user's tweets about a specific topic are averaged together to generate a single vector for each user. Averaging multiple tweets' vectors highlights the consistent semantic patterns in a user's tweets. Therefore, a user's vector embed the semantic information of the tweets authored by that user.

4.3 Model

This subsection demonstrates the process of detecting the stance of the users on twitter using our framework. Our model projects user representations to a lower dimensional space, and then clusters users.

[3] `https://tfhub.dev/google/universal-sentence-encoder/4`

4.3.1 Projection

We project each user vector onto a two-dimensional plane using Uniform Manifold Approximation and Projection (UMAP) algorithm [6][4]. UMAP attempts to project the data elements in a manner that reflects the similarity between them, such that more similar elements are placed closer together and less similar elements are placed further apart. UMAP is more computationally efficient than other projection techniques, such as the Force Directed graph drawing technique [89] and t-distributed Stochastic Neighbor Embedding (t-SNE) [38] and generally produces better projections [6]. Moreover, using UMAP for projection rearranges the topological structure of the data as to pre-process elements for clustering with a controllable level of granularity.

Parameters The `min_dist` and `n_neighbors` parameters of UMAP provide a level of control over how much of the global structure or the finer local densities in the data to preserve. User vectors are projected onto a two-dimensional euclidean space using UMAP algorithm. We use cosine distance as the main distance metric throughout the projection process and the clustering analysis. While similar documents can have a large euclidean distance due to the size of the documents, cosine distance can still capture the similar orientation of the documents' vectors. Thus, cosine distances are more reliable for semantic similarity tasks than euclidean distance. We found that the best performing combination of UMAP parameters for our model is a low `min_dist` (e.g. 0.0–0.25) and a relatively high `n_neighbors` (e.g. 80–100).

4.3.2 Clustering

We cluster the projected user vectors using hierarchical density based clustering (HDBSCAN) [90], which finds clusters of varying densities [5]. We use HDBSCAN algorithm to generate a condensed tree of clusters. HDBSCAN is controlled by the following

[4] `https://umap-learn.readthedocs.io/en/latest/`

[5] We used the hdbscan library at `https://pypi.org/project/hdbscan/`

parameters:

- *min_cluster_size:* the minimum number of grouped elements to be considered a cluster.
- *min_samples:* the minimum number of neighbors an element has to have to be considered a core point.

Parameters We opted to let UMAP pack elements less tightly and put slightly more emphasis on the global structure of the data, which in turn reflects the presumed polarization in the data if present. This setting for UMAP, while capturing more of the global structure of the data, can generate noise in the organization of local neighbors. This issue is handled by HDBSCAN's `min_samples` and `min_sample_size` parameters, which are set dynamically as functions of N; the number of elements to be clustered. For our approach, we formulate the functions for `min_samples` and `min_cluster_size` as:

$$min_samples = max(\sigma, \quad \left\lfloor \frac{N}{10 \times \omega} \right\rfloor)$$

$$min_cluster_size = max(\sigma, \quad \left\lfloor \frac{N}{\omega} \right\rfloor)$$

The smaller the value of `min_samples`, the less dense the clusters will be, potentially rendering less points as noise. Since we are interested in finding polarized stances in a filtered set of documents about a certain topic, we opt to use a large `min_cluster_size` to undermine anomalous semantic patterns in the polarized space. The nature of this approach leverages the presumed polarization in the data. Otherwise small datasets, i.e. less than or equal σ users, will result in a single large cluster. For our experiments, we set $\sigma = 10, \omega = 100$. The formulation of and the values used in these equations are decided empirically to generate the desired meaningful clusters of semantically polarized stances.

The condensed cluster tree generated by HDBSCAN is then flattened and summarized

so that each user is assigned a single cluster index. The condensed cluster tree, however, is preserved for further qualitative analysis of the cross-stance relations driving the polarization structure in the data.

Labelling clusters Given a set of stance labels for the clustered users, each cluster is assigned the most common label in that cluster.

When is projecting user vectors better? It can be argued that the high dimensional vector of a user, obtained by averaging the sentence embeddings of the user's tweets, can be directly used for the rest of the stance analysis. However, clustering is typically less effective and less efficient in high-dimensional spaces. clustering in high dimensional spaces suffers from the curse of dimensionality [91], and the distances between points, whether similar or dissimilar, begin to converge, which adversely affects clustering [92]. Moreover, transformer models need to have high dimensions when trained on huge datasets. This high dimensionality makes for reliable language models suitable for fine-tuning for a variety of downstream tasks. This leaves us with the motivation to project high-dimensional sentence encoders onto lower dimensional vectors suitable for clustering. We validated this approach by conducting side experiments, where we tried to cluster users without projection, and the clustering results were practically meaningless.

CHAPTER V

EXPERIMENTS

We validated the efficacy of our approach by comparing it to existing supervised and unsupervised methods on an annotated dataset. Moreover, we analyzed the various components of our approach. In this chapter, we provide the details of our battery of experiments.

5.1 Datasets

In this section we describe the datasets we used in our experiments. Our case study is focused on sports. However, using a different textual domain provides a stronger baseline to evaluate our approach. Therefore, we validated our approach on two different datasets, each in a different domain.

5.1.1 UEFA Dataset

We collected $1,834,738$ tweets about the UEFA super cup football showdown between Liverpool FC and Chelsea FC, between the 8^{th} and 17^{th} of August 2019 [1]. To filter out bots, we checked the account status of the users on the 19^{th} of March 2020. Accounts that were suspended by March 2020 were filtered out of the dataset. We ran our preprocessing pipeline on the dataset to ensure that we filtered out tweets that have less than 3 tokens, and users with less than 3 unique tweets. We labeled a set of users as Liverpool (LFC) or Chelsea (CFC) fans by matching stance-indicative keywords in users' bios. The Liverpool's motto, `YNWA`[2], as well as the keywords, `Liverpool FC`

[1] See Appendix A.1 for the complete list of keywords used in the data collection.
[2] *You Will Never Walk Alone.*

and #LFC, were used to label users as Liverpool fans. Whereas, the Chelsea's motto KTBFFH[3], as well as the keywords, Chelsea FC and #CFC were used to label users as Chlesea fans. Table 2 summarizes the UEFA dataset[4].

Target	Keywords	Users	Tweets	Unique Tweets
CFC	KTBFFH, Chelsea FC, #CFC, Chelsea fan	3,821	46,369	17,672
LFC	YNWA, Liverpool FC, #LFC, Liverpool fan	8,141	99,582	29,192
	Total	11,962	145,978	46,213
	Other	24,694	166,678	166,678

Table 2: UEFA dataset.

5.1.2 Trump Dataset

We used the labeled users in the Trump dataset (English) from [66] as a benchmark dataset. The Trump dataset has $13,731$ users, with $7,421$ and $6,310$ labeled as *PRO* and *ANTI* Trump respectively. They use stance-indicative hashtags in the account descriptions of the users. The hashtag #MAGA[5] was used to label *PRO* Trump users. Whereas, any of the hashtags #resist, #resistance, #impeachTrump, #theResistance, or #neverTrump was enough to label a user as *ANTI* Trump. The Trump dataset was collected over a period of three days (Oct. 25-27, 2018) focusing on the midterm elections. We ran our pre-processing and filtering pipeline on the dataset. The final dataset is demonstrated in table 3.

Target	Keywords	Users	Tweets	Unique Tweets
PRO	#MAGA	5,803	339,886	22,819
ANTI	#resist, #resistance, #impeachTrump, #theResistance, #neverTrump	4,788	293,449	25,749
	Total	10,591	311,335	43,836

Table 3: Trump dataset.

[3] *Keep The Blue Flag Flying High.*

[4] There are some tweets used by both CFC and LFC groups. Therefore, Total (unique) tweets is **not** simply the summation of CFC and LFC (unique) tweets.

[5] *Make America Great Again*.

5.2 Validation

In this section we show that our approach performs better than existing comparable methods with far less restraining assumptions.

5.2.1 Baselines

We compared our method to supervised and unsupervised setups. Supervised methods provide a solid baseline with a strong assumption; availability of labels. Whereas, unsupervised methods provide a more comparable baseline to our approach.

Supervised For the supervised method, we represented each user using a vector of all unique accounts that the user retweeted. Then, we used a Support Vector Machine (SVM) classifier to perform stance classification.[6] This is consistent with the best setup used by [69] and [93]. We ran two setups of this supervised method. In the first (denoted as SVM_{200}), we used the set of the 200 most vocal users with each label (ones with the most tweets) for training and the remaining users for testing. This is to emulate a typical situation where a set of users is manually annotated. In the second (SVM_{xval}), we used five-fold cross validation on all the tagged users. Though the availability of hundreds or thousands of accounts is not very common in practice, this would likely give us an upper bound on the effectiveness of this method.

Unsupervised For the unsupervised method (denoted as *UnSup*), we used the method proposed by [66], which represents each user using a vector composed of all unique accounts that the user retweeted, projects the users onto a two dimensional space using UMAP, and clusters using mean-shift clustering. Unlike our approach *UnSup* requires a strong restriction of minimum unique retweets, which filters out a chunk of users leading to a potential drop in recall. Therefore, we used a relaxed filter of minimum

[6] `scikit-learn.org/stable/modules/generated/sklearn.svm.SVC`

1 retweet per user for the *UnSup* method. Since unsupervised methods essentially perform clustering, they could potentially produce an arbitrary number of clusters and not necessarily 2 clusters. Hence, we assigned a label to each cluster based on the majority class in that cluster. For example, if the majority of users in a cluster were anti-Trump, then all users in that cluster are predicted anti-Trump. We opted not to force the algorithm to produce exactly two clusters not to lose any fine diversity in the stance of the same group, that may have been missed in our annotation. For example, the annotations are simply PRO or ANTI Trump, which does not account for party affiliation or ideological tendencies of the users. Higher diversity in stances translates into finer and more clusters [94].

5.2.2 Results

Table 4 compares the results of our method with the baselines using precision ($\mathcal{P}$), recall ($\mathcal{R}$), and F1 measures ($\mathcal{F}$). The results show that using more training data for the SVM classifier (for SVM_{xval}) improves results over using less data (SVM_{200}). We grid-searched [7] the parameter space of SVM for the best kernel and regularization parameter `C`. Linear kernel with `C` of 0.1 scored the highest on the Trump dataset. Whereas, radial basis function (RBF) kernel with `C` of 3 achieved highest micro F1 on the UEFA dataset. The *UnSup* method generally yielded high precision that is often higher than using the supervised methods; however, it led to much lower recall. Our method, where users are represented using USE embedding vectors, was competitive with the baseline supervised methods with the distinct advantage of being completely unsupervised. When compared to *UnSup*, though our method led to lower precision, it led to higher recall, leading to a comparable overall average F1-score. Given the results, we can see that our method has several advantages, namely: it is unsupervised, and it does not rely on any Twitter specific features such as retweets.

[7] `scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

	Models	Trump			UEFA		
		$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PRO-LFC	SVM_{200}	0.96	0.87	0.91	1.0	0.34	0.5
	SVM_{xval}	0.96	0.95	0.95	0.92	1.0	0.95
	UnSup	0.95	0.75	0.84	0.98	0.89	0.93
	Ours	0.91	0.89	0.9	0.99	0.98	0.98
ANTI-CFC	SVM_{200}	0.86	0.95	0.91	0.41	1.0	0.58
	SVM_{xval}	0.94	0.95	0.94	0.99	0.81	0.89
	UnSup	0.93	0.75	0.83	0.98	0.83	0.9
	Ours	0.88	0.88	0.88	0.95	0.97	0.96
AVG	SVM_{200}	0.91	0.91	0.91	0.7	0.67	0.54
	SVM_{xval}	**0.95**	**0.95**	**0.95**	0.95	0.9	0.92
	UnSup	0.94	0.75	0.84	0.98	0.86	0.91
	Ours	0.89	0.88	0.89	**0.97**	**0.97**	**0.97**

Table 4: Benchmark results for Trump and UEFA datasets. PRO and ANTI are for Trump dataset, while LFC and CFC are for UEFA dataset.

5.3 Model Analysis

In this section, we validate the effectiveness of the various components of our approach. This section evaluates the technical limitations of our approach by inspecting the robustness of each of its components; data, features, and the stance detection model itself.

5.3.1 Data

In this subsection, we explore the robustness of our approach in different levels of data availability.

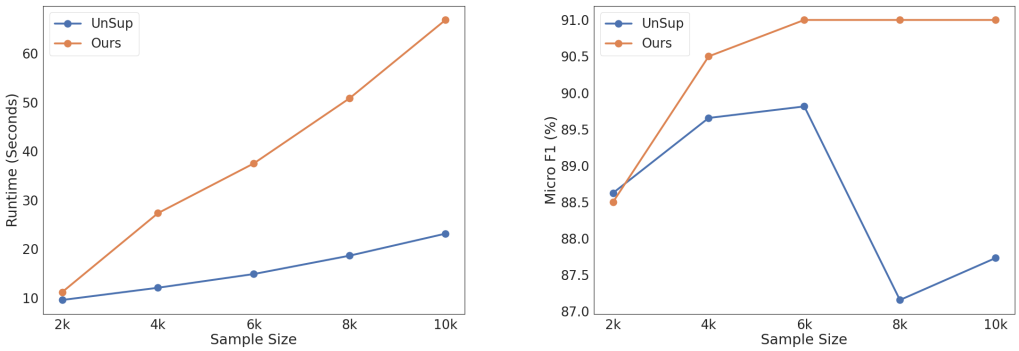
Sample size We evaluate the model with different numbers of users used in training. Table 5 shows the performance of the model with different sample sizes. We notice that our model is consistently capable of detecting stances in different sample sizes. We also report the runtime of our model with different sample sizes and compare it with previous *UnSup* method. The runtime per sample size is calculated as the average run time on both Trump and UEFA datasets. We also compare the performance of both methods, measured in micro F1 ($\mathcal{F}$) scores averaged on Trump and UEFA datasets. Figure 12

shows the efficiency and performance comparison between our approach and the *UnSup* method. We notice that *UnSup* is marginally [8] faster than our approach. This is due to the fact that the features used in *UnSup* are sparse vectors. However, this can be tolerated considering that our approach still performs generally better than *UnSup* without the reliance on platform-specific features (e.g. retweets). We also notice that *UnSup*'s performance does not monotonously improve by increasing the sample size. This could be attributed to the nature of the platform-specific features, i.e. retweets, used in UnSup. Larger sample sizes for UnSup can typically result in introducing new retweeted accounts, and expanding the one-hot representation of the users. UnSup results show that introducing more retweeted accounts to the features require a careful choice of the sample size. Moreover, UnSup filters out users with less activity. Therefore, increasing the sample size can result in filtering out more users using UnSup. Whereas, our approach depends on semantic features encoded by a sentence embedding model pretrained on a separate corpus, making it much more robust in terms of the sample size.

		Trump			**UEFA**		
	Sample Size	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PRO-LFC	2000	0.91	0.64	0.75	0.98	0.98	0.98
	4000	0.94	0.74	0.83	0.98	0.98	0.98
	6000	0.92	0.78	0.84	0.98	0.98	0.98
	8000	0.92	0.76	0.84	0.98	0.98	0.98
	10000	0.94	0.75	0.83	0.98	0.97	0.98
ANTI-CFC	2000	0.9	0.8	0.84	0.97	0.95	0.96
	4000	0.88	0.82	0.85	0.95	0.96	0.95
	6000	0.89	0.83	0.86	0.96	0.96	0.96
	8000	0.9	0.83	0.86	0.95	0.96	0.96
	10000	0.89	0.84	0.86	0.95	0.97	0.96
AVG	2000	0.9	0.72	0.8	0.97	0.97	0.97
	4000	0.91	0.78	0.84	0.97	0.97	0.97
	6000	0.9	**0.81**	0.85	0.97	0.97	0.97
	8000	0.91	0.79	0.85	0.97	0.97	0.97
	10000	**0.91**	0.8	**0.85**	0.97	0.97	0.97

Table 5: Model's performance with different sample sizes.

[8] We ran a t-test to compare the mean run time of both approaches, with a 0.03 critical value. p-value = 0.047 ¡ 0.03



(a) Runtime (b) Performance

Figure 12: Performance of our model vs UnSup with different sample sizes averaged over Trump and UEFA datasets.

Variance We assessed the model's robustness against different levels of variance in the data. Having more users with less activity (i.e. a fewer number of unique tweets) can potentially introduce more variance in the data. Therefore, we evaluate the model's performance with different settings for the filter of minimum number of unique tweets per user. Typically, if the semantic style has a stance signal, we would expect that the more tweets required to represent a user, the stronger the stance signal would be. Therefore, we expect that using a larger value of minimum tweets per user would result in a more precise stance classification (*higher precision*), but rather for a smaller set of users (*lower recall*). We experimented with three levels of minimum number of unique tweets per user; 2, 3, and 4. The resulting dataset with each of the filters is demonstrated in table 6. Table 7 shows the model's performance with different thresholds for the minimum number of unique tweets per user. We see that the model's performance generally improves upon increasing the minimum number of unique tweets per user, on the cost of filtering out more users from the analysis.

	Trump			**UEFA**		
	Users	Tweets	Unique Tweets	Users	Tweets	Unique Tweets
2	11,405	314,406	44,096	16,349	154,749	49,508
3*	10,591	311,335	43,836	11,962	145,978	46,864
4	9,628	306,676	43,434	9,389	138,260	43,238

Table 6: Resulting datasets with different thresholds for the minimum number of unique tweets per user.

		Trump			UEFA		
	Min Tweets	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PRO-LFC	2	0.93	0.69	0.79	0.98	0.97	0.97
	3*	0.91	0.89	0.9	0.99	0.98	0.98
	4	0.91	0.9	0.91	0.99	0.98	0.99
ANTI-CFC	2	0.89	0.82	0.85	0.93	0.97	0.95
	3*	0.88	0.88	0.88	0.95	0.97	0.96
	4	0.9	0.88	0.89	0.97	0.97	0.97
AVG	2	**0.91**	0.75	0.82	0.96	0.97	0.96
	3*	0.89	0.88	0.89	0.97	0.97	0.97
	4	0.9	**0.89**	**0.9**	**0.98**	**0.98**	**0.98**

Table 7: Model's performance with different minimum thresholds of the number of unique tweets per user.

5.3.2 Features

In this subsection, we analyze the effectiveness of sentence embeddings in stance classification. We experiment with different sentence embeddings.

Sentence Embeddings We evaluate the robustness of our model in terms of sentence embeddings. We test our approach with different sentence embedding models. We experimented with the sentence transformer variation of robustly optimized BERT (SBERT) [9] [35]. Their BERT model was first fine-tuned on the natural language inference dataset (AllNLI) and then on the train set of the STS dataset. Each tweet is represented by a 768-dimensional vector using SBERT. We also experimented with Google's Multilingual Universal Sentence Encoder (MUSE) with pre-trained CNN embeddings [95]. MUSE embeddings were trained on a large multilingual corpus containing text in 16 languages, including eastern languages like Turkish, and Arabic. MUSE has been shown to produce competitive results for a variety of natural language processing and retrieval tasks [95][10]. Table 8 shows the performance of our model with three different sentence embedding models. We notice that USE models perform significantly better than SBERT. This could be attributed to the fact that USE models were

[9] https://github.com/UKPLab/sentence-transformers

[10] https://tfhub.dev/google/universal-sentence-encoder-multilingual/3

trained simultaneously on various tasks. This puts SBERT at a disadvantage when it comes to generalization. Upon experimenting with different parameters for UMAP, SBERT was significantly less consistent in its performance than USE models. Moreover, using a multilingual sentence encoder (i.e. MUSE) did not compromise much of the performance as opposed to its monolingual counterpart (i.e. USE). This suggests that our approach can be applied to different datasets with different languages using a single multilingual sentence encoder.

		Trump			**UEFA**		
	Encoder	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PRO-LFC	SBERT	0.88	0.61	0.72	0.98	0.56	0.72
	MUSE	0.91	0.86	0.89	0.98	0.94	0.96
	USE*	0.91	0.89	0.9	0.99	0.98	0.98
ANTI-CFC	SBERT	0.83	0.71	0.77	0.78	0.71	0.74
	MUSE	0.89	0.86	0.87	0.9	0.95	0.92
	USE*	0.88	0.88	0.88	0.95	0.97	0.96
AVG	SBERT	0.85	0.66	0.74	0.88	0.64	0.73
	MUSE	0.9	0.86	0.88	0.94	0.94	0.94
	USE*	**0.89**	**0.88**	**0.89**	**0.97**	**0.97**	**0.97**

Table 8: Model's performance with different sentence embedding models.

5.3.3 Model

In this subsection, we demonstrate the effectiveness of using UMAP and HDBSCAN for dimensionality reduction and clustering respectively. For dimensionality reduction, first, we experiment with different algorithms for dimensionality reduction, namely, PCA, and t-sne. Then, We analyze the robustness of UMAP by testing different hyper-parameters for vector projection method. Finally, we assess the loss of information upon projecting user vectors onto a lower dimensional space using UMAP. For clustering, we analyze the performance of our model with different clustering techniques.

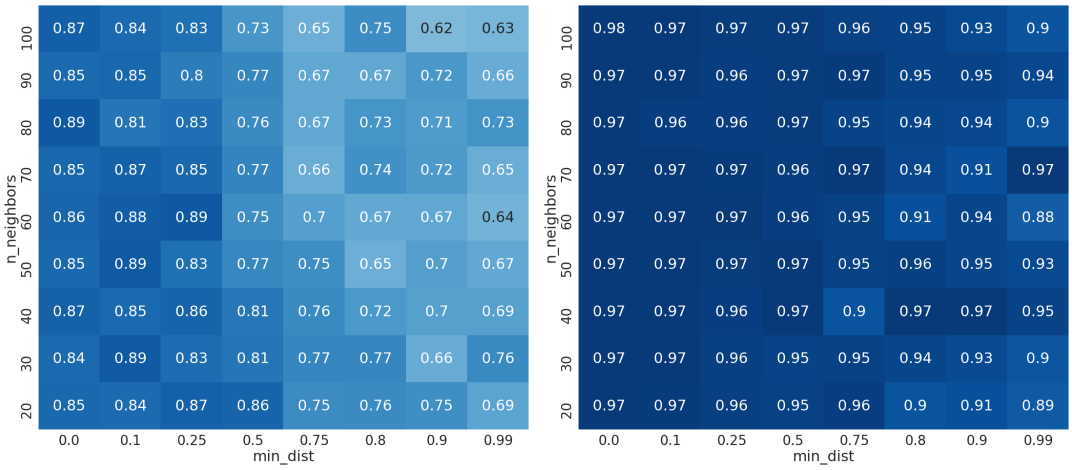
Vector projection We evaluated different dimensionality reduction methods to project user vectors. We tested UMAP against two baseline dimensionality reduction methods; PCA, and t-SNE. PCA generally captures more of the global structure, with the cost

of missing much of the local structure of high-density sub groups in the data. On the other hand, t-SNE focuses more on the dense localities in the data while missing out much of the global structure. Both of PCA and t-SNE have been a strong baseline for dimensionality reduction. Table 9 shows the performance of our model using each of the dimensionality reduction techniques. We used 2 components for all dimensionality reduction algorithms. This is to ensure comparability and maintaining the ability to visualize users. We notice that using PCA maintains a high precision while recall has plummeted significantly. This observation is consistent with ability of PCA to capture the global structure of the data without capturing finer details in the topology of the data. One major advantage of using PCA, however, is its efficiency. PCA takes on average 0.1 seconds to project user vectors in each of Trump and UEFA datasets. The results show that t-SNE performs quite well on both precision and recall. However, t-SNE took on average 150.2 seconds to project user vectors. Finally, UMAP produces the best results with moderate average runtime of 52.7 seconds per dataset.

		Trump			**UEFA**			
	Projector	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	
PRO-LFC	PCA	0.75	0.09	0.16	0.94	0.17	0.28	
	t-SNE	0.89	0.63	0.74	0.99	0.81	0.9	
	UMAP*	0.91	0.89	0.9	0.99	0.98	0.98	
ANTI-CFC	PCA	0.83	0.03	0.05	0.75	0.22	0.34	
	t-SNE	0.9	0.62	0.74	0.91	0.85	0.88	
	UMAP*	0.88	0.88	0.88	0.95	0.97	0.96	**Runtime**
AVG	PCA	0.79	0.06	0.1	0.85	0.19	0.31	**0.1**
	t-SNE	0.89	0.63	0.74	0.95	0.83	0.89	150.2
	UMAP*	**0.89**	**0.88**	**0.89**	**0.97**	**0.97**	**0.97**	52.7

Table 9: Model's performance with different dimensionality reduction techniques.

Hyper-parameters We evaluated the robustness of the model in terms of different hyper-parameters for UMAP. Precisely, we tested the framework with multiple values of `min_dist`, and `n_neighbors`. The range of `min_dist` varied from 0.0, which means any two elements can be considered neighbours in the embedding space, to 0.99, which means only elements within 0.99 cosine distance between them can be considered neighbours in the embedding space. Lower values of `min_dist` result in clumpier



(a) Trump (b) UEFA

Figure 13: Model's performance with different UMAP parameters.

embeddings, which is typically useful for clustering and polarization analysis. Whereas, `n_neighbors` is tested between from 20 to 100. Larger values of `n_neighbors` let each point have a larger neighborhood in estimating the manifold structure of the data. Larger neighborhoods around points let the UMAP embeddings focus on a more global structure of the data, potentially losing local structures and fine details. Figure 13 shows the model's performance with different UMAP hyper-parameters on both UEFA and Trump datasets. The scores shown in figure 13 are the micro average of F1 ($\mathcal{F}$) scores. We notice that the model produces consistently high performances with different UMAP hyper-parameters. We see, however, that the combination of a low value of `min_dist` (e.g. zero) and a high value of `n_neighbors` (e.g. 80 for Trump, and 100 for UEFA) produces the projection best clustered with our model. This leads us to conclude that our model is highly robust against hyper-parameters.

How much information is lost with UMAP? We evaluate the effect of dimensionality reduction on the stance information captured by sentence embeddings. First, we validate the ability of using sentence embeddings of capturing the stance of users in a supervised setting. We represent users with their corresponding embedding-based representation without UMAP projection. Unprojected vector representations were scaled

using a *MinMax* scaler [11]. The formula for *MinMax* scaler is defined as:

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \times (X_{max} - X_{min}) + X_{min}$$

The scaled user vectors are fed as features to a support vector machine classifier (SVM). Then, We used the UMAP projections of the vector representations of users as features for a support vector machine classifier (SVM). We used a radial basis function (RBF) kernel and a regularization parameter `C` of 3. We compare the performance of supervised stance classification using Universal Sentence Encoder (USE) directly, against using UMAP project the USE vectors onto a 2-dimensional space. Table 10 shows the effect of reducing the dimensionality of user vectors on the classification performance. We notice that projecting USE 512-dimensional use vectors onto a 2-dimensional space with UMAP only reduces the classification performance with a very small margin. This implies that using UMAP to project user vectors does not lose much of the stance signal in user vectors.

		Trump			**UEFA**		
	Features	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
PRO-LFC	USE	0.93	0.93	0.93	0.99	0.99	0.99
	UMAP*	0.91	0.9	0.9	0.99	0.98	0.98
ANTI-CFC	USE	0.91	0.92	0.92	0.97	0.97	0.97
	UMAP*	0.88	0.89	0.88	0.95	0.97	0.96
AVG	USE	0.92	0.92	0.92	0.98	0.98	0.98
	UMAP*	0.89	0.89	0.89	0.97	0.97	0.97

Table 10: The effect of dimensionality reduction on the model's performance.

Clustering The clustering method used in the current state-of-the-art in unsupervised user stance detection[66] used mean-shift algorithm for clustering stance-bearing users. We compared the performance of our model using HDBSCAN with using mean-shift for clustering. While both mean-shift and HDBSCAN are density-based clustering methods, only HDBSCAN is a hierarchical clustering technique. Therefore, we opted to choose HDBSCAN over mean-shift to account for the latent hierarchical structure

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html`

of stances. For example, users can be classified as pro or anti Trump, while the cluster of pro-Trump can have party-affiliated supporters as well as conservatives with no party affiliations. In this scenario, non-hierarchical clustering when forced to produce finer clusters would only detect the clusters themselves but not the connection between them; there is no clustering tree to reveal the connection between clusters of pro Trump with and without party affiliations. This gives an interpretability advantage to hierarchical clustering over non-hierarchical ones. We also report the results of HDBSCAN's predecessors, namely, DBSCAN, and OPTICS. We used the same parameters for DBCSAN, OPTICS, and HDBSCAN; `min_cluster_size` was set to 105 and 119 for Trump and UEFA datasets respectively. Whereas, `min_samples` was set to 10 and 11 for Trump and UEFA datasets respectively. We used the default parameters of mean-shift provided by scikit-learn's implementation [12]. We set DBSCAN's ϵ [13] to 0.1 to ensure that DBSCAN does not produce a single cluster. We also noticed that OPTICS worked better with cosine distance, while DBSCAN and HDBSCAN were better with euclidean distance. Table 11 shows the performance of our approach with each of the clustering algorithms. Mean-shift achieves comparably high performance with HDBSCAN, though with a significantly longer runtime. DBSCAN seems to achieve the closest to HDBSCAN in terms of efficiency and performance. However, HDBSCAN still holds the advantage of generating a hierarchical clustering tree. OPTICS seem to perform quite poorly on recall and requires significantly longer runtime than DBSCAN and HDBSCAN. These results suggest that HDBSCAN is not just best suited for hierarchical clustering of stances, but also it is very efficient.

[12] `scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html`

[13] The maximum distance between two samples for one to be considered as in the neighborhood of the other

		Trump			UEFA			
	Clusterer	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	
PRO-LFC	Mean-shift	0.9	0.9	0.9	0.99	0.98	0.98	
	DBSCAN	0.92	0.86	0.89	0.99	0.94	0.96	
	OPTICS	0.92	0.53	0.67	0.92	0.58	0.71	
	HDBSCAN*	0.91	0.89	0.9	0.99	0.98	0.98	
ANTI-CFC	Mean-shift	0.88	0.88	0.88	0.95	0.97	0.96	
	DBSCAN	0.89	0.86	0.87	0.96	0.94	0.95	
	OPTICS	0.89	0.61	0.72	0.98	0.49	0.65	
	HDBSCAN*	0.88	0.88	0.88	0.95	0.97	0.96	**Runtime**
AVG	Mean-shift	0.89	**0.89**	0.89	0.97	0.97	0.97	88.78
	DBSCAN	0.91	0.86	0.88	**0.98**	0.94	0.96	**0.08**
	OPTICS	**0.91**	0.57	0.7	0.95	0.53	0.68	39.77
	HDBSCAN*	0.89	0.88	**0.89**	0.97	**0.97**	**0.97**	0.19

Table 11: Model's performance with different clustering methods.

CHAPTER VI

ANALYSIS AND DISCUSSION

In this chapter, we showcase an application of our method. First, we utilized our method to quantitatively analyze fine-grained topics in the the UEFA dataset. Trump dataset has been previously analyzed in a similar context [66]. We analyze the mutual information between stances on the different topics. We also analyze the polarization degrees on these topics. Then, we qualitatively analyze the results of our approach, and discuss its limitations. We analyze the semantic differences in clusters across various topics. We inspect the cases where our model failed, by analyzing the tweets whose authors were misclustered. We compile a list of patterns and social phenomena most challenging for our approach to accurately classify stances.

6.1 Topics

For our analysis, we picked 4 potentially polarizing topics in the UEFA dataset; the match, the coaches, the stars, and the referee (Ref). The *match* topic highlights the polarization on the two clubs playing the UEFA final, namely, Chelsea FC (CFC) and Liverpool FC (LFC). The *coaches* topic highlights the polarization on the performance of the coaches of each club, namely, Frank Lampard for CFC, and Jürgen klopp for LFC. The *stars* topic highlights the polarization on the performance of the prominent stars of the teams, such as Mohamed Salah for LFC, and N'Golo Kanté for CFC. Finally, the *referee* topic highlights the polarization on the performance of the referee, Stéphanie Frappart, especially that she was the first female referee of a major UEFA final for men. We extracted the tweets of each topic from the UEFA dataset with indicative keywords [1].

[1] We handled slight derivatives of the words such as é in (Kante or Kanté), case folding, and spaces in (SuperCup, or Super Cup).

The *Referee* topic is not tweeted about as frequently as the other topics. To balance this effect, we relaxed the threshold for the minimum number unique tweets per user down to 2 for the *Referee* topic. The final dataset for each topic is described in table 12.

Topic	Keywords	Users	Tweets	Unique Tweets
Match	Super Cup, LivChe, CheLiv, Chelsea FC, Liverpool FC, YNWA, KTBFFH, LFC, CFC	42,095	336,373	92,687
Coaches	Lampard, Klopp	4,325	31,172	6,433
Stars	Salah, Mane, Firmino, Henderson, Van Dijk, Adrian, Kante, Tammy, Jorginho, Zouma, Pedro, Arrizabalaga	4,589	24,966	6,698
Referee	Referee, Frappart, VAR	1,170	2,806	774
	Total	52,179	395,317	106,592

Table 12: UEFA topics used in the analysis.

6.2 Quantitative Analysis

In this section we show how we utilized our method to quantitatively analyze fine-grained topics in the UEFA dataset. We address the following questions:

– Can we infer the stance on a topic from a different topic?

– What does the polarization degree inform us on different topics?

6.2.1 Stance Alignment

One of the symptoms and reinforcing causes of polarization is "biased assimilation", where individuals readily accept confirming evidence and are rather critical when provided with disconfirming evidence [14]. We inspected if this phenomenon implies that there is correlation between clusters on different topics. In this subsection, we test our approach to analyze the mutual information between stances on different topics in the UEFA dataset. We are interested in analyzing what the stance towards a target a tells us about the stance towards a different stance target b. In other words, can a stance towards a topic have a signal about the stance towards a different topic?

Methodology For each topic, users sharing semantically similar stances towards that topic are clustered together. Given the cluster labels for each user for each topic, we perform a mutual information analysis to quantify the relation between stance-bearing topics. We relaxed the `n_neighbors` value to 20, to let the model detect finer details in the data structures. This ensures that local structures are not lost in few large clusters. We use normalized mutual information adjusted for randomness (AMI) [2] [96] to assign an information mutuality score for each pair of stance-bearing topics. The mutual information between topics is then visualized in a heat map. The formula for mutual information (MI) between two clustering solutions is:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} log \frac{N|U_i \cap V_j|}{|U_i||V_j|}$$

MI is generally higher for two clustering with a larger number of clusters. Therefore, we use the MI version adjusted for chance (AMI), which is defined as:

$$AMI(U,V) = \frac{MI(U,V) - E(MI(U,V))}{avg(H(U),H(V)) - E(MI(U,V))}$$

Can we infer the stance on a topic from a different topic? Figure 14 shows how the stances on different topics can be inter-dependant. We see that users are clustered similarly on the *Match* topic, the *Coaches* topic, and the *Stars* topic. Whereas, the clustering on the *Referee* topic is relatively different from the other topics. This suggests that the stances towards the clubs are dependent on the stances towards the coaches and the stars, rather than the referee. This implies that the performance of the referee did not stir significant controversy between the polarized clusters. This can also be partially due to the relatively smaller size of the dataset of the *Referee* topic.

6.2.2 Polarization Quantification

We analyzed the polarization degrees across different topics. This analysis is particularly useful to stratify the polarizing topics within a community.

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html`

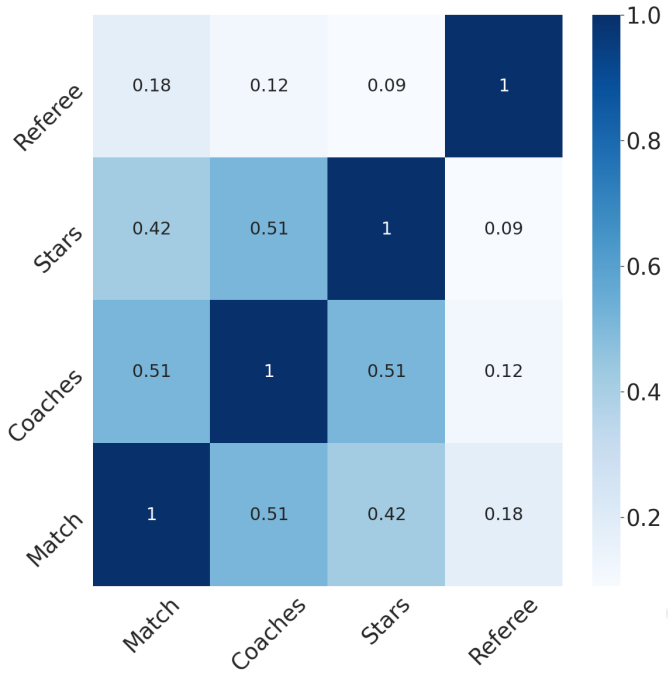


Figure 14: Inter-dependence of stances on different topics measured in (AMI).

Methodology We computed the polarization between the user clusters using Random Walk Controversy (RWC) measure [86, 85]. Given a graph of connected users, where the nodes denote users and the weights of the edges denote the user cosine similarities, computed based on retweets, RWC computes the shortest graph traversal from a set of random users to prominent users either with the same or different stances, where prominent nodes are the top n nodes with the most connections in a community [86]. Two nodes are considered connected if their cosine similarity is not zero. The score is formulated as: $RWC = P_{AA}P_{BB} - P_{AB}P_{BA}$, where A and B are different classes and P_{XY} is the probability that a random node in X would reach a highly connected node in Y.

What does the polarization degree inform us on different topics? We computed RWC polarization measure on the aforementioned 4 topics. We used a minimum cosine similarity of 0.5. Unlike [86] who are using retweets, we are using user vectors obtained from by encoding the tweets they authored. Due to the large size of the *Match* dataset, we calculated the RWC for 5 different samples of 5000 users, and reported the average.

Table 13 shows the RWC values, which range between 0 and 1, where 0 implies no polarization and 1 implies extreme polarization. We notice that there is a remarkably higher polarization on the *Coaches* topic.This can be due to the binary nature of the topic; each cluster is talking mainly about their own coach. Whereas, we see lower polarization degrees on the more inclusive topic of *Stars*. This suggests, that users from both groups talk about the stars of other teams with a certain level of semantic similarity. The moderate polarization on the *Referee* topic can be attributed to the small size of the dataset. The smaller set of users aggregates the divergence between the styles present in the dataset. Whereas, the larger set of users, typically in the *Match* topic, gives room for a variety of semantic patterns, thus undermining the polarization between CFC and LFC fans on the topic. These results suggest a future work of further analysis of the correlation between the polarization degrees and the semantic divergence between different clusters.

Topic	**RWC**
Match	0.62
Coaches	0.95
Stars	0.78
Referee	0.73

Table 13: RWC polarization measure across topics.

6.3 Qualitative analysis

In this section, we utilized our method to qualitatively analyse the clusters in the UEFA dataset across various topics. We assessed the results against social theories and analyzed the semantic differences in clusters. We inspected the limitations of our model and compiled a list of semantic patterns and social phenomena where our model is likely to fail.

6.3.1 Semantic Difference in Clusters

It has been shown that polarized groups show semantic divergence [79]. For our analysis, we identify the most prominent terms in each cluster to show how people talk about the same issue in different contexts. We elected the analysis of most prominent terms as we are using semantic features to represent users for clustering. Furthermore, we inspected the influence of the context of polarizing events on the semantic style of stance expression.

Methodology We assign *prominence* scores for terms using valence scores [12] and term frequencies. The valence score of a term in a set of tweets D_a captures the degree of its occurrence in D_a compared to another set D_b. We define the prominence score of a term in a set of tweets as the product of its valence score and its term frequency as follows:

$$\Pr(t, D_a, D_b) = \log\left(tf_{t,D_a}\right) \times \left(2\frac{\frac{tf_{t,D_a}}{|D_a|}}{\frac{tf_{t,D_a}}{|D_a|} + \frac{tf_{t,D_b}}{|D_b|}} - 1\right)$$

where t is the term of interest, D_a is a set of tweets (i.e., tweets of users in a cluster of interest), D_b is the other tweet set, tf_{t,D_a} and tf_{t,D_b} denote the term frequency of t in D_a and D_b respectively, and $|D_a|$ and $|D_b|$ are the sums of the frequencies of all the terms in D_a and D_b respectively.

Results Figure 15 shows word clouds of the most prominent terms for the UEFA topics. We notice a remarkable contrast between how users in different clusters use distinctive terms towards a target. For the *Match* topic, CFC clusters use "KTBFFH" [3] and "proud", and the LFC clusters use "YNWA" [4], and "champions". This focus of CFC fans on the satisfying performance of their team, and the LFC fans on winning the cup, highlights the divergence in the semantic style between the two groups. Different topics show more fine-grained divergence in the semantic style. For example, the *Coaches*

[3] *Keep The Blue Flag Flying High*. Motto used by CFC fans.
[4] *You Will Never Walk Alone*. Motto used by LFC fans.

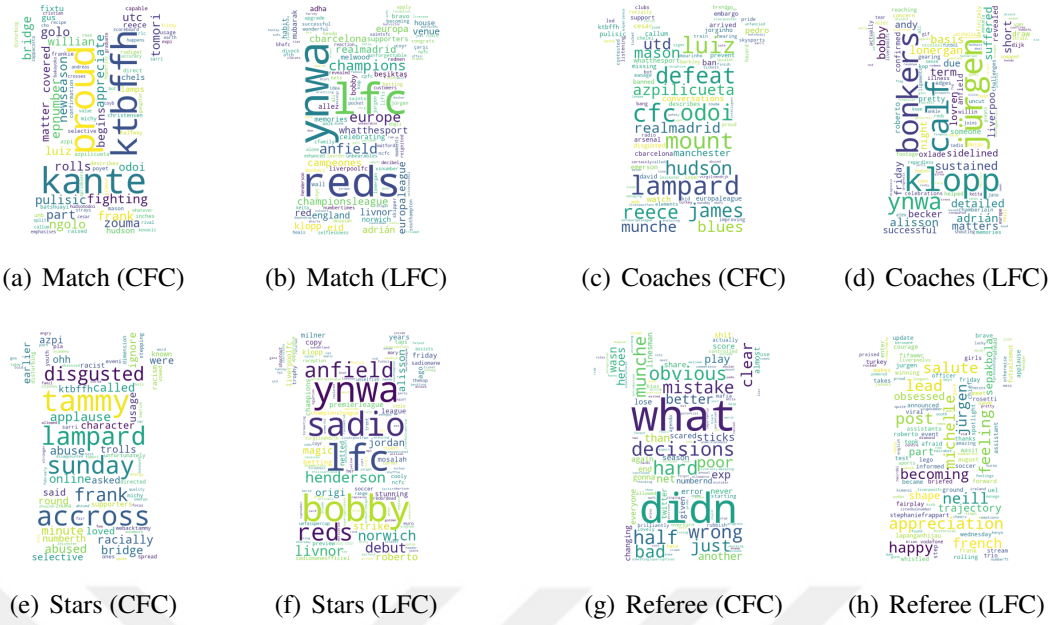
topic shows how LFC fans focus on Klopp (LFC's coach), and CFC fans focus on Lampard (CFC's coach). Interestingly, LFC fans use the terms "calf" referring to the injuries Klopp had to deal with. Whereas, CFC fans use the term "munche" referring to the game Chelsea lost against Manchester United a few days before the game. Similarly, the topic of *Stars* show how LFC fans mention their own stars more frequently (i.e. Sadio, and Bobby [5]). Whereas, CFC fans focus more on "Tammy" [6], who missed the last decisive penalty that costed Chelsea the cup. We also see the term "disgusted" referring to the reaction towards the CFC fan who racially abused Tammy Abraham after the game. Specifically, the term "disgusted" was aggregated by retweeting the official twitter account of Chelsea FC reporting Lampard's comment on the racial abuse:

> RT @ChelseaFC: Earlier today, Frank Lampard said he was '**disgusted** by the so-called Chelsea fan' that racially abused Tammy Abraham.

Finally, we find most prominent terms in the *Referee* topic less indicative of the stance. This confirms our finding from the stance alignment, that there is little to no polarization on the *Referee* topic.

Does the context of polarizing events influence the semantic style of stance expression? In the light of results of the LFC and CFC showdown, it is important to report the events related to the event. Liverpool's main goal keeper, Alisson Becker, was out of the line-up because of a calf injury. However, Liverpool's substitute goalkeeper, Adrian Castillo, secured the cup for his team by protecting his net from Tammy Abraham's decisive penalty. On the other hand, Chelsea had great hopes with their relatively new coach and iconic player Frank Lampard. However, CFC have lost in the penalties after showing an arguably better performance than LFC throughout the match. The loss of CFC came with a lot of injuries in their deck, and after a heavy loss against Manchester United FC (MUN) on the 9^{th} of the same month; a week before the showdown. The

[5] Nickname for LFC's star, Roberto Firmino
[6] Tammy Abraham



(a) Match (CFC) (b) Match (LFC) (c) Coaches (CFC) (d) Coaches (LFC)

(e) Stars (CFC) (f) Stars (LFC) (g) Referee (CFC) (h) Referee (LFC)

Figure 15: Word Clouds Generated by prominent terms in each cluster for different topics.

loss against MUN was still vividly tweeted about shortly before the showdown against Liverpool. One of the frequently retweeted tweet:

> " Thanos came to Old trafford, snapped his fingers and Chelsea's defence dissappeared[7]. "

We also notice a remarkable change in the tweeting behaviour or fans before, during, and after the match. CFC and LFC fans tweeted fairly with the same frequency before the match. However, after the match, CFC fans' tweets plummeted, while LFC fans' tweets dominated the social media trends, especially with their hashtag: #YNWA (You'll Never Walk Alone).

6.3.2 Limitation Analysis

We inspected nearly 2000 tweets of users from different clusters. We focused on analyzing the tweets of the users misclustered, or on the cluster's edge close to another cluster. We refer to LFC and CFC bases as regions in the clustering space where users

[7]The typos are preserved as the original tweet.

are clustered with LFC and CFC *labeled* fans respectively. We report a list of semantic patterns mostly challenging for our model.

Possible Reason	**Sample Tweet**
Selective Retweeting	RT @LFC: Firmino scores the first penalty of the shootout!
Cross Mentioning	I respect Liverpool and they had chances in the game but it was clear to me that we were the stronger team
Implicit Criticism	We won the Super cup with the worst performance since Klopp took over haha #LIVCHE
Post-event Teasing	@Calabastu VAR knows your penalty wasn't the right call #lfc #cfc
Pre-event Teasing	#Lampard it's only #Liverpool on Wednesday #cfc #lfc #ynwa

Table 14: Possible reasons of misclustering with examples.

Selective Retweeting We noticed that a lot of users in the CFC base belong to news and highlights pages. That reflects a media focus on the anticipated achievement of Frank Lampard, CFC's coach, with Chelsea after his reforms in the team. That is confirmed further when we found more betting and news accounts clustered towards CFC base. We believe this is an interesting question to study for future research. We found it very difficult to annotate LFC fans in the CFC base, as their stances are implicit and not expressed clearly. We usually find their retweeting pattern focus on match highlights related to LFC. The selective retweeting of the highlights related to one club was found to be the ruling tweeting behaviour in LFC and CFC bases. Furthermore, news accounts and sports highlights bots add more noise to the stance signal of highlight tweets. So if an LFC fan only retweets LFC-related highlights, and a CFC fan only retweets CFC-related highlights, a news account retweeting both LFC and CFC-related highlights results in both CFC and LFC fans clustered relatively close to each other. Therefore, users who selectively retweets highlights about one team are likely to be confused, or at least fall at the borders of stance clusters.

Cross Mentioning Users who express a positive stance towards the opposite target are likely to be misclassified by our model. For example, despite losing the match,

some CFC fans were found expressing their respect for the opposing, winning, team, thus, confusing the model into clustering them closer to the LFC base.

Implicit Criticism Although the fan bases of each of the opposing clubs expressed some criticism, each had a remarkably different target for their criticism. The LFC fans that expressed some criticism, quipped about the poor performance of their team. This caused the model to cluster some of the LFC fans, who use semantic style, closer to the CFC base.

Post-event Teasing Whereas, CFC expressed much less criticism in their tweets. This can be explained by the overall arguably better performance of CFC players in the match. Post-event teasing can take the form of objection or gloating. For example, some CFC fans steered their criticism towards the referee. Table 14 shows an example of a CFC fan mentioning an LFC fan, to tease him by objecting to the video assistant referee.

Pre-event Teasing Moreover, a different form of teasing the other team came as a remarkable highlight by some LFC fans who posted provocative tweets just before the match.

Summary Overall, we see that using ambiguous language to express a stance in a sarcastic manner is a challenge for semantic based stance clustering as it requires background knowledge. Moreover, we see that the semantic style of the tweets is a subject to the details of the event, such as the line up of the teams, the output of the match, and the pre/early-season results influence the semantic style of fans in expressing their stances. We believe the analysis of the timelines of stance-bearing users during multiple polarizing events would provide further insights in future work.

CHAPTER VII

CONCLUSION

In this work, we propose a novel unsupervised user stance detection method to capture fine-grained divergences in a community across various topics. Our method employs pre-train sentence encoders represent users based on the content of their tweets on a particular topic. Next, our method projected users onto a lower dimensional space, bringing similar users closer together and pushing dissimilar users further apart, and then clustered the users. We validated our approach on two datasets in different domains. We showed that our approach performs in bar with supervised methods, and outperforms previous unsupervised approaches. Moreover, our approach enjoys the two advantages of returning a hierarchical clustering solution of user stances, and not requiring platform-specific features.

We utilized our method to explore the polarization between Twitter users on UEFA Super Cup's 2019 final as a case study. We compiled a dataset of more than 300k tweets, and used a twitter-based heuristic to accurately tag nearly 12k users as Liverpool FC, or Chelsea FC fans. We observed that our method can be used to detect fine-grained stances of users towards different topics with high accuracy. We conduct quantitative and qualitative analysis with our method on the data we collected for the case study. We applied our method to cluster user stances towards various polarizing issues among the UEFA fans, noting correlations between positions across topics. Additionally, we used the resultant clusters to quantify polarization on topics of interest. We conclude that semantic styles reflect the polarization in stances between users. We report the weaknesses of our approach to shed light on future work.

APPENDIX A

DATA COLLECTION KEYWORDS AND DATES

A.1 UEFA Super Cup (8–17 August 2019)

- #ADR13N
- #Adrian
- #CFCFamily
- #CHELIV
- #EqualGame
- #FrankLampard
- #KTBFFH
- #Klopp
- #LFC
- #LIVCHE
- #Lampard
- #MoSalah
- #SuperCup
- #UEFASuperCup
- #YNWA
- #cfc
- #chelsea
- #chelseafc
- #frappart
- @10SadioMane
- @AdriSanMiguel
- @ChelseaFC
- @LFC
- @Pedro17
- @VirgilvDijk
- @chelseayouth
- @fikayotomori_
- @tammyabraham
- @wshjackson

Bibliography

[1] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013.

[2] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," 2013.

[3] Y. Wang, Y. Hou, W. Che, and T. Liu, "From static to dynamic word representations: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 1611–1630, 2020.

[4] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Beijing, China), pp. 1681–1691, Association for Computational Linguistics, July 2015.

[5] "Encoder of greater-than-word length text trained on a variety of data." [Online; accessed Mar 8, 2020].

[6] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2018.

[7] M. Nedrich, "Mean shift clustering," 2015. [Online; accessed Mar 1, 2020].

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online; accessed Mar 3, 2020].

[9] F. Alahmadi, "Groundwater aquifer delineation by spatial cluster modeling in madinah, western kingdom of saudi arabia," *American Journal of Geographic Information System*, vol. 8, pp. 126–130, 05 2019.

[10] M. Mcpherson, L. Smith-Lovin, and J. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, pp. 415–, 01 2001.

[11] P. Guerra, W. Meira Jr, C. Cardie, and R. Kleinberg, "A measure of polarization on social media networks based on community boundaries," *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, pp. 215–224, 01 2013.

[12] M. D. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini, "Political polarization on twitter," in *Fifth international AAAI conference on weblogs and social media*, 2011.

[13] S. Aref and Z. Neal, "Legislative effectiveness hangs in the balance: Studying balance and polarization through partitioning signed networks," *CoRR*, vol. abs/1906.01696, 2019.

[14] P. Dandekar, A. Goel, and D. T. Lee, "Biased assimilation, homophily, and the dynamics of polarization," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5791–5796, 2013.

[15] B. Hofstra, R. Corten, F. van Tubergen, and N. B. Ellison, "Sources of segregation in social networks: A novel approach using facebook," *American Sociological Review*, vol. 82, no. 3, pp. 625–656, 2017.

[16] R. S. Baron, S. I. Hoppe, C. F. Kao, B. Brunsman, B. Linneweh, and D. Rogers, "Social corroboration and opinion extremity," *Journal of Experimental Social Psychology*, vol. 32, no. 6, pp. 537 – 560, 1996.

[17] P. W. Singer and E. T. Brooking, *LikeWar: The Weaponization of Social Media*. Eamon Dolan/Houghton Mifflin Harcourt, 2018.

[18] R. Jiang, S. Chiappa, T. Lattimore, A. György, and P. Kohli, "Degenerate feedback loops in recommender systems," *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society - AIES '19*, 2019.

[19] C. Lai and M. Luczak-Roesch, "You can't see what you can't see: Experimental evidence for how much relevant information may be missed due to google's web search personalisation," in *Social Informatics* (I. Weber, K. M. Darwish, C. Wagner, E. Zagheni, L. Nelson, S. Aref, and F. Flöck, eds.), (Cham), pp. 253–266, Springer International Publishing, 2019.

[20] C. Lord, L. Ross, and M. Lepper, "Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence," *Journal of Personality and Social Psychology*, vol. 37, pp. 2098–2109, 11 1979.

[21] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, p. 1137–1155, Mar. 2003.

[22] Y. Goldberg and G. Hirst, *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017.

[23] J. R. Firth, "A synopsis of linguistic theory 1930-55.," vol. 1952-59, pp. 1–32, 1957.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[25] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[26] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[27] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, (Valencia, Spain), pp. 427–431, Association for Computational Linguistics, Apr. 2017.

[28] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *NIPS*, 2017.

[29] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[30] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[33] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[34] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," 2018.

[35] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.

[36] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, (San Francisco, CA, USA), p. 737–744, Morgan Kaufmann Publishers Inc., 1993.

[37] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," Nov. 1901.

[38] L. van der Maaten and G. Hinton, "Viualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.

[39] L. van der Maaten, "Barnes-hut-sne," 2013.

[40] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[41] A. Najjar, D. Reinharz, C. Girouard, and C. Gagné, "A two-step approach for mining patient treatment pathways in administrative healthcare databases," *Artificial Intelligence in Medicine*, vol. 87, 04 2018.

[42] J. Das, P. Mukherjee, S. Majumder, and P. Gupta, "Clustering-based recommender system using principles of voting theory," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 230–235, Nov 2014.

[43] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, pp. 32–40, January 1975.

[44] R. F. Ling, "On the theory and construction of k-clusters," *The Computer Journal*, vol. 15, pp. 326–332, 01 1972.

[45] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231, AAAI Press, 1996.

[46] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, p. 49–60, June 1999.

[47] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining* (J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, eds.), (Berlin, Heidelberg), pp. 160–172, Springer Berlin Heidelberg, 2013.

[48] D. W. Muller and G. Sawitzki, "Excess mass estimates and tests for multimodality," *Journal of the American Statistical Association*, vol. 86, no. 415, pp. 738–746, 1991.

[49] I. Augenstein, A. Vlachos, and K. Bontcheva, "USFD at SemEval-2016 task 6: Any-target stance detection on twitter with autoencoders," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, (San Diego, California), pp. 389–393, Association for Computational Linguistics, June 2016.

[50] G. Zarrella and A. Marsh, "Mitre at semeval-2016 task 6: Transfer learning for stance detection," 2016.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[52] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 876–885, Association for Computational Linguistics, Nov. 2016.

[53] C. Xu, C. Paris, S. Nepal, and R. Sparks, "Cross-target stance classification with self-attention networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 778–783, Association for Computational Linguistics, July 2018.

[54] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," 2016.

[55] P. Wei, J. Lin, and W. Mao, "Multi-target stance detection via a dynamic memory-augmented network," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, (New York, NY, USA), p. 1229–1232, Association for Computing Machinery, 2018.

[56] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014.

[57] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," 2015.

[58] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Marquez, and A. Moschitti, "Automatic stance detection using end-to-end memory networks," 2018.

[59] A. Aldayel and W. Magdy, "Assessing sentiment of the expressed stance on social media," 2019.

[60] Q. Sun, Z. Wang, S. Li, Q. Zhu, and G. Zhou, "Stance detection via sentiment information and neural network model," *Frontiers of Computer Science*, vol. 13, pp. 127–138, 2018.

[61] A. Aldayel and W. Magdy, "Your stance is exposed! analysing possible factors for stance detection on social media," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, Nov. 2019.

[62] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[63] O. Fraisier, G. Cabanac, Y. Pitarch, R. Besançon, and M. Boughanem, "Stance classification through proximity-based community detection," in *Proceedings of the 29th on Hypertext and Social Media*, HT '18, (New York, NY, USA), p. 220–228, Association for Computing Machinery, 2018.

[64] V. Simaki, P. Simakis, C. Paradis, and A. Kerren, "Detection of stance-related characteristics in social media text," in *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, SETN '18, (New York, NY, USA), Association for Computing Machinery, 2018.

[65] K. Darwish, W. Magdy, and T. Zanouda, "Improved stance prediction in a user similarity feature space," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pp. 145–148, ACM, 2017.

[66] K. Darwish, P. Stefanov, M. Aupetit, and P. Nakov, "Unsupervised user stance detection on twitter," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, pp. 141–152, May 2020.

[67] Y. Duan, F. Wei, M. Zhou, and H.-Y. Shum, "Graph-based collective classification for tweets," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 2323–2326, ACM, 2012.

[68] J. Borge-Holthoefer, W. Magdy, K. Darwish, and I. Weber, "Content and network dynamics behind egyptian political polarization on twitter," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 700–711, ACM, 2015.

[69] W. Magdy, K. Darwish, N. Abokhodair, A. Rahimi, and T. Baldwin, "# isisisnotislam or# deportallmuslims?: Predicting unspoken views," in *Proceedings of the 8th ACM Conference on Web Science*, pp. 95–106, ACM, 2016.

[70] W. Magdy, K. Darwish, and I. Weber, "# failedrevolutions: Using twitter to study the antecedents of isis support," *First Monday*, vol. 21, no. 2, 2016.

[71] F. M. F. Wong, C. W. Tan, S. Sen, and M. Chiang, "Quantifying political leaning from tweets and retweets," in *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[72] F. M. F. Wong, C. W. Tan, S. Sen, and M. Chiang, "Quantifying political leaning from tweets, retweets, and retweeters," *IEEE transactions on knowledge and data engineering*, vol. 28, no. 8, pp. 2158–2172, 2016.

[73] I. Weber, V. R. K. Garimella, and A. Batayneh, "Secular vs. islamist polarization in egypt on twitter," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 290–297, ACM, 2013.

[74] K. Darwish, "To kavanaugh or not to kavanaugh: That is the polarizing question," *arXiv preprint arXiv:1810.06687*, 2018.

[75] M. Kutlu, K. Darwish, and T. Elsayed, "Devam vs. tamam: 2018 turkish elections," *arXiv preprint arXiv:1807.06655*, 2018.

[76] R. Mousavi and B. Gu, "The effects of homophily in twitter communication network of u.s. house representatives: A dynamic network study," *SSRN Electronic Journal*, 01 2015.

[77] C. R. Sunstein, *Republic.com 2.0*. Princeton University Press, 2007.

[78] S. Kumar, W. L. Hamilton, J. Leskovec, and D. Jurafsky, "Community interaction and conflict on the web," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 933–943, International World Wide Web Conferences Steering Committee, 2018.

[79] J. An, H. Kwak, O. Posegga, and A. Jungherr, "Political discussions in homogeneous and cross-cutting communication spaces," 2019.

[80] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[81] N. Garg, L. Schiebinger, D. Jurafsky, and J. Zou, "Word embeddings quantify 100 years of gender and ethnic stereotypes," *Proceedings of the National Academy of Sciences*, vol. 115, no. 16, pp. E3635–E3644, 2018.

[82] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214–224, 2017.

[83] M. Yang, X. Wen, Y.-R. Lin, and L. Deng, "Quantifying content polarization on twitter," in *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, pp. 299–308, IEEE, 2017.

[84] A. Morales, J. Borondo, J. C. Losada, and R. M. Benito, "Measuring political polarization: Twitter shows the two sides of venezuela," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 3, p. 033114, 2015.

[85] K. Garimella, G. D. F. Morales, A. Gionis, and M. Mathioudakis, "Quantifying controversy on social media," *ACM Transactions on Social Computing*, vol. 1, no. 1, p. 3, 2018.

[86] K. Darwish, "Quantifying polarization on twitter: The kavanaugh nomination," in *International Conference on Social Informatics*, pp. 188–201, Springer, 2019.

[87] A. Trabelsi and O. R. Zaïane, "Phaitv: A phrase author interaction topic viewpoint model for the summarization of reasons expressed by polarized stances," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 13, pp. 482–492, Jul. 2019.

[88] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[89] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[90] L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, pp. 33–42, IEEE, 2017.

[91] M. Verleysen, "Learning high-dimensional data," *Nato Science Series Sub Series III Computer And Systems Sciences*, vol. 186, pp. 141–162, 2003.

[92] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pp. 217–235, Springer-Verlag, 1999.

[93] K. Darwish, W. Magdy, A. Rahimi, T. Baldwin, and N. Abokhodair, "Predicting online islamophopic behavior after# parisattacks," *The Journal of Web Science*, vol. 3, no. 1, 2017.

[94] A. Rashed, M. Kutlu, K. Darwish, T. Elsayed, and C. Bayrak, "Embeddings-based clustering for target specific stances: The case of a polarized turkey," 2020.

[95] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Multilingual universal sentence encoder for semantic retrieval," 2019.

[96] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, p. 2837–2854, Dec. 2010.

VITA

- **Education**
  - Bachelor of Science with honors (July 2018) in Computer Science and Engineering, Istanbul Şehir University, Istanbul, Turkey.
  - Master of Science (expected September 2020) in Computer Science, Özyeğin University, Istanbul, Turkey.
- **Academic Employment**
  - Undergraduate Teaching Assistant, Department of Computer Science and Engineering, Istanbul Şehir University, September 2016 - June 2018.
  - Research Assistant, Department of Computer Science and Engineering, Istanbul Şehir University, Intelligent Serious Games for Social and Cognitive Competence Erasmus+ Strategic Partnership Project, June 2016 − August 2018.
  - Graduate Teaching Assistant, Department of Computer Science, Özyeğin University, September 2018 − Present.
- **Academic Awards**
  - Valedictorian of Computer Science Graduates class 2018, Istanbul Şehir University.
- **Poster Presentations**
  - M Kutlu, K Darwish, C Bayrak, A Rashed, T Elsayed. “Embedding-based Qualitative Analysis of Polarization in Turkey” in *International Conference*

on Social Informatics (SocInfo), November 2019.

- **Publications**

  - A Rashed, M Kutlu, K Darwish, T Elsayed, C Bayrak "Embeddings-Based Clustering for Target Specific Stances: The Case of a Polarized Turkey" accepted for *Proceedings of the 15th International Conference on Weblogs and Social Media, ICWSM 2021*, arXiv preprint arXiv:2005.09649.

  - SS Hassan, Y Samih, H Mubarak, A Abdelali, A Rashed, SA Chowdhury "ALT Submission for OSACT Shared Task on Offensive Language Detection" in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. $61-65$, European Language Resource Association, May 2020.

  - A Rashed, Aİ Seçer, A Aboudakika, A Bulut" "Orienting social event streams as data stories" in *Mining Intelligence and Knowledge Exploration* (A. Groza and R. Prasath, eds.), (Cham), pp. $138-147$, Springer International Publishing, 2018.

  - AR Mohammed, A Rashed, S Shirmohammadi "A synthetic instrument for diagnosis and performance measurement of individuals with visual sequential memory deficit" in *2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. $21-26$, May 2017.

  - AR Mohammed, A Rashed, S Shirmohammadi, "Memori: A serious game for diagnosing and treating visual sequential memory deficit" in *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)*, pp. $1-7$, April 2017.

  - N Nasiri, S Shirmohammadi, A Rashed, "A serious game for children with speech disorders and hearing problems" in *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)*, pp.

1 – 7, April 2017.