ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

Çağatay Neftali TÜLÜ

A SEMANTIC VECTOR SPACE MODEL USING EUCLIDEAN DISTANCE BASED RELATEDNESS

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

A SEMANTIC VECTOR SPACE MODEL USING EUCLIDEAN DISTANCE BASED RELATEDNESS

Çağatay Neftali TÜLÜ

PhD THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Doctor of Philosophy by the board of jury on 13/06/2019

.....	.....	.....
Assoc. Prof. Dr. Umut ORHAN SUPERVISOR	Prof. Dr. Selma Ayşe ÖZEL MEMBER	Asst. Prof. Dr. Alper Kamil DEMİR MEMBER
.....	.....	
Prof. Dr. Olcay Taner YILDIZ MEMBER	Asst. Prof. Dr. Buse Melis ÖZYILDIRIM MEMBER	

This PhD Thesis is written at the Department of Computer Engineering, Institute of Natural and Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

ABSTRACT

PhD THESIS

A SEMANTIC VECTOR SPACE MODEL USING EUCLIDEAN DISTANCE BASED RELATEDNESS

Çağatay Neftali TÜLÜ

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

Supervisor : Assoc. Prof. Dr. Umut ORHAN
Year: 2019, Pages: 107
Jury : Assoc. Prof. Dr. Umut ORHAN
: Prof. Dr. Selma Ayşe ÖZEL
: Asst. Prof. Dr. Alper Kamil DEMİR
: Prof. Dr. Olcay Taner YILDIZ
: Asst. Prof. Dr. Buse Melis ÖZYILDIRIM

In this thesis, it is aimed to develop an efficient method to measure the semantic relatedness of the words. Although computer-based studies have achieved good results on this subject for nearly three decades, they have not succeeded to produce relatedness measurement close to human intuition. In this study, a WordNet-based approach is preferred, because WordNet has a graph adapted model. In addition, it is inspired by the so-called word embedding model, which is based on the dense representation of word prototypes in the low-dimensional vector space. Through proposed model, randomly positioned word prototypes are located into appropriate positions in multidimensional vector space with help of iterative learning algorithm that optimizes WordNet relation weights and word prototype positions that use Euclidean distance based relatedness. Both the positions of the words in the vector space and the weight of the semantic relations that connect the words on WordNet are determined effectively through the proposed model. The results obtained in the benchmark tests show that the new proposed model produces more successful results than the previous word-level semantic similarity studies. This approach might present a different perspective not only on semantic similarity studies but also on solving many other natural language problems.

Key Words: WordNet, Word Level Relatedness, Semantic Relatedness, Semantic Space, Distributional Semantic

ÖZ

DOKTORA TEZİ

ÖKLİD UZAKLIĞINI KULLANARAK ANLAMSAL YAKINLIK HESAPLAYAN BİR VEKTÖR UZAYI MODELİ

Çağatay Neftali TÜLÜ

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Doç. Dr. Umut ORHAN
Yıl: 2019, Sayfa: 107
Jüri : Doç. Dr. Umut ORHAN
: Prof. Dr. Selma Ayşe ÖZEL
: Dr. Öğr. Üyesi Alper Kamil DEMİR
: Prof. Dr. Olcay Taner YILDIZ
: Dr. Öğr. Üyesi Buse Melis ÖZYILDIRIM

Bu tezde, kelimelerin anlamsal ilişkilerini ölçmek için etkili bir yöntem geliştirilmesi amaçlanmaktadır. Her ne kadar bilgisayar tabanlı çalışmalar bu konuda yaklaşık otuz yıl boyunca iyi sonuçlar elde etse de insan sezgisine yakın bir ilişki ölçümü üretmeyi başaramamışlardır. WordNet, çizge uyarlı modele sahip olduğu için bu çalışmada WordNet tabanlı bir yaklaşım tercih edilmiştir. Ayrıca, kelime prototiplerinin düşük boyutlu vektör uzayındaki yoğun temsiline dayanan kelime gömme denen modelden esinlenilmiştir. Önerilen model ile, başlangıçta büyük boyutlu uzaya rasgele konumlandırılmış kelime prototipleri, WordNet ilişki ağırlıkları ile prototip konumlarını aynı anda optimize eden yinelemeli bir öğrenme algoritması ile vektörel uzayda uygun pozisyonlara yerleştirilmesi sağlanmıştır. Prototiplerin çok boyutlu uzayda uygun konumu bulması için Öklid uzaklığına bağlı bir anlamsal yakınlık fonksiyonu kullanılmıştır. Yapılan kıyaslama testlerinde ise elde edilen sonuçlar bu çalışmanın daha önceki kelime seviyesindeki anlamsal benzerlik çalışmalarına göre daha başarılı sonuçlar ürettiğini göstermiştir. Bu yaklaşımın sadece anlamsal benzerlik değil, daha başka birçok doğal dil problemine çözüm getirmede farklı bir bakış açışısı sunacağı öngörülmektedir.

Anahtar Kelimeler: WordNet, Kelime Benzerliği, Anlamsal Yakınlık, Anlamsal Uzay, Dağınık Semantik

EXTENDED SUMMARY

In natural language processing applications, it is important to measure semantic similarity effectively. The numerical representation of this measurement brings advantageous to many areas of Artificial Intelligence. It is a fact that computers and smart devices make serious contributions to the solution of various real-life problems in this century. Effective measurement of semantic similarity allows technological devices to be smarter and more advanced. Advantages and contributions of the effective text similarity can be spread into multiple areas such that; a smart robot with the knowledge and skills to chat and interact with people, the intelligent systems that can be controlled by voice, chat-bots which have the ability to solve the problems of the customers automatically in the call-centers, automatic answer grading system that grades the millions of the student exams in a very short time span and more.

The words that we see as the smallest significant building blocks in a text actually have a meaning behind them and a meaningful text is formed by the combination of these meanings. Although the similarity of the two texts was initially looked at only as a proportion of the same words in the two compared texts, the fact that the content and the different words used for the same meanings give encourage to the semantic similarity researchers to go into deep in similarity analysis. Thus, the different meanings of the words are brought together with different groups of words as a concept, and these concepts are related with each other in various types of semantic relationship where WordNet's ontological dictionary structure points to. With this structure, the interactions between the concepts are expressed more easily and effectively. Parallel to WordNet, corpora which are formed by processing and labelling of millions of words contribute to the solution of natural language problems as well. While many methods have been developed in both WordNet and corpus based, the hybrid methods that use both of them have also been successful also.

In the word level similarity, although there are serious developments in English by using WordNet and large corpora, on the other hand, there are just a few studies conducted on this subject in Turkish. Some of the reason for this handicap is;

the morphologically rich but complex structure of Turkish language in which you can make long sentences even with a word by adding suffixes and this complexity creates obstacles into the researchers to get results in the shorter period of time. Complete and efficient parsing of the morphological structure and turning it into a semantic level is considered as a starting point for semantic in natural language applications. The fact that a fully effective solution of the morphosyntactic structure is not handled in Turkish causes the studies conducted on the meaning side to be insufficient. For these reasons, the semantic similarity in Turkish has not gone into as deep as in English. Although WordNet studies have been taken a serious step in time for with BalkaNet's Turkish WordNet, the necessary progress has not been achieved over time until now. Turkish WordNet has not been developed fully and efficiently for so long time. As of 2018, Turkish WordNet named KeNet has been developed and released to be used by the researchers. So far, there are not much research results obtained about efficiency and success of this Turkish WordNet. The success of this WordNet will enable researchers to make more semantic analysis and similarity studies and this will bring advanced studies and successful results respectively in the Turkish Language.

The compatibility of WordNet with the graph structure allows the concepts to be treated as a bundle of words, represented by a node on the graph network. In addition, each of the concepts on WordNet has at least one relation to other concepts which makes WordNet a complete graph. This advantage enables to apply graph based mathematical models and algorithms into WordNet. Although there are semantic networks other than WordNet, some of them have explained in this study also, the fact that WordNet has been developed by language experts and computer experts is one of the other important reasons that make it different than other semantic networks. WordNet is very costly since it was created as a result of years of work, so there are some alternative semantic networks are created with anonymous collaborations on the internet. One of the important products of this initiative is Wiktionary, developed by Wikipedia. Graph algorithms on this type of networks have become available due to the ability of them with rich ontological structure. By successful application of PageRank method which is developed by Google to rank the web pages according to

their importance allowed to retrieve vector representation of each concept in the WordNet. And these vectors are used in the semantic similarity studies effectively as in UKB, ADW and many other studies.

Although the idea of representation of the words with vectors has been thought from the beginning, the vector operations on the word vectors obtained from the large-scale corpora have not attracted attention since they increase the cost and reduce the efficiency. However, methods such as Word2vec and GloVe, which show that words on billions of words can be represented by a vector with a maximum of 300-dimensions, have given a new perspective to the semantic representations of words. The contribution of vectors, also known as word embedding, to the semantic similarity and relatedness of words, is realized by applying the vector similarity methods.

Throughout the thesis, the vector representation of the words on WordNet, the positions on the graph network, the shortest path between the nodes on the graph network representing the words, the several variations such as all paths of a certain length were examined and their contributions to semantic similarity were investigated. Particularly, the vector representations obtained by the PageRank algorithm of the word similarities on the WordNet network have been emphasized and experimental studies on the UKB and ADW methods have been explained. WordNet's effect on the similarity of nodes and edge relations on a graph is also experienced visually. The effect of relation types on the similarity, whether the shortest path is sufficient for similarity, whether an effective similarity algorithm can be developed by weighting the relationship types is explained with experimental studies.

At the end, proposed method named “A Semantic Vector space Model Using Euclidean Distance Based Relatedness” which uses WordNet to find optimum positions the word prototypes in the multidimensional vector space by training them using mature datasets prepared by experts such as MEN3000, WordSim353and RG65 have been developed. Benchmark tests performed with the method has shown better performance than the current widely knows methods. This method is used to represent the words as prototypes in a multi-dimensional space, and the estimation of commonly used benchmark datasets by using WordNet relations has been performed with high correlation and low error. Weights have been identified in WordNet relations by the

proposed method, using these weights highly correlated word pairs have been determined, and this relationship level has not been exactly specified neither WordNet nor benchmark datasets. With the proposed algorithm, the words has been taken out of the synset pattern in WordNet and regained their identity and thus their relation to the third level words could be expressed as different from other synonyms. In addition, in the evaluations made by semantic relatedness measurement, it has reached a higher rate of success than other approaches in the literature which convert the words into a vector. Another important finding in the proposed method is the fact that WordNet verification has been made by using a combination of WordNet relations prepared by scientists and numerical values given by people who participated in conceptual similarity surveys and also determination and classification of the weights in WordNet relationships have been examined. At the end of the study, it was determined that the synonym relation was the highest and the antonym-hyponym-hypernym relations had the second most valuable weight values. All other types of relationships were placed in the third group with almost the same weight. In this way, it can be concluded that WordNet relations are consistent and can be divided into three important semantic groups.

Finally, new relationship discoveries have been made with the proposed method and different values of the dimension (N) parameter have been tried, which specifies the size of the word vector space. As the representation of words in high-dimensional space with one-hot-vector is known to cause both memory problems and computational complexity due to high sparsity, it is generally aimed to choose the smallest space that provides similar performance. However, in the experiments performed, many N values have been determined which provide almost the same performance. As a second control point, therefore, visual control of the discoveries has been provided. Word vectors that started with random distribution in training (when high dimensional space is selected) have been observed to start away from each other and thus in a low semantic relationship is gained. By the way, the word prototypes, whose semantic relationship level are defined in training datasets have positioned correctly, thus successful relationship discoveries have been conducted. Although more experimentation is needed, the consistent results provide the hope that the proposed method can give a new perspective to many NLP problems.

GENİŞLETİLMİŞ ÖZET

Doğal dil işleme uygulamalarında anlamsal benzerliğin etkin bir şekilde ölçülmesi ve bu ölçümün sayısal değerlerle ifade edilebilmesi, bilgisayarların akıllı cihazların insan hayatına bu denli dokunduğu bir dönemde karşılaşılan çeşitli problemlerin çözümüne ciddi katkılar sunacağı bir gerçektir. İnsanlarla sohbet edebilecek bilgi ve beceriye sahip bir robottan, ses ile kumanda edilebilen akıllı sistemlere, çağrı merkezlerinde müşterilerin problemlerini insansız olarak çözebilecek yeteneğe sahip sohbet-robotlarına, milyonlarca öğrencinin girip cevap verdiği açık uçlu soruları çok kısa sürede tamamen kendi algoritması ile değerlendirip insan eli değmeden etkin bir şekilde sonuçlandırabilen devasa ölçme değerlendirme sistemlerine kadar, doğal dilde anlamsal benzerlik ve anlamsal benzerliğin ölçülebilmesinin insan hayatına vereceği doğrudan veya dolaylı katkılar azımsanmayacak kadar fazladır.

Bir metnin içerisindeki en küçük anlamlı yapıtaşı olarak gördüğümüz kelimeler aslında arkalarında bir anlam barındırmakta ve bu anlamların bir araya gelmesiyle de anlamlı bir metin oluşmaktadır. İki metnin benzerliğine en başta sadece aynı kelimelerin oranı olarak bakılmış olsa da içeriğin ve kullanılan farklı kelimelerin aynı anlamlar ifade etmesi konunun şekilden daha çok, öze yönelik olarak çalışılması gereken bir konu olduğunu göstermiştir. Dolayısıyla kelimelerin farklı anlamlarının farklı kelime gruplarıyla bir araya getirilip bir kavram olarak ele alındığı, bu kavramların da birbirleriyle çeşitli tipteki anlamsal ilişki içerisinde olduğu WordNet isimli ontolojik sözlük yapısı oluşturulmuş ve bu yapı üzerinde kavramların birbirleriyle olan etkileşimleri daha kolay ifade edilebilmiştir. Buna paralel olarak doğal dil problemlerinin çözümünde başta milyonlarca, günümüzde ise milyarlarca kelimenin işlenip etiketlenmesi ile oluşturulan derlemler de doğal dil problemlerinin çözümüne ciddi katkılar sağlamaktadır. Hem WordNet hem de derlem tabanlı literatürde birçok yöntem geliştirilirken, ayrıca ikisini de kullanan hibrit yöntemler de ciddi başarılar sağlamıştır. Son yıllarda, NNLM (neural network language model) olarak bilinen, derlemler üzerindeki n-gram yapısını kullanarak eğitilen ve bunun sonucunda kelime temsillerinin optimum şekilde elde edildiği yeni yöntemler de başarı göstermiştir. Kelime benzerliğinde, WordNet ve büyük derlemler kullanılarak İngilizce için ciddi ilerlemeler kaydedilse de Türkçede bu konuda yapılan çalışmalar oldukça

azdır. Bunun sebeplerinden bazıları, Türkçe'nin İngilizceden farklı olarak sondan eklemeli bir yapıda olması, cümle yapısının çok farklı olması, bir kelimeye yapılan eklerle bir cümlenin oluşturulabilmesi gibi farklı etkenler karşımıza çıkmaktadır. Morfolojik yapının tamamen çözülüp anlamsal yapıya geçilmesi doğal dil uygulamalarında anlam için bir başlangıç noktası olarak kabul edildiğinden, Türkçede morfosentaktik yapının tam olarak etkin bir çözümünün tam olarak sağlanmamış olması da anlam tarafında yapılan çalışmaların yetersiz kalmasına sebep olmaktadır. Bu sebeplerden, Türkçede anlamsal benzerlik halihazırda derlemlerden elde edilen istatistiki bilgiler ve Türkçe sözlüğün kendi yapısını kullanarak kelimeler arası ilişki çıkarımından öte gidememiştir. BalkaNet isimli WordNet çalışması ile Türkçe WordNet için zamanında ciddi bir adım atılmış olsa da gerekli ilerlemeler zaman içerisinde sağlanamamış, istenen zenginlikte ve etkinlikte bir Türkçe WordNet günümüze değin tam olarak oluşturulamamıştır. Günümüzde, 2018 yılı itibariyle KeNet isimli Türkçe WordNet kullanıma sunulmuştur. Bu WordNet'in kullanımı, yapılan çalışmalar ve elde edilen sonuçlara yönelik henüz elimizde sağlıklı bir veri bulunmamaktadır. Bu WordNet'in başarısı da Türkçede anlamsal benzerlik çalışmalarına vereceği olumlu katkı ile kendini gösterecektir.

İngilizce WordNet'in (Princeton WordNet) çizge yapısına uygunluğu, kavramların birer kelime demeti olarak ele alınması, onların çizge ağı üzerinde bir düğüm ile temsil edilmesine olanak sağlamaktadır. Ayrıca, WordNet üzerinde bulunan kavramların her birinin diğer kavramlarla ilişki içerisinde olması WordNet'i bir tam çizge ağı yapmakta ve bu yapı üzerinde de çizge tabanlı benzerlik algoritmalarının uygulanabilmesine olanak sağlanmaktadır. WordNet haricinde çalışma içerisinde de değinilen farklı sözlük ağları olmasına rağmen WordNet'in dil uzmanları ve bilgisayar uzmanlarınca geliştirilmiş olması onu farklı kılan diğer önemli nedenlerden birisidir. Uzmanların senelerce çalışarak oluşturacağı kelime ağları çok maliyetli olacağından alternatif olarak anonim katkıların olduğu özgür ansiklopediler üzerinden gerçeklenen WordNet benzeri ontolojik yapılar da oluşturulmuştur. Bunlardan en sık kullanılanı Wiktionary'dir ve Wikipedia tarafından geliştirilmiştir. WordNet ve Wiktionary gibi ontolojik bir ağ yapısında oluşturulan kavramsal haritaların çizge ağı yapısında modellenebilmesi sayesinde bunların üzerinde çizge ağı tabanlı algoritmaların çalıştırılabilmesine olanak sağlamaktadır. WordNet üzerindeki her bir kavramı temsil

eden kelime gruplarının (synset) birer web sayfası ve bu kelime gruplarının diğerleriyle olan ilişkilerinin de web sayfası üzerindeki bağlantılarla ifade edilebilmesi sayesinde, Google tarafından geliştirilen ve arama motorunun web sayfalarının önem derecelerine göre sıralamasını yapan PageRank metodunun da WordNet'e uyarlanmasına olanak sağlamıştır ve bu sayede elde edilen PageRank vektörlerinin benzerliklerinin hesaplanması ile WordNet kavramlarının birbirlerine olan benzerlikleri de hesaplanmıştır. Kelimelerin vektörlerle temsili fikri en baştan beri düşünülmüş olsa da büyük boyutlu derlemlerden elde edilen kelime vektörlerinin üzerinde vektörel operasyonların yapılması işlem maliyetini yükselttiği ve işlem zamanını uzatması önemli birer dezavantaj olarak görülmüştür. Fakat milyarlarca kelimeden oluşan derlemler üzerindeki kelimelerin en fazla 300 boyutlu bir vektörle temsil edilebileceğini gösteren Word2vec ve GloVe gibi yeni nesil metotlar kelimelerin anlamsal benzerliğine yeni bir bakış açısı kazandırmıştır. Kelime temsilleri (Word Embedding) olarak da bilinen bu yöntemlerden elde edilen kelime vektörlerinin anlamsal yakınlığına katkısı klasik vektörel benzerlik metotlarının uygulanması belirgin olarak ortaya çıkmıştır.

Tez çalışması boyunca, WordNet üzerindeki kelimelerin vektörel temsilleri, çizge ağı üzerindeki konumları, kelimeleri temsil eden çizge ağı üzerindeki düğümlerin arasındaki en kısa yol, belli uzunluktaki bütün yollar gibi farklı varyasyonlar denenerek anlamsal benzerliğe katkıları araştırılmış, literatüre katkı sunacağı düşünülerek geliştirilen yöntemlerin de ayrıca üzerinde durulmuştur. Özellikle hem WordNet ağı üzerindeki kelime benzerliklerinin PageRank algoritması ile elde edilen vektörel temsilleri üzerinde durulmuş, bunu gerçekleyen metotlarının uygulaması da yapılmıştır. WordNet'in bir çizge ağı üzerindeki düğüm ve kenar ilişkilerinin benzerlik üzerindeki etkisi de görsel olarak deneyimlendirilmiştir. İlişki tiplerinin benzerliğe etkisi, en kısa yolun benzerlik için yeterli olup olmadığı, ilişki tiplerinin ağırlıklandırarak etkin bir benzerlik algoritması geliştirilip geliştirilemeyeceği yine çalışmalar ile gösterilmiştir.

En sonunda, hem WordNet'in çizge yapısını kullanan WordNet üzerindeki ilişki tiplerini ağırlıklandıran ve sınıflayan, hem de WordNet üzerindeki kelimelerin vektör uzayında rassal olarak konumlandırıp, MEN3000, WordSimS353 ve RG65 gibi uzmanlarca hazırlanmış olgun veri kümelerindeki kelimeleri kullanarak eğitip optimum konumunu belirleyen ve bu belirlenen konumlar üzerinden yapılan kıyaslama

testlerinde başarılı sonuçlar veren "Öklid Uzaklığına Göre Anlamsal Yakınlık Hesaplayan Bir Vektör Uzayı Modeli" metodu önerilmiştir. Bu metot ile kelimelerin uzaydaki prototipler olarak temsil edilmesi sağlanmış, aynı anda WordNet ilişkileri de kullanılarak yaygın kullanılan kıyaslama veri kümelerinin kestirimi yüksek korelasyon ve düşük hata ile gerçekleştirilmiş, WordNet ilişkilerine ağırlıklar belirlenmiş, belirlenen bu ağırlıklar kullanılarak aralarındaki ilişki tam olarak ne WordNet'te ne de kıyaslama veri kümelerinde belirtilmemiş yüksek oranda ilişkili kelime çiftleri keşfedilmiştir. Önerilen algortima ile kelimeler için WordNet'teki kavramsal temsil yapısının (synset) kalıbı dışına çıkılmış ve bu sayede diğer kelimelerle olan ilişkileri, diğer eş anlamlılarından farklı ifade olarak edilebilmiştir. Ayrıca anlamsal benzerlik ölçümü üzerinden yapılan değerlendirmelerde önerilen metodun, literatürdeki diğer anlamsal benzerlik yaklaşımlardan daha yüksek oranda başarıma ulaştığı gözlenmiştir. Önerilen çalışmadaki diğer bir önemli bulgu ise bilim insanları tarafından hazırlanan WordNet ilişkileri ile kavramsal benzerlik anketlerine katılan insanların verdiği sayısal değerleri bir arada kullanarak aslında bir nevi WordNet doğrulaması yapılmış ve ayrıca WordNet ilişkilerine ağırlıklar belirlenmeye çalışılmıştır. Çalışma sonucunda oldukça tutarlı bir şekilde "eş anlam" ilişkisinin en yüksek, "zıt anlam"-"üst kavram"-"alt kavram" ilişkilerinin de ikinci en değerli ağırlık değerlerine sahip olduğu tespit edilmiştir. Diğer ilişki tiplerinin tamamı neredeyse aynı ağırlıklarda olacak şekilde üçüncü gruba yerleştirilmiştir. Bu şekliyle WordNet ilişkilerinin tutarlı olduğu ve ayrıca üç önemli anlamsal gruba ayrıldığı söylenebilmektedir.Son olarak önerilen yöntem ile yeni ilişki keşifleri yapılmaya çalışılmış ve bu algoritmada kelime vektör uzayının boyutunu belirten N parametresinin farklı değerleri denenmiştir. Kelimelerin ikili temsil vektörleri (one hot representation) ile yüksek boyutlu uzayda temsilinin yüksek seyreklik (sparsity) yüzünden hem bellek problemi hem de hesaplama karmaşası yarattığı bilindiği için genellikle benzer başarımı sağlayan en küçük uzay seçimi hedeflenmiştir. Fakat önerilen metot ile gerçekleştirilen deneylerde neredeyse aynı başarımı sağlayan birçok N değeri belirlenmiştir. Bu yüzden ikinci bir kontrol noktası olarak keşiflerin gözle kontrolü sağlanmıştır. Eğitime rasgele dağılımla başlayan kelime vektörlerinin (yüksek uzay derinliğinin seçildiğinde) yapay olarak yaratılan büyük seyreklik sayesinde birbirlerinden uzak ve bu sayede düşük semantik ilişkide başlaması sağlandığında, eğitim kümelerinde anlamsal ilişkisi tanımlanmış olan kelime temsillerinin eğitimle birbirine yaklaşması sağlanmış ve daha başarılı ilişki keşifleri ortaya konmuştur. Elde edilen tutarlı sonuçlar, önerilen metodun birçok NLP problemine yeni bir bakış açısı kazandırabileceğine dair yeni bir umut oluşturmaktadır.

ACKNOWLEDGEMENTS

I would like to present my endless gratitude to my supervisor Assoc. Prof. Dr. Umut ORHAN for his unlimited support, guidance, patience and motivation for the successful completion of this research work.

I wish to express my special thanks to Prof. Dr. Selma Ayşe ÖZEL, Prof. Dr. Olcay Taner YILDIZ, Asst. Prof. Dr. Alper Kamil DEMİR, and Asst. Prof. Dr. Buse Melis ÖZYILDIRIM for their valuable help, support and suggestions.

I am also thankful to Enis ARSLAN and Erhan TURAN for their support and motivation.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

ADW	:	Align, Disambiguate and Walk
CBOW	:	Continuous Bag of Words
HITS	:	Hyperlink-Induced Topic Search
IC	:	Information Content
IDF	:	Inverse Document Frequency
LCS	:	Lowest Common Subsumer
LS	:	Link Strength
LSO	:	Lowest Super-Ordinate
NLP	:	Natural Language Processing
NNLM	:	Neural-Network Language Models
PMI	:	Pointwise Mutual Information
POS	:	Part of Speech
PPMI	:	Positive Pointwise Mutual Information
SVD	:	Singular-Value Decomposition
TF	:	Term Frequency
VSM	:	Support Vector Machine
WO	:	Weighted Overlap
WSD	:	Word Sense Disambiguation

1. INTRODUCTION

The similarity as a concept has been one of the factors that takes attentions since the history of mankind and pushed people to do research on this subject. Generating decision as a result of the similarity measurement and taking action according to that result and achieving to solve real-life problems of the people at the end shows the importance of this concept. It is possible to apply similarity measurement into several aspects such as similarity of the real objects, similarity of the emotions, feelings and more. In the case of similarity between objects, the features of the objects are compared but in case of similarity of the concepts, actions reason behind the concepts and responses as a result of the actions are compared. In the beginning, human intuition was judging the similarity. But now computers and smart devices that increasingly affect human's daily life are measuring the similarity and generating reactions. With the aim of modelling machines similar to human behaviour which is known as artificial intelligence, it is possible to make decisions by machines and perform actions in line with these decisions. For example, there is a similarity-based decision-making process in the image recognition system, which uses an image by itself or by identifying an object on the camera using image processing techniques. In order to make this decision, various mathematical operations and algorithms are used, which results that mathematical modelling of the concepts to be used for the similarity measurement is extremely important.

Natural languages are the elements expressed in sound and text which is developed spontaneously and have certain rules developed within a certain period of time in order to provide the mutual communication of the people living in the same geography. Natural language processing is the meaning of the text processing according to rules of the language by the machines. Morphological, syntactic and semantic rules are taught to the machines using tools of the languages like lexicons, WordNet, corpora and more. Natural language processing is described as the

intersection of linguistic and artificial intelligence (Prakash et al., 2011). The ability of the machine to process the natural language and generating the right decisions as a result of this process shows how daily life processes performed by the machines instead of human effectively. By going into the semantic level of the language, machines can be modelled as human, for example, a robot which can understand voice command given by the human. There must be a speech to text processing in this scenario. If this scenario is accomplished, it means language is transformed into text, the next step is the natural language understanding process. So using natural language understanding process machine gives meaning to the command given by the human and using its embedded algorithms it gives a decision and generates action according to a decision.

By going into the semantic dimension, there has been an increasing amount of studies in natural language processing applications. These applications are text classification, text summarization, word sense disambiguation, sentiment analysis, named entity recognition, document classifications and more (Pilehvar and Navigli, 2015; Lacobacci et al., 2016; Yousefi-Azar and Hamey, 2017; Dragoni and Petrucci, 2017; Abdalraouf and Ausif, 2018). Most of the NLP applications use textual similarity and relatedness methods and generate decisions accordingly. By means of textual similarity and relatedness, all the process of the exam evaluations are performed by the machines and this results in accuracy and speed in terms of exam results generations and distributions (Sultan et al., 2016). International exam development and evaluation centre's like TOEFL and Pearson use fully automated exam processing system, from start to the end of the exam evaluation; all steps are performed by the machines. NLP methods and applications are extremely used in all these processing steps. Benefits of all those operations performed by the machines mean more accuracy and time-saving. In order to get semantic meaning from the texts and then process the semantic parts requires many linguistic tools like lexicons, WordNet, ontologies, corpora, and mathematical models and algorithms.

The process of making raw texts ready for semantic processing is called morphological processes and these constitute the preliminary steps of natural language processing before going into the semantic level. In particular, the morphological analysis in word and sentence level makes the text understandable and semantically operable. It should be noted that the mistakes that might be done in the morphological part will directly affect the semantic side also. Morphologically analysing the text and making it semantically operable can be more difficult in agglutinative languages such as Turkish and it becomes even more complicated with its unregulated sentence structure. Although there are studies about morphological analysis of Turkish texts, extracting semantically rich words and phrases and proceeding with them into the meaning of the text is difficult. There is still no full-functional morphological analyser and POS tagger of Turkish, which constitutes a serious obstacle for Turkish texts to go into semantic analysis level. These obstacles bring serious problems in lemmatization of the words and progressing into the meaning from there. The prevalence of the studies carried out indicates that morphological analyser, POS tagging, and lemmatization parts will generate better results in the future. Not only on the morphological side but also on the semantic side there is limited number of successful studies. The lack of a nationalized, fully accessible, mature WordNet and comprehensive well-prepared corpora shows that there are milestones to be taken in NLP Studies for the Turkish Language.

English is less complex than Turkish, more work has been done in English and these studies have started early. Especially in the morphological dimension, many problems have been resolved already. Using mature and accurate tools in the morphological analysis made a faster transition into the semantic level and enabled to have serious studies to be performed on this subject. By the way, semantic text similarity and relatedness are analysed in word, sentence, short paragraph and document level in English and this is made starting from word level to document level. Many different approaches are applied in the word level and many successful

results are obtained. (Gomaa and Fahmy, 2013; Tsatsaronis et al., 2010; Wojtinnek and Pulman, 2011; Hughes and Ramage, 2007; Elavarasi et al., 2014; Altınel and Ganiz, 2018). Successful results taken in Word level bring more studies to conduct in a sentence, short text, and document levels.

Semantic similarity studies have been conducted not only in Linguistic level but also covers other scientific areas using domain-specific WordNet and corpora. Especially biology and medical sciences have successful results on semantic similarity and relatedness level, the reason for these positive results are because of the corpora and WordNet ontologies created especially for those scientific areas.

1.1. Measuring Semantic Word Relatedness

Words are the smallest units that make up a text, the text is constructed by words, so it is very usual to start with words that are the smallest building blocks for the similarity and relatedness studies of the two textual items. Especially in the semantic relatedness studies which use knowledge base and lexicon, firstly it is started from word level, then sentence level and it is gone to the document level. According to the Harris hypothesis (Harris, 1954), the words forming two texts with similar content tend to be similar. Therefore, it is a correct approach to look at the similarities and relatedness of words for two texts to be compared. After measuring the semantic relatedness of the words in the texts to be compared, we can determine the semantic relatedness of the texts formed by these words. For example, during the grading process of the short answers by the machine gives advantages and accuracy if it is possible to measure the similarity of the candidate's answer and correct answer. There many different methods developed for word-level similarity and relatedness (Speer et al., 2017; Pilehvar and Navigli, 2015; Yih and Qazvinian, 2012; Camacho-Collados et al., 2015; Agirre et al., 2009; Hirst and St-Onge, 1998; Resnik 1995) and obtained successful results on those studies. But still, there is a lack of studies who can perform relatedness judgment closely in line

with human intuition. Before going into semantic similarity/relatedness measurement, there need to be clarified similarity, relatedness, and distance in semantic level. Generally, semantic similarity and semantic distance are used in similar meaning, but semantic relatedness is used in different meaning (Zhang et al., 2013). With the semantic relations, it is aimed to mean distance in terms of different relational level such as hypernym, hyponym, part-whole, member of the domain and more (Morris and Hirst, 2004). But generally, synonym level relation is meant in the similarity of the two textual items. For example, consider “pilot” and “airplane”, there is no similarity between them in physical and object level but there are strong relations between them in a functional point of view. But if phrase “semantic similarity” is mentioned in this document, semantic relatedness should be taken into account. It should be noted that similarity (synonym, antonym and more) is just a type of relatedness.

There are three main approaches in the semantic similarity and relatedness studies. First one is the corpus based approach. In this approach, the main part is the corpus, using information from the corpus, two textual items are compared and semantic similarity of the words was calculated (Lin, 1998; Collobert and Weston, 2008; Islam and Inkpen, 2006; Hassan and Mihalcea, 2011). Corpora are the collection of annotated large texts extracted from the web or other encyclopaedic sources with automatic content discovery methods. There can be general purpose corpora or domain specific corpora also. There are many corpuses available to use publicly generated for Turkish and other languages. For example, Google book has more than 200 billion annotated words and it can be reachable and queried publicly. Words in the corpora are tagged according to context and n-grams, so conceptual and semantic level of the words are not taken into account. So semantic similarity of the words in the corpora are made in context level. Generally, the similarity of two words in the same corpora is measured according to context, documents, and n-grams on which two words are used together.

Another approach is known as knowledge based approach, in this approach ontologies are used in which words are represented as concepts in a concept map. This knowledge bases could be for the general purpose linguistic knowledge bases like WordNet (Pilehvar and Navigli, 2015; Hughes and Ramage, 2007; Agirre et al., 2009; Hirst and St-Onge, 1998) or it might be for domain specific (Bentivogli et al., 2004; Smith and Fellbaum, 2004; Beisswanger et al., 2008). English WordNet is one of the most important multipurpose generally available linguistic WordNet. There are 117K concepts (synsets) and 155K different words exist in it. 26 different relations connecting the synsets. This WordNet is constantly updated and available to researchers in a well-defined structure, can be downloaded and used directly from the internet by the researchers or queried directly. Another important semantic network is Wiktionary which is available for public access. In this type of networks, concepts are expressed in words and those concepts are connected to each other with relations. Because of the graph like the structure of the WordNet, it is possible to make relational measurements among the concepts. The difference between WordNet and Wiktionary is that WordNet has been developed by computer science and linguistic experts, but Wiktionary uses Wikipedia as a content provider that everyone contributes to developing the content. It is possible to make mathematical graph calculations on semantic networks especially if the network is considered as a graph, the words (synsets) are represented by nodes on the graph, and the semantic relations between words are treated as edges. For example, the shortest path, LCS (the lowest common subsumer) or the ontological depth measurements can be measured between the words. Graph-based approaches are very popular and have shown good results for good similarity, especially in a well-prepared word network.

There are hybrid methods which are using semantic network and corpora together (Camacho-Collados et al., 2015; Yih and Qazvinian, 2012; Radinsky et al., 2011; Speer et al., 2017). In hybrid methods, not only the relational information on the semantic network but also the statistical information of the words obtained

by the corpus are also used. Simply, semantic path information can be obtained from WordNet and TF, IDF information from the Corpus can be taken and the similarity can be calculated. Hybrid methods developed as a result of effective use of information from both the semantic network and corpora provide very successful results. Computational costs required by both the corpus side and the knowledge base are one of the negative aspects of hybrid approaches.

1.2. Importance of Graph-Based Approach in Semantic Relatedness

WordNet is one of the most preferred semantic networks to use in similarity/relatedness measurement. One of the main reasons behind is that they are defined as a graph logic and their structures are similar to graphs. Today, one of the most widely preferred semantic networks is Princeton English WordNet. In the WordNet's structure, each concept is represented by a node on the graph, while each relationship represents the edges that bind these nodes together. WordNet represents a dense graph model with this structure. Several mathematical models have been developed in the dense graph model for the calculation of similarity. There are methods that have achieved serious success in the literature from these approaches. Generally, these approaches can be expressed with various measurement approaches such as path length, concept depth or the closest common concept. WordNet is easy to adapt to the graph structure, and WordNet is easily available to get publicly to allow us to start a similarity measurement study at this point. Especially, it is easy to implement and transfer the WordNet to a graph database and the desired queries can be made easily using a mature and generally available graph database. At this point, open source and free Neo4J Community edition database is preferred, it is easy to perform calculations on the graph with special query functions provided by this database. Method for generating a graph DB from the text structure of the WordNet is described in detail in the following sections of the thesis. After converting WordNet onto the graph db, firstly, path length based methods are studied. Firstly, the shortest path (shortest path) based

methods are studied since graph DB is providing an efficient and fast shortest path extraction query. Widely used and successful method PageRank (Brin and Page, 1998) is also implemented on the graph DB and results taken from this approach is also compared.

Preliminary studies until proposed method are described in more detail in the materials and methods chapter; firstly, it was emphasized how the human approach can be modelled on semantic word similarity/relatedness. First of all, using all paths of word pairs in the widely used human judged datasets on the WordNet graph network are taken into consideration. In general, the relatedness of the shortest path between word pairs corresponds the similarity level of the word pairs; however, there can be observed in many exceptional cases. In the study (Tulu et al., 2018), the weighting of the relationship types on the WordNet graph network and the feasibility of calculating similarity by using these weights are emphasized. Another approach studied for determining the relationship weights was the method of determining weights with multiple linear regression approach using the nodes on all roads connecting the two concepts detailed in chapter 3.

1.3. Proposed Solution to Semantic Word Relatedness Measurement

Since studied graph-based methods (Tulu and Orhan, 2017; Tulu et al., 2018) which are detailed in the methods section of the chapter 3 have not provided expected results, new approaches and methods are searched, at this point, we analysed how to combine graph-based methods and NNLM (Neural Network-Based Language Model). NNLM models are generally corpus based models; they transform large texts into low dimensional and dense word vectors. Word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017) are the widely accepted NNLM studies based on machine learning techniques.

In the proposed method “A Semantic Vector Space Model Using Euclidean Distance Based Relatedness” is called SemSpace, a word on graph is considered as

a prototype and the idea of representing them in the vector space is studied and concluded that using an iterative learning algorithm which is based on learning vector quantization method (Kohonen, 1995), we can distribute prototypes in the multidimensional vector space randomly and train them using the words in the datasets to be tested until it converges with randomly started and iteratively optimized WordNet graph relations. It is planned such a method that randomly distributed word prototypes in the semantic space will be trained using Euclidean distance based similarity, so the objective function will decide whether to move the prototypes closer or move them away. By the way, WordNet weights of the adjacent nodes will be determined using the adjusted similarity distance determined with Euclidean distance. In order to determine the optimum closeness of the words in the dataset and the single-length path weights that represent human judged similarity, iterative changes of weights obtained during training have been monitored and they have adopted the principle of keeping best values and best prototype positions when they reach the optimum Spearman rank correlation value.

Unlike the methods in NNLM model, SemSpace method uses only WordNet and does not require a corpus. In the literature there are several method which creates word embedding's from the WordNet, one of them is known as wnet2vec (Saedi et al., 2018), it is aimed to generate word vectors from the WordNet relations. WordNet relations are not weighted in this study, they all have considered with same weights. And this method is compared with only Word2vec by means of its success and performance. This method outperforms word2vec in Simlex-999 benchmark dataset only. In other widely preferred datasets such as RG65, WordSim353 (WS353) and MEN3000, its performance is lower than the word2vec method.

In the experimental studies, the most commonly used similarity data from the natural language applications as training and test data were RG65 (Rubenstein and Goodenough, 1965), WS353 (Finkelstein et al., 2001) and MEN3000 (Bruni et al., 2014) data were used. In SemSpace method, benchmark tests with RG65

WS353, MEN3000 datasets show higher correlation than the current successful WordNet based methods (Pilehwar et al., 2013). Especially in reaching these values, positive contributions of keeping higher vector dimension is observed. Another benefit seen in this study is that it can be made relatedness discoveries for new word pairs. Many words pairs which do not have closeness in WordNet were observed that they are semantically related.

1.4. Thesis Organization

The other parts of the thesis are planned as follows. In the second chapter, the literature has been reviewed and important and successful studies that have entered the literature for semantic similarity and relatedness are explained. In the second chapter, methods which use machine learning for semantic similarity and relatedness are also mentioned.

In chapter three, materials and method are explained. All the materials used to perform the thesis study are explained one by one. They are WordNets, datasets, graph database tools and more. For the methods section, all the methods experienced during the thesis study are explained one by one. At the end of the chapter, proposed semantic relatedness method “A Semantic Vector Space Model Using Euclidean Distance Based Relatedness” explained in detail.

In chapter four, outputs of the proposed method is analysed in details. Retrieved measurement scores with widely known benchmark datasets is analysed and compared with the successful methods in the literature, obtained WordNet weights are reviewed and discovered semantically related word pairs are examined.

Last chapter, chapter five contains the conclusion for the thesis; it includes comments and observation about the study and future outlook for the implementation of the method proposed.

2. LITERATURE REVIEW

The semantic text relatedness studies in the literature are generally classified into three different categories. These are knowledge based approaches, corpus-based approaches, and hybrid approaches. In addition, different approaches to the classification of methods have been made. For example, Zesch and Gurevych (2010) classified semantic text similarity and relatedness studies based on path-based, Information Content (IC) based, dictionary based, and vector-based. It can also be classified as a computational and stochastic approach. Corpus based approaches are generally named as stochastic approach and knowledge based approaches are generally known as computational approaches. While knowledge-based methods measure the text similarity and relatedness using an externally available knowledge base and/or dictionary, the corpus-based methods use word statistics from the corpora. Corpus-based methods, the results of the semantic text relatedness measures are generally used in text categorization applications. There are many successful studies on the measurement of semantic text similarity and relatedness, especially in the English language. These studies have provided accelerated and improved results due to increased computing capabilities of computers and smart devices.

2.1. Knowledge Based Semantic Relatedness Approaches

Knowledge-based approaches use an externally or manually generated semantic network when measuring the relatedness. WordNet is one of the most popular semantic information networks as of today (Fellbaum, 1998). Especially in recent years, free communities on the internet like Wikipedia, which have been created by many users as a result of the widespread use of the Internet, have gained importance. The most important source for this is Wiktionary, developed by Wikipedia. Several criteria are used in similarity measurement in semantic network based approaches. These can be several criteria such as the path length on the

ontology network, the depth of the concept on the network, the depth of the common concepts.

One of the first studies in the literature regarding knowledge-based semantic text relatedness is the study of measurements using a spreading activation theory based on the theory of semantic modeling of the human memory proposed by Quillian (1967) (Collins and Loftus, 1975). According to this theory, search in human memory continues to spread until there is an intersection of two or more concept nodes. According to Spreading Activation Theory, the concept sequences in our memory are taken as cognitive nodes and these nodes are connected to each other through relations. Spreading activation networks are schematically similar to some kind of web network, but the short edges between the nodes show that they are in close relationship with each other, this makes it easier to get a relation to the original concept.

Rada et al. (1989) stated that semantic networks can be used in the measurement of semantic relatedness, only the relation of is-a (taxonomic relationship) on semantic networks can be evaluated, and other types of relationships should be excluded in the semantic similarity measurement. The natural way to measure the semantic similarity in taxonomy is to measure the distance between the nodes corresponding to the concepts that are compared, the shorter the path from one node to the other, the higher the similarity is. If there is more than one path between the two concepts, the shortest path is preferred (Rada et al., 1989). On the other hand, Lesk (1986) stated that while the semantic similarity is calculated, the dictionary meanings of the concepts can be used, and there is a direct proportion between the similarities of the two concepts with the number of common words in the dictionary meanings of the two concepts. With this approach, Kozima and Furugori (1993) have used the Longman Dictionary of Contemporary English. Words on the dictionary have been defined by converting the words into a semantic network, the dictionary definition of each word was represented as a node, and relations between the nodes are created according to

word dictionary definitions. If a word is used in the definition of another word, then in this case relation is created between them. When calculating the similarity of two words, a set of common words describing the two concepts were used. Roget's Thesaurus in the system of the hand-created dictionary is used such a similar structure. In this structure, word groups are categorized and each category is expressed as discrete sets. The relations between the concepts in such structures (Thesaurus) are revealed not as numerical values, but semantically close or not close.

Hirst and St-Onge (1998) have made the following assumption regarding the semantic similarity of the words defined on WordNet. Two words are closely related to each other if they fit one of the following conditions.

- If defined in the same synset (car and automobile)
- If the antonym relationship exists between them (long, short)
- If one word contains another (compound) (train, freight train)

If there is an appropriate path between the concepts associated with words, then the relationship between them is close to the environment or can be defined as a regular relationship. In order for the path to be considered appropriate, the path length should not exceed five and should be able to fulfill eight different patterns proposed by the method. The main point to know about this method is to find the direction change parameter. The number of change of direction and the numerical calculation of the semantic relationship contribute to the calculation of the concept of starting from the initial concept towards the target concept.

$$rel_{HS} = C - len(c_1, c_2) - k * turns(c_1, c_2) \quad (2.1)$$

In Eq.(2.1), C and k are constant values and $C = 8$, $k = 1$. $turns(c_1, c_2)$ shows the number of direction change from c_1 concept to c_2 in the WordNet.

In Sussna's (1993) approach, the concepts that remained at the bottom of the taxonomy were more semantically close to each other than the upper concepts. In this method, the edges connecting the two concepts on the WordNet network are taken as in two directions, each relationship is represented as a numerical exact weight or a weight range. For example, for hypernym, hyponymy, holonym and meronym relationship types, weights of min 1 (min_r) and max 16 (max_r) are taken. To find the weight of each type of relationship from concept *X* to concept *B*;

$$w(X \rightarrow_r Y) = max_r - \frac{max_r - min_r}{n_r(X)} \tag{2.2}$$

In Eq. (2.2), *r* represents type of the relation, each relation type has min and max value defined, *X* shows the source concept and *Y* shows the destination concept, $n_r(X)$ is the number of relation type *r* leaving the node *X*. The semantic relatedness of two adjacent nodes *A* and *B* can be calculated by the equation (2.3).

$$rel(A,B) = \frac{w(A \rightarrow_r B) + w(B \rightarrow_{r'} A)}{2 * max\{depth(A), depth(B)\}} \tag{2.3}$$

In Eq. (2.3), *r* represents the relationship from *A* is to the *B*, the opposite relationship from *B* to *A* is represented with *r'*, *depth(A)* shows taxonomic depth from root node to node *A*.

In the Wu and Palmer (1994) approach, the concepts to be compared are determined on the ontological structure (ie WordNet), and the shortest path between the two concepts is determined by using the type of is-a relationship on the structure. After identifying the lowest concept (LCS - Lowest Common Subsumer) linking the concepts, the ratio between the root conception of each concept and the root convergence of LCS is calculated by taking the proportions of the two concepts as shown in Eq. (2.4).

$$Sim_{WP}(X,Y) = \frac{2*N}{N_1+N_2} \quad (2.4)$$

The N_1 and N_2 parameters given in the equation (2.4) show the distances of the X and Y concepts on the hierarchical ontology of IS-A (Hyponomy) to the root concept. N, on the other hand, shows the distance from the root concept of the lowest common concept of the two concepts (LCS). Guessoum and Miraoui (2016) explain this structure as in Figure 2.1.

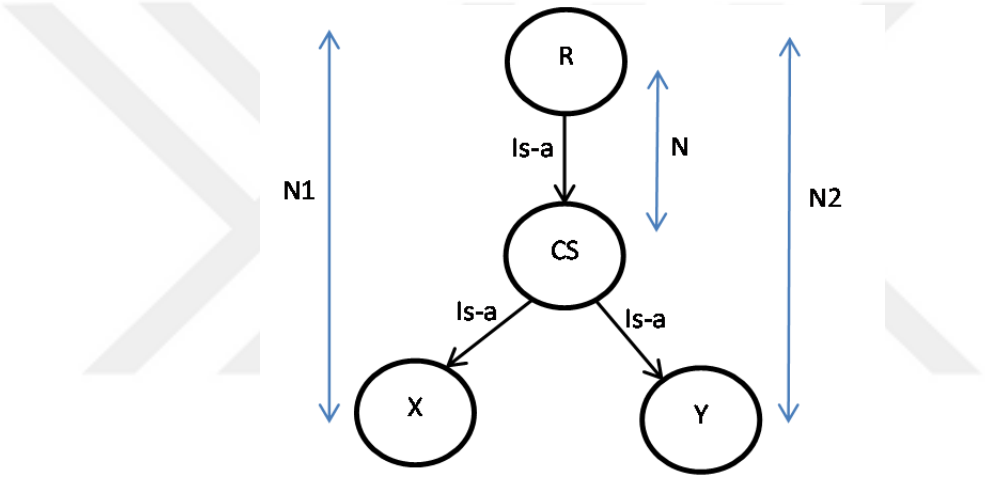


Figure 0.1. Wu and Pallmer Ontology (Guessoum ve Miraoui, 2016)

In the Wu-Palmer method, the closeness of the two concepts in the same hierarchy can be less than the proximity of the two concepts on the different hierarchy. Different approaches have been proposed in order to eliminate this discretion.

In Leacock and Chodorow (LC) (1998), a normalized pathway was used. In order to represent the two concepts defined in c_1 and c_2 on the WordNet, the LC similarity of these two concepts is formulated as in (2.5).

$$Sim_{LC}(c_1,c_2) = -log \frac{len(c_1,c_2)}{max_{c \in WordNet} depth(c)} \quad (2.5)$$

In Eq. (2.5), firstly the shortest path length between the concepts c_1 and c_2 is found and then it is divided by the depth of the deepest concept on WordNet.

2.1.1. PageRank Based Semantic Relatedness Studies

PageRank method is developed by Google inventors (Brin and Page, 1998). In the beginning, this method was intended to rank the searched web pages according to their importance. Importance of a pair was determined by the incoming links. As a logic, an important url is linked by an important url. It is in a graph structure. Web url's represents the nodes in the graph and links between url's corresponds to edges of the graph. The weighting of web url's which are the results of the user search query given by Google is done by the PageRank method. As a result of this ranking method, the most important web url was listed at top of the output. The name of the method means the weighting of the pages. The basis of the method lies in the fact that the pages that are visited much or which are referred to with too many links from other pages have a high degree of importance. Original PageRank formula;

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} \ldots\ldots + \frac{PR(T_n)}{C(T_n)}\right) \quad (2.6)$$

To show the PageRank value of *PR(A)*; the PageRank function given in Eq. (2.6) is a recursive function because the PageRank value of the current PageRank page also depends on the PageRank values of the other pages. $PR(T_i)$ shows the PageRank value of the T_i page that provides the link to page *A*. $C(T_i)$ gives the number of outbound links of the T_i page. The number *d* is a real number and is known as the damping factor. It usually takes a value between 0 and 1. The value of PageRank is equal in the first iteration, and this value is given as (*1/n*), *n*

represents the number of pages in the web space. We define the generalized iterative PageRank function as in Eq. (2.7);

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{PR_t(P_j)}{C(P_j)} \tag{2.7}$$

For any web page, the PageRank value is actually the sum of the links that give links to that page, and to division to the total pages.

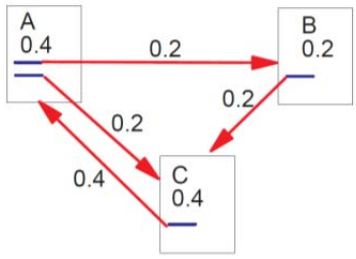


Figure 2.2. A Sample Simplified PageRank Calculation

As seen in Figure 2.2, there are two links on *page A*. One of these links is *B* and the other is *C* page. In the *B* page, there is only one link, which indicates the *B* page. On the *C* page, there is only one link, which indicates the *A page*. Based on the assumption that the total weight of the links is 1 in a simple calculation, if we start from *page A*, there are two links to the outside. In this case, if the *X* value is given for each link, then *X-weighted* links are displayed from *A to B* and from A *to C*. Since *B* is directing the only link in the *X-weighted to C*, the weight of the link from *B to C* is *X*, and after that there are two *X-weighted* links to *C*, since the sum of these links is sent from *C to A* with a single link and the weight is expressed as *2X*. In this case, *OUT (A) = 2X, OUT (B) = X, OUT (C) = 2X, since 2X + X + X = 1 there is X = 0.2*.

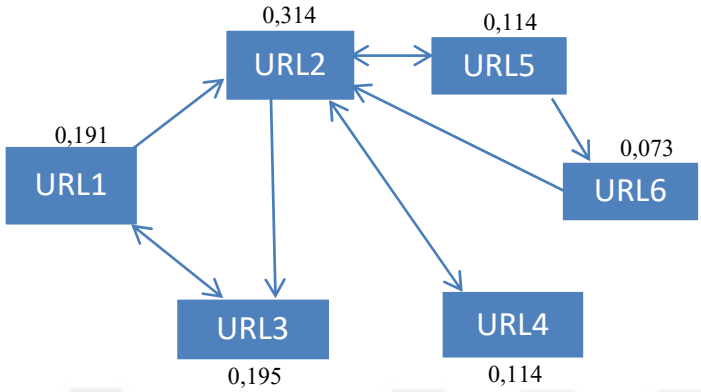


Figure 2.3 Relation between web pages in symbolic web space and PageRank values obtained accordingly

In Figure 2.3, The symbolic web space is composed of six different web pages and the pages refer to each other through links. When the PageRank method is applied to this space, the *URL2* web page that has the highest importance is listed first. A list of *URL2, URL3, URL1, URL4, URL5, URL6* is obtained when sorting according to PageRank values that are normalized over *1.0*.

According to the PageRank algorithm, the Graph with the N node is represented as a row stochastic transition matrix. This matrix is also called Markov Chain Matrix $M \in \mathbb{R}^{NXN}$. On the graph network provided with Figure 2.3, the transition matrix of the *NXN* size is obtained as shown in Eq. (2.8).

$$M = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.8}$$

The Markov Chain matrix (2.8) is actually an adjacency matrix, and the values obtained for nodes that are adjacent to each other are different from zero. Each row and each column represent the points on the graph. The sum of the index values of the vectors in each row is summed to 1. Of course, this applies to a fully connected graph. If there was a node without any external connections, all of the columns for that node would be zero. Likewise, the number of nodes that are outgoing indicates the value expressed by (*1/n*) on that index. The PageRank vector is defined as iterative on this *M* matrix as follows.

$$S^{t+1} = (1-\alpha)S^0 + \alpha S^t M \tag{2.9}$$

$$S^1 = (1-0.85) * \left[\frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\right] + 0.85 * \left[\frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\right] * \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

S^0 is the PageRank vector in an initial iteration and all values are taken by *1/N*. α is the damping factor and is usually taken as *0.85*. According to this procedure, iteration is performed until the convergence value is reached.

$$|S^{t+1} - S^t| < \epsilon \tag{2.10}$$

The ϵ value given in the equation (2.10.) is taken very low in the form of *0.001* and less. Here, the scalars of the PageRank values obtained in each iteration are taken into consideration and iteration is terminated when the convergence value is reached. If the initial vector is represented by a value other than *1/N*, then the method PageRank is called Personalized PageRank. The aim here is to give the density at the beginning of certain nodes. In general, the number of iterations in the

PageRank application is continued up to a maximum of 30, after the 30 iteration, it is assumed to be converged.

One of the most important knowledge-based approaches in word level semantic relatedness is the PageRank (Brin and Page, 1998) based approach. In this approach, WordNet was used as a knowledge base. PageRank is normally used to prioritize and sort web URLs as a response to a user search, but it has important advantages in terms of its compatibility with graph-based architecture and it has been used in semantic similarity studies between WordNet and its concepts. One of the first studies that handled WordNet as a graph network and applied the PageRank method is the Random Walks on WordNet method of Hughes and Ramage (2007). According to this method, it is expressed by Eq. (2.11), it is used the probability distribution that an agent surfing in the graph network to be in a node in iteration *t*.

$$n_i^{(t)} = \sum_{n_j \in V} n_j^{t-1} P(n_i|n_j) \tag{2.11}$$

In Eq. (2.11), $P(n_i|n_j)$ shows the probability of the agent to jumping from one node (n_i) to another (n_j), n_j^{t-1} is the probability distribution of the n_j during the iteration number *t-1*. If there is a direct connection from node *i to j*, then $P(n_i|n_j)>0$. In the Hughes and Ramage (2007) method, three different nodes were formed while the graph was created. The first of these nodes is synset nodes and corresponds to the synset on WordNet. The other is the so-called TokenPOS nodes, which are directly connected to all the synsets they are related to. For example, dog # n is a TokenPOS node, which connects to all the syncs that have the name and the name of the dog. The third type of nodes is called token, which are nodes that directly confront the word without POS information. For example, the word dog is a token and is directly linked to the dog#n and dog#v tokenpos. Synsets are linked

to each other by relationship types defined on WordNet. Overview of the proposed approach is shown in Figure 2.4

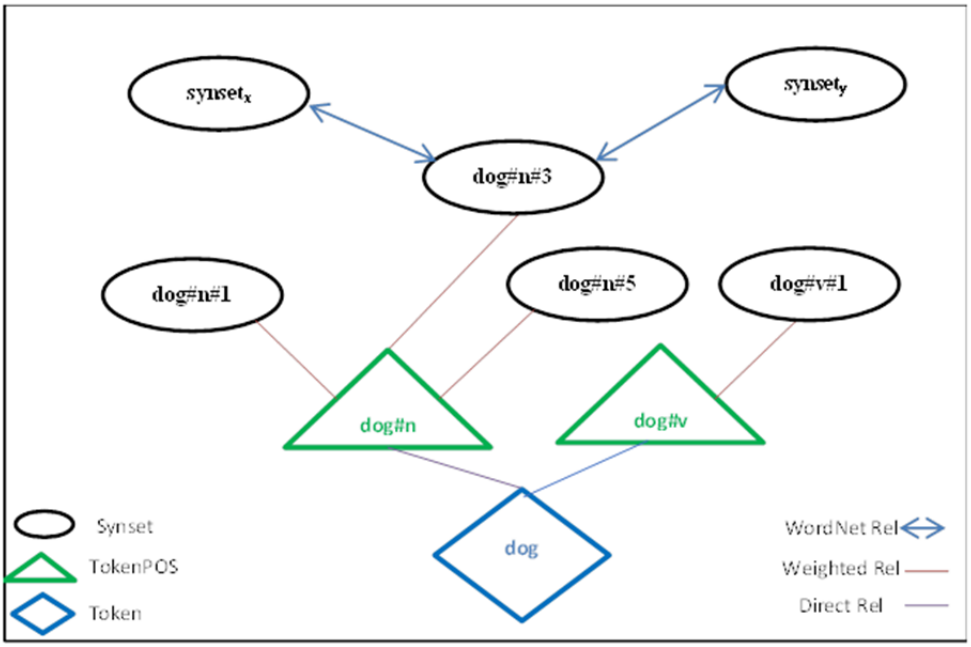


Figure 0.4 Graph overview of Hughes and Ramage (2007) approach

As shown in Figure 2.4, an association of each WordNet relationship in the absolute opposite direction is also defined. The binding of the TokenPOS to the synset was via the unidirectional weighted edges and these values were obtained from the given SemCor frequency values given on the WordNet. This situation intuitively reveals a higher number of visits to the more commonly used word. For example, while the weight from node dog#n to dog#n#1 is 43.2, the weight from node dog#n to dog#n#5 is 0.1. Weights from Token to TokenPOS were taken as the sum of the weights of TokenPOS towards Synset. Moreover, in order to be able to show the relationship of each synset with the words in the dictionary description and to use it in the graph based structure, one edge was created towards the TokenPOS, which contains the words that define it from the related synset. For

example, except for stop words in dog#n#1, one edge is defined. Nonmonotonic document frequency (NMDF) method (Haveliwala et al., 2002) was used to weight these edges. Because the NMDF method is more useful for dictionary-based similarities than other methods. Likewise, another edge type has been added from the synset to the TokenPOS. These edges are created from Synet to TokenPOS, the most common word that represents Synset.

As a result of explained process, 422K node and 5133K edge graph were obtained. On this sparse graph, the nodes are directly connected to each other at a rate of about ten thousand. For each edge type, transition matrices (as in eq. 2.8) also known as Markov chain matrix are created separately and each index value on these transition matrices is calculated with $c_{ij} = P_K(n_i|n_j)$. The c_{ij} value can be between [0 1] or weight value according to the edge type. Then the transition matrixes obtained for all edge types are collected and full transition matrix *M* is obtained. The stationary distribution vector is calculated for each node after the transition matrix. The Stationary distribution vector and the PageRank vector are similar.

$$v^{(t)} = \beta v^0 + (1-\beta)v^{t-1}M \tag{2.12}$$

Eq. (2.12) shows determined the PageRank vector *(v)* of any node in iteration *t*. v^0 is the initial pagerank vector of the node it is taken as first index 1 and other indexes 0 like one hot vector representation, β is the dumping factor, generally taken as *0.85* and *M* is the transition matrix. Similarity methods based on the similarities of vectors were applied to calculate similarity on stationary distribution vectors. These vectors are also called probability distributions vectors. The most classical vector analogy is Cosine similarity. Equation (2.13) is also given the cosine similarity calculation for the two probability distributions vector.

$$sim_{cos}(p,q) = \frac{\sum_i p_i q_i}{||p||\,||q||} \tag{2.13}$$

In the Eq. (2.13), p and q are the vectors i is the index, p_i and q_i are the value of corresponding indexes in p and q vectors. The more successful alternative method Kullback-Leibler divergence method is defined as the equation (2.14).

$$D_{KL}(p||q) = \sum_i p_i log \frac{p_i}{q_i} \tag{2.14}$$

Since the Kullback-Leibler divergence method becomes undefined in case one of the q_i values is zero, Jensen-Shannon (JS) divergence method has been developed in order to overcome this limitation.

$$D_{JS}(p||q) = \frac{1}{2} D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2} D_{KL}(q||\frac{p+q}{2}) \tag{2.15}$$

Another study that uses the PageRank method and achieves successful results is the UKB: Graph-Based Word Sense Disambiguation and Similarity (Agirre and Soroa, 2009) method. UKB is a collection of programs for performing graph-based Word Sense Disambiguation (WSD) and lexical similarity/relatedness using a pre-existing knowledge base. UKB applies random walks, e.g. Personalized PageRank, on the Knowledge Base (KB) graph to rank the vertices according to the given context. It includes tools to produce graphs from KBs like WordNet. In this study, a graph which contains the synsets and their relations on WordNet was developed and Pagerank method was applied to this graph and word vectors belonging to each synsets called semantic signature was generated. Using these vectors, semantic relations measurements of the synsets were performed. In addition, these prepared vectors were presented to researchers and different methods and algorithms on those vectors were studied. In fact, Align Disambiguate Walk (ADW) (Pilehvar et al. 2013) has used these vectors to make successful

similarity studies in terms of words, sentences and short text. Moreover, with the UKB method, not only WordNet, but also more different encyclopedic/dictionary networks can be converted into a graphical network and the semantic similarity can be made on the graph creation script is presented to the user. As a result, WordNets developed for different languages are used to obtain their own word vectors and thus they are used in similarity studies.

The starting point of the UKB method is to find an unsupervised solution to the Word Sense Disambiguation problem. The benchmark tests using WordNet 3.0 data on UKB achived 0.83 for RG65 data and 0.58 spearman rank for WS353 dataset. ADW (Pilehvar et al. 2013) is another successful study that has achieved successful results using the Pagerank method. It is the study that has achieved the most successful results on RG65 and WS353 benchmarking data in the word similarity among studies using WordNet as a knowledge base. When Wiktionary was used as a knowledgebase, Spearman correlation 0.92 was obtained on RG65 data, it is the best results in the literature (Pilehvar et al. 2013). ADW used the semantic signatures of the synset directly in their study, signatures were produced by UKB method.

They studied different vector similarity metrics on these signatures. The most successful method was Jensen Shannon (2.15) and Weighted Overlap (2.16).

$$Sim_{wo}(S_1, S_2) = \frac{\sum_{h \in H}(r_h(S_1) + r_h(S_2))^{-1}}{\sum_{i=1}^{|H|}(2i)^{-1}} \tag{2.16}$$

In the Eq. (2.16) S_1 and S_2 shows r_h Weighted Overlap (Pilehvar et al., 2013) function to represent semantic signature vectors provided by UKB method. Intersection set of non-zero index value between two semantic signatures is given by H. If no weight is between two signatures, the function results to zero. As already known, the value of zero in the intersection will result in zero. The overlap

of high-valued indexes at the intersection of the two vectors means that the similarity is higher.

2.2. Corpus Based Semantic Relatedness Approaches

Corpus is defined as all documents compiled to access information in a language and submitted to user research. These are such documents as scientific books, scientific and technical papers, theses, technical notes, documents, contracts, treaties, regulations etc. documents, or those related to them. Semantic information obtained through the corpus can be used to measure semantic similarity between texts. Corpus-based approaches also differ among themselves.

Corpus-based approaches can first be given to the Latent Semantic Analysis (LSA) (Dumais et al., 1988), which is a text classification method that shows the concept of the words in a corpus. The LSA matches both the words in the given collection to the concept in the field, causing us to determine which word is closer to which concept. The LSA is based on the assumption that "similar words are used in similar texts". The LSA model first analyzes the text, calculates the frequency of each term on the text and creates a weight matrix. In this matrix, rows represent singular terms in the text, and columns represent the text instance. The cells of the matrix represent the frequency of each term in the given text sample. A mathematical technique called singular value decomposition (SVD) is used to reduce the number of columns by retaining the number of rows in the weight matrix. The terms are then compared by evaluating the cosine of the angle between the vectors generated by the two rows. In the measurement of the similarity of the two terms defined on the matrix, the rows in the weight matrix expressing the term are treated as vectors and a cosine similarity calculation is made between them.

Another successful corpus-based method is the Hyperspace Analogue to Language (HAL) method (Lund and Burgess, 1996). As a result of this process, a vector is created for each word on the corpus, which contains numerical values that express the meaning of the word. This procedure was executed on a large natural

language processing corpus called Usenet and word vectors were created. The coordinates represented by the word vectors obtained were analyzed in order to show the relationships between the words in a large space. In the similarity and multidimensional scaling processes, it was found that there was very important semantic information on these vectors (Lund and Burges, 1996). The similarity of vector similarities and similarity of these reactions were compared. It was concluded that the resulting vectors could be a basis for semantic memory modeling hyperspace analogue to language (HAL). In this method, the word co-occurrence matrix of the word NXN is obtained by collecting the weights of the words together in a certain frame length. If you want to find a word which is similar to which the word is obtained, the rows of those words are taken as vector and the similarity between them is calculated using Minkowski family of distance metrics.

$$Distance = \sqrt[r]{(|x_i - y_i|)^r} \tag{2.17}$$

In Eq. (2.17), when $r = 2$ is taken, the distance is the Euclidean distance. When $r = 1$, there is a city-block distance.

Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2009) has been proposed as a new method that can make semantic interpretations on a corpus that is collected from free encyclopedias. By this method, the semantic representation of the concepts in multi-dimensional space is obtained by extracting from large encyclopedias like Wikipedia and it is an effective method for measuring semantic similarity. There are concepts on a document and the words that make up that document. In the ESA method, the TF-IDF values of each word are given by Eq. (2.18).

$$T[i,j] = tf(t_i, d_j) log \frac{n}{df_i} \tag{2.18}$$

T is sparse table where each of the *n* columns corresponds to a concept, and each of the rows corresponds to a word that occurs in d_i. t_i is the corresponding term while d_j is the document in (2.18). *tf* term frequency is calculated by Eq. (2.19).

$$tf(t_i, d_j) = \begin{cases} 1 + log\ count(t_i, d_j), & if\ count(t_i, d_j) > 0, \\ 0, & otherwise \end{cases} \quad (2.19)$$

The salient semantic analysis (SSA) (Hassan and Mihelcea, 2011) is used to interpret the salient concepts obtained from the encyclopedic corpora and interpret them semantically. The difference between this method and other corpus-based methods is that it calculates the distance between the concept-based profiles. Concept-based profiles constitute the salient concepts that are derived from the context on a large corpus.

The decisive concepts on the document that expresses the whole document are called salient concepts, and a salient concept is expressed by the weight vector formed by salient concepts. Salient concepts are also called semantic profiles of that concept. The semantic profile of each concept on the Wikipedia collection is the other concepts used to express that concept. The semantic closeness of two different words is obtained by combining the semantic profiles of those words. If there is m number of the token on a given C corpus, an E is a co-occurrence matrix with $E = NXW$ equation is obtained if the number of words is *N* and the size of the concept (the singular concepts defined in Wikipedia) is *W*, and the window here is expressed as *k*. Each E_{ij} element defined in matrix *E* is shown in equation (2.20).

$$E_{ij} = f^k(w_i, c_j) \quad (2.20)$$

In Equation (2.20), f^k shows how many times terms w_i and c_j are present together in a certain window range.

Each element on the $P = NXW$ of the PMI (Pointwise Mutual Information) matrix can be expressed as follows;

$$P_{ij} = log_2 \frac{f^k(w_i,c_j)Xm}{f^C(w_i)Xf^C(c_j)} \tag{2.21}$$

The values of $f^C(w_i)$ and $f^C(c_j)$ in Eq. (2.21) are the frequency values of the word w_i and the concept of c_j on the collection. By filtering in each P_i rows, β_i is obtained (Islam and Inkpen, 2006) and thus the values below a certain threshold are eliminated and the best PMI values are obtained.

$$\beta_i = (log_{10}(f^C(w_i)))^2 * \frac{log_2 N}{\delta}, \delta \geq 1 \tag{2.22}$$

In Eq. (2.22), δ is a constant value that varies according to the size of the corpus. In order to measure the similarity for the two words given on this matrix, the equation of the modified cosine similarity Eq. (2.23) is used.

$$Score_{cos}(A,B) = \frac{\sum_{y=1}^{N}(P_{iy} * P_{jy})^{\gamma}}{\sqrt{\sum_{y=1}^{N} P_{iy}^{2\gamma} * \sum_{y=1}^{N} P_{jy}^{2\gamma}}} \tag{2.23}$$

A and B are the words to be semantically compared, P_i and P_j are the corresponding columns of word A and B in the PMI matrix P and y is the index value of P_i and P_j vectors. The γ value given by (2.23) is used to control the deviation in weights. In the similarity of cosine, the value of 1 refers to the terms that are identical and the λ normalization factor is used in order to obtain more meaningful scores as the similarity range should be in the range [0,1]. It

corresponds to the maximum score to be obtained while measuring the relatedness between words *A* and *B*, according to this assumption;

$$Sim(A,B) = \begin{cases} 1 & Score_{cos}(A,B) > \lambda \\ \dfrac{Score_{cos}(A,B)}{\lambda} & Score_{cos}(A,B) \leq \lambda \end{cases} \quad (2.24)$$

When the model was modified slightly to the Second Order Co-Occurrence Pointwise Mutual Information (Islam and Inkpen, 2006) (SOCPMI), stronger and better results than cosine similarity were obtained. By using the P_i and P_j lines in the *P* matrix obtained for the words *A* and *B*, the SOC similarity in the Eq. (2.25) is obtained.

$$Score_{soc}(A,B) = \ln\left(\frac{\left(\sum_{y=1}^{N}(P_{iy})^{\gamma}\right)}{\beta_i} + \frac{\left(\sum_{y=1}^{N}(P_{jy})^{\gamma}\right)}{\beta_j} + 1\right) \quad (2.25)$$

In Eq. (2.25) given $P_{iy} > 0$, $P_{jy} > 0$ and constant value with the degree of deviation on the PMI values are taken under control, the values obtained should be normalized with the equation (2.21) that corresponds to β_i which are the highest scoring PMI terms. The γ value is used to control the deviation in weights and *N* is the total number of the words.

2.3. Hybrid Text Similarity Approaches

In the hybrid approach, not only the knowledge base but also the use of the corpus, data from two different sources of information is blended. The idea behind the hybrid approach is neither knowledge based nor corpus based approach does not satisfy semantic similarity measurement results. In particular, the widespread use of mobile devices on the Internet and the availability and use of large data from social media lead to the popularization of hybrid approaches. As a Hybrid

approach, Resnik (1995) conducted studies to measure semantic similarity between defined concepts on taxonomy and ontologies. Taxonomy, ie, the science of classification, is the science that makes the classification of the concepts according to the hypernym and hyponym relations. Ontology is defined as concepts, data and/or entities that define one, many or all fields (whole domain), to define their characteristics and to define the relations between them. In his study, Resnik (1995) showed that semantic similarity was not only the shortest path on the inter-conceptual taxonomy but also Information Content (IC). Thus, he has achieved 0.90 spearman correlation on MC30 (Miller and Charles, 1991) dataset. The information content of a concept is measured by the negative log similarity (Ross, 1976). Therefore, this leads to the higher the awareness of the concept, the less informativeness of the concept. Therefore, it is concluded that the IC values of the frequently used concepts are low. With this assumption, IC and semantic similarity are formulated as in Eq. (2.26).

$$Sim(c_1, c_2) = max_{c \in S(c_1, c_2)} = [-logp(c)] \quad (2.26)$$

$S(c_1, c_2)$ is the common concepts set of the c_1 and $c_2$in the taxonomy which constitute the level (Hypernym) concepts in taxonomy. *p(c)* is the probability value of the word *c* in a given corpus (2.28). Since common concepts are generalized as they go upward in the taxonomy, IC concepts of them are low, the value of log likelihood, which is generally the lowest common concept, is high. Resnik used WordNet, which has over 50,000 names defined as taxonomic structure. In order to obtain IC values, it used frequency values of related words in Brown Corpus of American English (Francis and Kucera, 1982). The frequency value of each name on the corpus is obtained by the sums of that name and its upper-level concepts.

$$freq(c) = \sum_{n \in words(c)} count(n) \tag{2.27}$$

The set of Words *(c)* is the set that expresses all the upper concepts of concept *c*. The probability of concept *c* is also calculated by the relative frequency process.

$$p(c) = \frac{freq(c)}{N} \tag{2.28}$$

In Eq. (2.28), *N* gives the number of all names in the corpus used.

In the Jiang and Conrath (1997) approach, the semantic distances between the concepts placed on taxonomy suggested so that they could be enriched by the statistical analysis obtained from the corpus data and better results could be obtained. In their approaches, they synthesized the information content data obtained from the corpus of the concepts and edge weights between concepts in the taxonomic structure and obtained *r = 0.82* correlation on MC30 dataset. The main methods for the determination of edge weights between concepts can be listed as follows; The node frequency within the network affects the edge weight at the correct rate. As the depth of the node in the taxonomy increases, the edge weights between the concepts increase. The relationship between concepts is also an important factor and the type of relationship is the factor in determining the edge weight.

Jiang and Conrath (1997) determined a combined approach considering these factors when determining the edge weight. They named the edge weighting as Link Strength (*LS*), and when calculating it, they weighted the edge by taking the difference of the *IC* values between the parent concept and the child concept.

$$LS(c_i, p) = -\log\big(P(c_i|p)\big) = IC(c_i) - IC(p) \tag{2.29}$$

In the Eq. (2.29) p is the parent concept, c_i shows the hyponyms of this concept. Therefore, the weight of the edge between the parent concept and any child concept has reached into the conclusion that the difference between the two *IC* values.

$$Dist(w_1, w_2) = IC(c_1) + IC(c_2) - 2 * IC(L_{super}(c_1, c_2)) \tag{2.30}$$

While w_1 and w_2 represent two words in the Eq. (2.30), the semantic similarity between these two words is calculated by the *IC* values of the concept, which is the common upper concept of the two concepts with the sum of the *IC* values of the concepts that contain two words.

Lin (1998) put forward the assumption of a universal similarity. According to this assumption, the similarity of the two ordinary objects is related to the amount of their common characteristics. The similarities of those who have more common features are more similar. In addition, the similarity of the two concepts A and B is closely related to their differences, the more the differences, and the less similar. The similarity of A and B is that point A and B are identical. In summary, he stated that the similarity of the two objects is directly proportional to the partnership of the information used for the definitions of the objects. In the definition of two objects from this assumption that the more common information we use, the more similar these two objects can be considered. The commonality of the two concepts is treated as the *IC* value of the two concepts. We can express this on the taxonomic structure with the Eq. (2.31).

$$sim_L(c_1, c_2) = \frac{2xlogp(lcs(c_1,c_2))}{logp(c_1)+logp(c_2)} \tag{2.31}$$

lcs: lowest common subsumer, ie c_1, and c_2 is defined as the concept of in the taxonomic structure. *p(c)* is the frequency values of the concepts from the corpus. Lev Finkelstein et al. (2002) made an assumption based on context-based web search engine, they developed a similarity algorithm by using both their computational vector representation of words and their position on WordNet. For the searched words, at least 10,000 documents are sampled in 27 different categories. And each word is represented by a 27-dimensional vector. While each dimension in the vector represents a domain, the frequency of the word in this domain also corresponds to its value in this dimension. The distance between the vectors in this space also reveals the semantic distance between the words and the similarity measure of correlation based on the similarity of these words is used.

$$sim_{VB}(w_1, w_2) = \sum \frac{(\vec{w}_1 - \overline{w}_1)(\vec{w}_2 - \overline{w}_2)}{\sigma_1 \sigma_2} \tag{2.32}$$

In Eq. (2.32), $\vec{w}_1$and $\vec{w}_2$ represent the corresponding word vectors $\overline{w}_i$ which are 27-dimensional vector and σ_i represent the standard deviation values of the related words. In order to further develop the statistical-based semantic similarity, it was emphasized that the linguistic knowledge of these concepts should be integrated and this information was obtained from WordNet and these two similarities were integrated into the hybrid structure.

On WordNet, Resnik (1999) uses a knowledge-based semantic similarity to obtain a hybrid similarity equation as follows.

$$sim(w_1, w_2) = \alpha . sim_{VB}(w_1, w_2) + \beta . sim_{WN}(w_1, w_2) \tag{2.33}$$

In the Eq. (2.33) $sim(w_1, w_2)$ represents the vector-based similarity given in Eq. (2.32), while sim_{WN} represents the WordNet-based Resnik similarity given in (2.26). The α and β values show the values obtained from the word training sets

with the cross-validation technique. To test the success of this method, the MC30 data set was not sufficient and the WS353 dataset was prepared for this study. While preparing this dataset, 16 different subjects scored up to 0 (unrelated) to 10 (synonyms) for 350 different words (nouns). A 55% correlation was obtained with this hybrid method.

Yih and Qazvinian (2012) have represented the words with vector space models that they formed using heterogeneous information sources. Heterogeneous sources are classified into three categories: Corpus, Thesaurus, and Web Based. The approach is based on the hypothesis of Harris (1954). In this hypothesis, "the words used to express the same content are similar to each other". Therefore, the large-size sparse word vector represents the content and the words in this content also gain meaning in this way. The words around *+n, -n* of the target word on a given corpus are taken into count in the content. This is also referred to as the bag of the words and is represented by the TF-IDF term vector. Each term of the vector is weighted with $log(freq) \times log(N/df)$. Here, the *freq* value gives the number of passages in the whole corpus of the word, while *df* gives the frequency of the documents and the total number of documents in *N*. Word vectors were created using Wikipedia dump, web and thesaurus and cosine similarity of the word vectors were calculated. Compared to benchmark results taken from WS353, RG65, MC30, and MTurk-287 datasets, these results showed successful results with high correlation.

ConceptNet Numberbatch (Speer et al. 2017) method is the most successful study in the hybrid field and successful results were obtained with 0.828 Spearman rank correlation value in WS353 dataset. In this study, a hybrid approach is not only used with the knowledge base and corpus but also they have combined with word embedding's. ConceptNet is a knowledgebase graph. In this structure, the words and/or words consisting of more than one word are connected by weighted edges. In the beginning, ConceptNet was launched as a knowledge-based project called Open Mind Common Sense (Singh et al., 2002), which was created

with the contribution of a free encyclopedia concept. In version 5.5, additions and enrichments have been made from many different language sources in the world. The Retrofitting method (Faruqui et al., 2015) introduced a hybrid structure that learned from distributional semantic space and was called ConceptNet Number-batch. The ConceptNet, known as multi-source knowledgebase in the graph structure, has more than 8 million nodes and more than 21 million edges. The connections between the nodes are represented by 36 different relationship types. Some types of relationships are symmetric, while others are one-way. Asymmetrical and sparse term matrix was obtained from ConceptNet. Each cell on this matrix refers to the weights of the two opposite terms to each other. Word embedding's are obtained from term to term matrix using the method proposed by Levy et al. (2015). In this way, the word embedding's, ConceptNet-PPMI, are obtained. PPMI (positive pointwise mutual information) by using the SVD method was reduced the dimension to 300. Therefore, word embedding for each word was obtained and an approximate similarity calculation was made for any two nodes in the graph structure. The Retrofitting process and the existing word embedding matrix are made more regular by using knowledge graph. Retrofitting is defined as a graph-based learning technique to obtain the highest quality semantic vector using a relational resource. This technique is applied to the word vector derived from the graph generated by a lexical resource. In this way, it is considered a post-processing process. In this way, retrofitting can be used for any training method and pre-trained vocabulary vector. This method has also counted with very good results in terms of speed. For example, a graph structure consisting of 300-dimensional and 100,000 words takes only 5 seconds to apply retrofitting to this structure.

Let $V = \{w_1, w_2, \ldots w_n\}$ be a set of words. We can define Ω as a graph (V,E) consisting of nodes and no directional edges, an ontology showing the relationship of the words within the V set. From here the expression $\in E \subseteq VXV$ expresses the semantic relationship between two words (w_i, w_j). $Q^{\wedge}$ represents the cluster of the

vectors shown as $\hat{q_i} \in \mathbb{R}^d$, $w_i \in V$ and d is the dimension of the vector. Here, the aim is to obtain the closest neighboring nodes Ω within the vector indexes $Q^\wedge$ to be $Q = (q_1, q_2, \ldots q_n)$. Minimizing the difference between Q and $Q^\wedge$ means that the value of the objective function is reduced to a minimum, so the objective function is shown with Eq. (2.34).

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \parallel q_i - \hat{q_i} \parallel^2 + \sum_{(i,j)\epsilon E} \beta_{ij} \parallel q_i - \hat{q_i} \parallel^2 \right] \quad (2.34)$$

In this case, the word vectors are trained independently of the semantic network, and then they are subjected to retrofitting. Visible improvements were observed in the test results performed on the trained vectors using this method.

2.4. Machine Learning Based Semantic Word Similarity Approaches

The first widely accepted study on this subject used a feed-forward neural network (Bengio et al., 2003). This network has a linear projection layer and a non-linear hidden layer. Thus, the learning process of the vector equivalents of the words is realized in this way. This study was followed by many different studies. In another similar neural network architecture (Mikolov, 2007 and Mikolov et al., 2009), word vectors are learned by using a single hidden layer. For this reason, there is no need to create a complete NNLM for learning vocabulary vectors (Mikolov at al., 2013). Inspired by this simple approach, the Word2vec (Mikolov et al., 2013) method was developed. In this way, the word vectors obtained from this approach will contribute to the solution of the many NLP problems. It has also become possible to use vocabulary vectors derived from different corpora for future studies. By the word vectors obtained, several different similarity algorithms can be performed on it. A further disadvantage is that the cost of the word vectors to be obtained by this method is much greater than that obtained by other known methods. Here the word vectors are generated and their representations are obtained by learning the neural network.

Obtaining a word vector by using statistical methods (e.g. LSA, ESA) over a large corpus is quite costly. The operational complexity of these processes is another disadvantage. A new Log-Linear Model (Gujarati and Porter) was introduced in order to prevent the resulting time cost and operational contrast for the creation of the vectors. In this new architecture, NNLM is trained in two stages. The first is to learn continuous vocabulary vectors from a simple model, and the other is to train these vocabulary vectors using n-grams. Using one hot encoding vector representation it is not possible to find the similarity between two words. The first of the Log-Linear models is the Continuous Bag-of-Words Model (CBOW). Its structure is similar to the feed-forward NNLM.

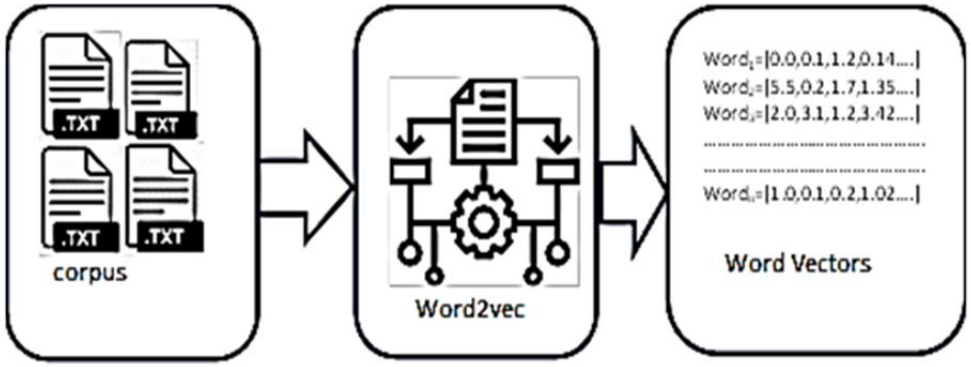


Figure 2.5. From corpus to word vectors converting process

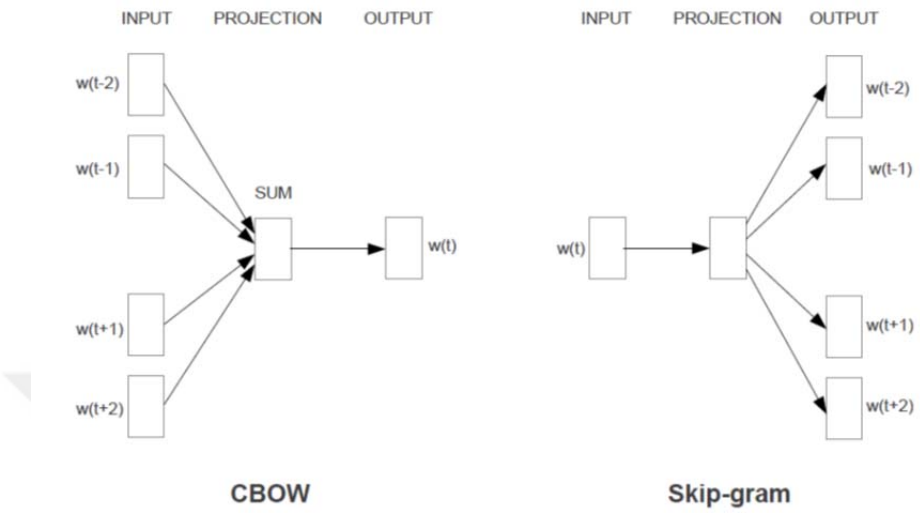


Figure 0.6. Structure of CBOW and Skip Gram Models (Mikolov et al., 2013)

According to Word2vec (Mikolov et al., 2013) logic, similar words are found in similar content. While CBOW is used to estimate the word that is not in this content by using content, Skip-gram is used to take the target word by guessing the words around the target word. The widely known contribution that the Word2Vec method brings to the literature is that it enables the acquisition of low-dimensional vocabulary vectors from large corpora with billions of words. The discrete representation of words in the vector space works well in the process of grouping similar words in natural language processing. Since it does not contain matrix multiplication of very large sizes, the phase of learning the Skip-gram model is relatively easy. The Skip-gram model is primarily trained to predict the words around a word or a word in a sentence. The aim in training is to maximize the mean logarithmic likelihood $w_1, w_2, w_3 \ldots . w_T$.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log \mathrm{p}(w_{t+j}|w_t) \tag{2.35}$$

In equation (2.35), *c* shows the size of the training context, *T* is the size of words in the training set. As the training set grows, the consistency in the results will increase and the training time will increase also. The basic Skip-gram formula $p(w_{t+j}|w_t)$ is defined by the softmax function as in (2.36).

$$p(w_o|w_i) = \frac{\exp(v'^{T}_{wo} v_{wI})}{\sum_{w=1}^{W} \exp(v'^{T}_{wo} v_{wI})} \tag{2.36}$$

v_w and v'_w show the input and output vector representations of the *w*, while *W* indicates the number of words in the vocabulary.

One of the unsupervised learning methods used to find the vector representation of words is the GloVe: Global Vectors for Word Representation (Pennington et al., 2014) method. For the process of learning, a corpus is used, together with the transition statistics of the words obtained from this corpus are collected, resulting in vectors representing the words. It is used to calculate the semantic similarity of the words related to the Euclidean distance or cosine distance of the two vectors. Sometimes, according to this calculation, the nearest neighbors can show rare but relevant words outside a person's memory.

The GloVe algorithm consists of the following steps;

1. X is a co-occurrence matrix showing how frequently each X_{ij} word is used with the word *j*. Generally, a corpus is scanned as follows; by specifying a window size, the information on whether the target word is around this distance (in front or behind) and how far away it is shown on the matrix. Thus, a restriction is obtained for each word.

$$w_i^t w_j + b_i + b_j = \log(X_{ij}) \tag{2.37}$$

In Eq. (2.37), w_i represents the vector of the parent word, while w_j represents the content vector. b_i *and* $b_j$are constant scalar sizes of these two vectors.

2. The cost function is obtained by the following in (2.38);

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij}) \left(w_i^t w_j + b_i + b_j - \log(X_{ij})\right)^2 \tag{2.38}$$

In eq.(2.38), V is the size of the vocabulary list. In (2.39) f is a weighting function and is used by the developers of GloVe equation.

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{X_{max}}\right)^{\propto} & if\ X_{ij} < X_{max} \\ 1 & otherwise \end{cases} \tag{2.39}$$

NNLM has recently developed a newly popular method, FastText (Bojanowski et al., 2017). The starting point of the FastText method is the fact that the word embedding vectors of morphologically rich languages (such as Finnish, Turkish) obtained by the existing Word2Vec or GloVe methods do not achieve the desired success in the semantic similarity text classification. NNLM-derived word representations obtained by the skip-gram or CBOW method or the words obtained from the neighboring words in the absence of an uncontrolled complete corpus of the words acquired a morphological change in the representation of the words are causing problems. In order to overcome this, FastText method takes into account the morphological structure of the words while creating distributional representational vectors of words. FastText uses the skip-gram model, and each word is represented as a set of characters n-grams. Each character is associated with a vector of n-grams and the words are considered as a sum of these vector representations. This method is especially fast because of this name. The training on large collections takes place quite quickly and allows for the representation of

non-trained word representations. As a result of running the method on the corpora of different languages, the best results were obtained in terms of similarity and analogy. If the size of the set of words W and a word in the cluster are represented by the index $w \in \{1, \ldots, W\}$, the goal is to obtain the vector representation of the word w. Here, the distributional hypothesis (Harris, 1954) was affected. Accordingly, word representations are best learned by the content around it. If a part of the words given in a large corpus is used as $w_1, \ldots w_T$ training set, the purpose of the skip-gram model is to maximize with log-likelihood with the formula given in (2.37).

$$\sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c, w_t) \tag{2.40}$$

C_t, given in (2.40) shows the index of the words surrounding w_t. The w_c content words around the word w_t are derived by the given word vectors. Let's assume that many scaling functions are given and word pairs are scaled with this function. In this case, it is possible to use the softmax function in order to estimate the content word.

$$\mathrm{p}(w_c|w_t) = \frac{e^{s(w_c,w_t)}}{\sum_{j=1}^{W} e^{s(w_t,j)}} \tag{2.41}$$

In (2.41), with w_t it is only given in the word w_c can be estimated. In this case, the determination of the content words should be converted to binary sorting and the content words should be estimated in this way. Taking this into account, the objective function is obtained as seen in Eq. (2.42).

$$\sum_{t=1}^{T} \left[\sum_{c \in C_t} \mathrm{l}\big(s(w_c, w_t)\big) + \sum_{n \in N_{t,c}} \mathrm{l}\big(-s(w_t, n)\big) \right] \tag{2.42}$$

In Eq. (2.42), in order to estimate w_t and around the w_c words, s scaling function is used to estimate the word and word vectors. In this case, if we define each word w with two-word vectors as u_w and v_w, they are also called input and output vectors. In this case, scaling occurs as the scalar product of them. And this is $s(w_t, w_c) = u_{wt}^T v_{wc}$. The model described here is a skip-gram and negative sampling model (Mikolov et al.,2013).

The purpose of creating a sub-word model is to take into account the internal structure of the word. Because in the vocabulary vectors obtained by the skip-gram and negative sampling model described above, the morphological structure of the words is not taken into account. In order to address the internal structures of the words, the scaling function s has been changed. Each word is treated as a character n-gram bundle, and the words boundaries are set at the end of each word with attaching <,> to the start and end. In addition, the word itself is also placed in the character n-grams. For example, for n = 3, the character n-gram of the where word is created as follows.

<wh, whe, her, ere, re> and <where>

Let's think of a character n-gram dictionary consisting of g n-grams. $\mathcal{G}_w \subset \{1, \dots, G\}$ Each n-gram g is denoted by the sum of the n-gram vectors to be represented by a z_g vector, to represent the n-gram set of the character in the word w. In this case, the scaling function (2.43) is expressed by the equation.

$$s(w, c) = \sum_{g \in \mathcal{G}_w} z_g^T v_c \quad (2.43)$$

By this simple model, the characters' n-gram representations can be shared between words. Thus, reliable representation of rare words can be obtained.

In order to measure the effectiveness and accuracy of FastText method, benchmark tests were conducted on similarity datasets. Spearman rank correlation method was used to measure the consistency of the method according to human responses given in the datasets. The word representations of each of the words on

the corpus were obtained by FastText, and the word vectors obtained by the method for word pairs in the data were calculated by cosine similarity. WS353 (Finkelstein et al. 2001) was used for English. For the French, Spanish, Arabic and Romanian languages other benchmark datasets were used. In the tests carried out on the WS353 data with the word vectors obtained from Wikipedia, 0.71 spearman correlation values were obtained. The contribution of this method to the continuous bag of words and skip gram is the use of character n-grams, and especially the morphological structure of this method is compatible with different languages.

For the NNLM and corpus-based similarity models, the results of the tests on the Common Crawl datasets of the Gigaword5 + Wikipedia2014 corpora and 42 billion words were tested and results in Table 2.1 is gained.

Table 0.1. Semantic similarity benchmark test results of popular NNLM method (Pennington et al., 2014)

Model	**Dimension**	**Word Count**	**WS353**	**RG65**
GloVe	300	6 Billion	0.658	0.778
CBOW (Word2Vec)	300	6 Billion	0.572	0.682
Skip-gram (Word2Vec)	300	6 Billion	0.628	0.697
SVD-L	300	6 Billion	0.657	0.751
GloVe	300	42 Billion	0.759	0.829
SVD-L	300	42 Billion	0.740	0.741
CBOW		100 Billion	0.684	0.754

According to Table 2.1, it is seen that GloVe method gives more consistent results than CBOW and Skip-gram. Although increasing the number of words seems to have a positive effect on the results, it has been observed that the increase in the number of words in the CBOW method did not give the desired contribution to the results.

3. MATERIALS AND METHODS

In this section of the thesis, materials and methods used in the studies are explained in details.

3.1. Materials

3.1.1. Princeton English WordNet

WordNet (Fellbaum, 1998) is a semantic ontological linguistic network includes semantic concepts and relations between them generated for the English language. It is developed by a group of linguist and computer science experts in Princeton University since the mid-1980s. Each concept defined on it has a definition in the English dictionary and these concepts are called Synset. In each Synset there are words representing that concept. Therefore, a word defined in WordNet can be seen on more than one Synset. In fact, Synset represents each of the definitions of words in the dictionary. The first feature that distinguishes the normal dictionary and WordNet is that the dictionary takes the words as a reference and defines the concepts, and WordNet takes the concepts as a basis and presents the words represented by the concepts in the concepts. Another feature is that WordNet associates concepts with each other, which is perhaps the most important feature that separates WordNet from the dictionary. All the concepts defined in WordNet are linked by 26 different types of relationships with each other. Each concept can be connected to one or more relationships with the other concepts. With this structure, WordNet features a complete graph network. Because all the concepts defined on WordNet are associated with at least one other concept.

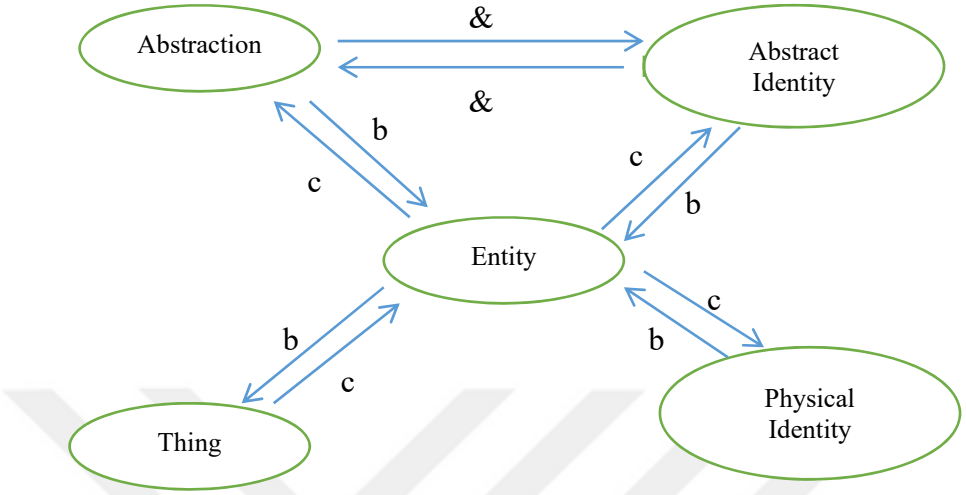


Figure 0.1. The appearance of the first concept defined on WordNet and its relations in graph model

Since WordNet is in graph structure, nodes are representing synsets and edges are representing the relations. Figure 3.1 shows the first synset in the center and adjacent synsets and their relations which are mapped in a graph model. Relation types are coded as; b: Hypernym, c: Hyponym, &: Synonym.

WordNet can be used online or as a database for researchers. The necessary data files can be downloaded and used from the official WordNet page of Princeton University. Statistics of the WordNet version 3.1 are given in Table 3.1 as; number of synsets according to POS (Part of speech), number of unique words and total word and sense pairs are given.

Table 0.1. WordNet 3.1 Statistics

POS	**Unique Strings**	**Synsets**	**Total Word-Sense Pairs**
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

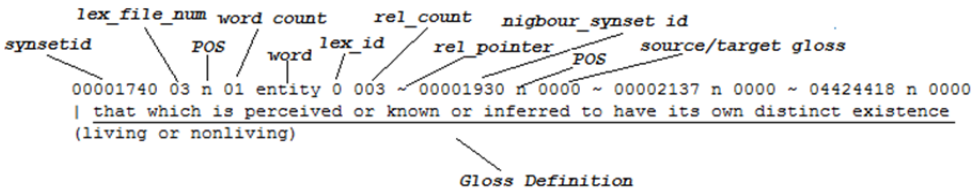


Figure 0.2. Information about a snyset record given on the WordNet data file and descriptions of these fields

WordNet data files are created separately according to the POS (part of speech). Only synsetid information is global and covers all files and consists of unique values. Figure 3.2 shows the structure of record given in a data file of WordNet. This record corresponds to a synset. Each of the defined concepts is known as synset on WordNet has an id (SynsetID) and a POS (Part of Speech) information. In addition to this, the dictionary meaning is also defined on this record. Other information for each synset record can be found on the Figure 3.2 one by one.

WordNet is used directly or indirectly in many natural language applications. Especially the relational structure and the fact that it is compatible with the graph network attracts the attention of those dealing with computational sciences. As mentioned in previous chapters, particularly successful studies were carried out on the graph network.

Table 0.2. Relation types defined in WordNet and example

Relation	**Relation**
Antonym (black->white)	Derivationally_related_form (intersection-> intersect)
Hypernym (lunch->meal)	Domain_of_synset_TOPIC (cell->biology)
Hyponym (meal->lunch)	Member_of_this_domain_TOPIC(biology->cell)
Instance_Hypernym(Genome_Project -> project)	Domain_of_synset_REGION(origami->Japan)
Instance_Hyponym(project->Genome_Project)	Member_of_this_domain_REGION(Japan->origami)
Member_Holonym(Defense_Secretary->US_cabinet)	Domain_of_synset_USAGE(mark->figure)
Member_Meronym(US_Cabinet->Defense_Secretary)	Member_of_this_domain_USAGE(figure->mark)
Substance_Holonym(paving->street)	Entailment(snore->sleep)
Substance_Meronym(street->paving)	Cause(relax->slow_down)
Part_Holonym(rebound->basketball)	Also_see(wash->wash_up)
Part_meronym(basketball->rebound)	Similar_to(behave->pretend)
Attribute(commerce->commercial)	Verb_Group(do->pretend)
Pertainym (annoyingly->annoying)	Participle_of_verb (applied->apply)

WordNet relation types defined in WordNet 3.1 and an example for each relation types are shown in Table 3.2. 26 different types of the relations are defined. Most of them are bi directional relations (like Hypernym<-> Hyponym).

A study has been done to create WordNet in different languages from English. One of the most important of these is Euro WordNet. Developed with support from the European Union, this word network covers a number of languages used in Europe. Structurally exactly same as Princeton WordNet, also it is liked to Princeton WordNet using Inter Lingual Index.

Therefore, each concept on Euro WordNet (Vossen, 1998) is associated with its equivalent on Princeton WordNet. BalkaNet was developed as part of the Euro WordNet project. Within the scope of the BalkaNet project, WordNet was developed for the languages spoken in six different Balkan countries (Turkish, Romanian, Bulgarian, Greek, Serbian and Czech).

3.1.2. The First Turkish WordNet: BalkaNet

BalkaNet (Bilgin et al., 2004) is known as the oldest Turkish WordNet. It is far behind Princeton WordNet in terms of the number of concepts it contains and the type of relationship it contains. The project started to be developed in 2001 and ended in 2004. While starting the BalkaNet project, a certain number of common concepts have been selected among all languages and a network has been developed with other concepts in which these concepts are related. The synonym, antonym, hyponymy relationships in Turkish WordNet were obtained by using the Turkish Dictionary automatically.

As shown in the Fig. 3.3, each record in the xml file of the BalkaNet is the synset (concept), each synset has an ID, id label also gives information about properties of that synset. POS tag gives information about part of speech of that synset. SYNONYM tag contains words that express this concept; LITERAL tag in SYNONYM tag gives each word and its sense rank. ILR tag gives information about the relation of this concept with other concepts; ID of the relational concept

is given here with relation type. DEF gives the definition of the concept, BSC gives information about which Base Concept this synset belong to.

```
<SYNSET><ID>ENG20-09819657-
n</ID><SYNONYM><LITERAL>radikal<SENSE>1</SENSE></LITERAL><LITERAL>köktenci<SENSE>1</SENSE></
LITERAL></SYNONYM><POS>n</POS><ILR>ENG20-00006026-
n<TYPE>hypernym</TYPE></ILR><STAMP>orhanb 2004/06/04</STAMP></SYNSET>
```

Figure 0.3. A synset sample of the BalkaNet in XML format

Most of the synset's in Turkish WordNet is associated with English WordNet 2.0. On Balkanet Turkish WordNet, there are only 15 thousand concepts and 21 thousand words representing these concepts and 11 different relationship types that connect them together. On this WordNet, whose conceptual coverage is quite narrow and the number of relations is insufficient, several semantic studies including semantic similarity (Dehkharghani et al., 2016; Tulu and Orhan 2017; Yucesoy, 2007) has been carried out to date.

3.1.3. RG65 (Rubenstein and Goodenough, 1965) Dataset

Widely used word similarity dataset of the English in the benchmark of the semantic relatedness studies. It is consist of 65-word pairs and 48 individual words with POS as a name. Word pairs were evaluated individually by 51 different undergraduate students and each was given a similarity score between 0-4. 0 indicates that two words are not identical to each other, while 4 indicate that two words are synonymous words. In terms of the choice of words, it is considered that there is a distribution which has a high probability of being synonymous, which has no relation to each other, that is, from full similarity to zero similarity.

3.1.4. WS353 (Finkelstein et al., 2001) Dataset

WS353 consists of 353 pairs of words and separated into two groups. In the first group, there are 153-word pairs and these word pairs are scored by 13

different people. Pairs of words were scored between 0-10. 0 shows that two words have had no relation at all, and 10 with two words are synonymous. In the other group, there are 200 pairs of words and these pairs are scored by 16 different people. Scoring was performed in the same way as the other one. It consists of a total 434 individual words, not only lexical but also special names which are not in many different dictionaries (Maradona, FBI) have been used. In addition, the POS information of the words was from different POS types. For example, in terms of relevance, the pairs of words in the first three and last three are shown in Table 3.3;

Table 0.3. The highest and lowest semantic word pairs in relatedness score in the WS353 dataset (given semantic similarity score out of 10)

Top three pairs	**Similarity Score**	**Last three pairs**	**Similarity Score**
sun sunlight	9.54	drink ear	0.38
automobile car	9.54	month hotel	0.31
river water	9.54	professor cucumber	0.23

3.1.5. MEN3000 (Bruni et al., 2014)

It is separated into train and test sets and used as a benchmark dataset in the semantic similarity tasks. It consists of 3000 pairs of words and word pairs are interpreted by the native English-speaking people and similarity scores are given by them. It is a combination of the words which are used at least 700 times in the collection of the ukWaC, Wackypedia, ESP-Game ve MIRFLICKR-1M sources. Words are from all type of POS, words in the pairs can have the same POS or different POS. It consists of 3000 different pairs of words, the majority of which are names. 2000 of the word pairs defined in the MEN data set were prepared for training and 1000 for test purposes. Word pairs were asked to 50 different people and to evaluate whether two words were related to each other or not. If it is related, score 1 is given, otherwise, score 0 is given to the pairs by each individual. These scores were summed and normalized over 50. For example, in terms of relevance, the first three and last three word pairs are shown in Table 3.4;

Table 0.4. The highest and lowest semantic word pairs in relatedness score in MEN3000 dataset (given semantic similarity score out of 50)

Top three pairs	score	Last three pairs	score
sun sunlight	50	muscle tulip	1
automobile car	50	bikini pizza	1
river water	49	bakery zebra	0

3.1.6. A Graph DB: Neo4J

Neo4J is developed by Neo Technology Inc, it is open source and the world's leading graph database management system. By its highly scalable performance and processing capacity, it provides the possibility of creating nodes and edges and making complex and deep inquiries among them very fast. It is used effectively in systems such as social networks that work according to the graph scenario. It consists of nodes and edges as the most basic data structure. An id is assigned to each node and edge automatically and all operations are performed in the background via these ids. It has it's own query language called cypher. It is very easy to store data and move it as graph db. It is possible to connect to the relevant database via its own web gui and to make the requested queries. Nodes and edges, which are the most basic data type, can be added to different data types. For example, a node may have many properties. There is also a type of node created. That is, the most basic data types are nodes and channels, and they have types, which can be random variables given by users. It can accommodate many different types of data in a node or edge. Neo4J provides API support for many programming languages. The API used in the database as a file to detect and graph-based operations on the file allows. It is possible to perform operations on graph db thanks to the API which is also integrated with Java language.

3.1.7. The Performance Measurement

Spearman rank correlation method is used in order to compare the success of the method against a given benchmark dataset. This method is widely used as a key performance and success indicator of the semantic similarity and relatedness

method. In the literature, in order to evaluate the proposed methods, result of this correlation is compared.

In the science of statistics, Spearman's rank correlation coefficient or Spearman's rho was named by the American statistician Charles Spearman, who first introduced this statistical measure. Math notation is often indicated by the ancient Greek letter ρ (pronounced rho). It is a non-parametric statistical measure and is used as a measure of the dependence between two variables, namely correlation. This means that Spearman's rho (ρ) coefficient makes no assumptions about the distribution of multiplicities for two variables, and how well the connection between these two variables can be described by any monotonic function. In order to calculate the ρ value, the sample data must be in sort order in two variables (Y and X). In general, this condition is not appropriate for sample data and the data is found in proportional scale or spaced scale or sequential scale without sorting, and in this case they are transformed into sorting order.

$$\rho = 1 - \frac{6\sum_{i=1}^{n}(X_i - Y_i)^2}{n(n^2-1)} \tag{3.1}$$

In the Eq. (3.1) Spearman Rank correlation (ρ) of the two sample distributions X_i and Y_i both of them has the same size (n). The correlation value is obtained between [0,1]. While 0 show the no correlation between the Xi and Yi, 1 shows the equal distributions between the variable among the X_i and Y_i.

On the other hand, according to our experience with artificial data analysis, we cannot guarantee that we can achieve the best results by just using Spearman correlation coefficient value. Therefore, we needed additional supplement. Thus, MAE (Mean Absulate Error) is chosen due to its ease of calculation and popularity. The mean absolute error is a measure of the difference between two continuous variables. The MAE is a linear score that measures the average magnitude of errors

in a series of estimates, regardless of their direction, where all individual errors are weighted equally on average. The MAE value can vary from 0 to ∞.

$$MAE=\frac{\sum_{i=1}^{n}|x_i-y_i|}{n} \quad (3.2)$$

In Eq. (3.2), x and y are two continuous variables and n is the number of samples in each variables.

3.2. Semantic Relatedness Measurement Methods

As explained in previous chapter, there are many different word level relatedness measurement methods exist in the literature. In this section, the methods that we try to generate are explained during the thesis study, first of the method is the semantic similarity measurement study using Turkish WordNet, it is supposed to perform semantic relatedness study using Turkish Words with widely used graph based semantic similarity calculation method that is called UKB. In the second method we tried to use an approach to weight semantic relations of the WordNet with multivariate linear regression approach. In the third one, we tried to get the weights of the semantic relations of the WordNet with different approach taking into account all the paths connection two words. All the methods we studied have not shown an acceptable performance in terms of the success against benchmark datasets. So the last method is generated as proposed method that backbones this thesis and it is explained in the chapter 3.3 as proposed solution.

3.2.1. PageRank Based Semantic Similarity Measurement Using Graph-Based Turkish WordNet (Tulu and Orhan, 2017)

It is a graph based semantic similarity study of Turkish words using WordNet (BalkaNet) with PageRank method. The study mainly used the application of UKB (Agirre and Soroa, 2009) and measured the semantic similarity

of the Turkish words using the Balkanet as a Graph-based WordNet. The BalkaNet Turkish WordNet file in the xml structure has been converted to standard WordNet format with an implemented java application. As a result, eight different files were created from the BalkaNet xml file, which was the data and index file for each pos. The resulting index and data files are standardized by converting to ukb format. The standard ukb files consist of .lex and .rel files, both are text files, and the lex file consists of each word and the corresponding synset id. The other one is the rel file, each line on this file consists of two synset and the types of relationships that connect them. Using these two files, the ukb application generates a binary graph.

This binary graph file can be used to calculate the similarity of each word pair defined in the graph. Dot product and cosine similarity options are offered for similarity calculations. The similarity calculation works as a script and takes two words from the standard input and calculates the semantic similarity of these two words as follows;

- Using the personalized PageRank method on the given graphical network, a probability distribution vector starting from the first word and circulating the whole network is created.
- Then for the second word, a probability distribution vector is created on the same graph network, this time starting from the second word and circulating the whole network.
- This is calculated by the similarity of the two vectors created, the vector similarity method given as a parameter.
- If this parameter is not given, cosine similarity is used by default.

In the UKB application, there is a WSD module that can be used to disambiguate the words.

In the study, the second stage is to test the similarity calculation for Turkish words. During the time that study was conducted, there was no generally

available Turkish Word Similarity benchmark dataset which was developed by native Turkish speakers. For this reason, the Turkish words correspondence of the RG65 data were obtained with the help of the experts. Since some of these words were not defined on BalkaNet, similarity calculations were made for 48 pairs of words instead of 65.

In the tests performed by 48 pairs of Turkish words on the RG65 data, 0.54 Spearman correlation value and 0.50 Pearson correlation value were obtained. Although, Results obtained from the benchmark tests were not successful, there some important points of this study as; it is a first word similarity applications using PageRank method on the graph based Turkish WordNet BalkaNet. By this study, word vectors of the Turkish WordNet were obtained and cosine similarities on the vectors were applied. This point would most likely to provide contributions to the literature of the Turkish semantic similarity studies.

3.2.2. Semantic Relationship Weights Determination on WordNet by Using Multivariate Linear Regression Method

In this study, the aim is to measure the similarity or semantic relatedness of the two WordNet concepts which are directly or indirectly related in the WordNet graph network. By weighting the semantic relations of the WordNet, it is assumed that the weights determined on the path with a path-based approach on the graph structure can be used to calculate the similarity of the two concepts by means of a mathematical model.

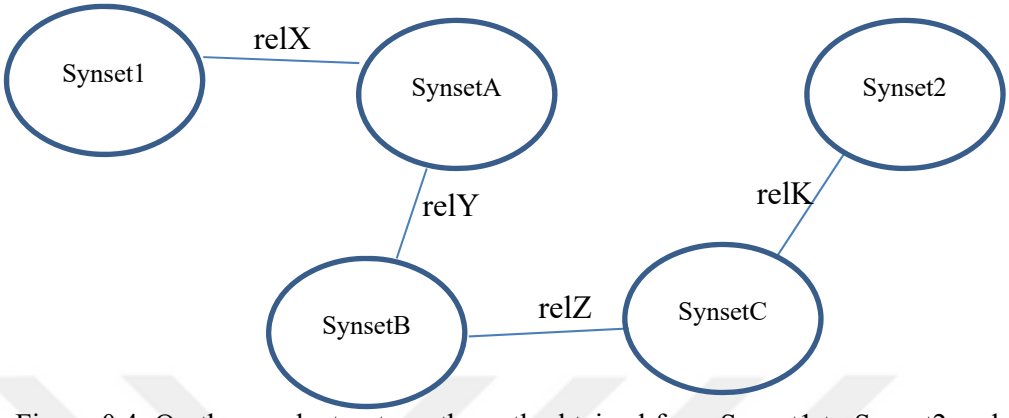


Figure 0.4. On the graph structure, the path obtained from Synset1 to Synset2 and the relations on it.

As it is seen in Figure 3.4, the semantic relatedness between Synset1 and Synset2 in a path-based approach is assumed as;

It is expressed as *Sim(Synset1,Synset2)=f(relX,relY,relZ,relK)*. This shows that semantic relatedness can be expressed by a function dependent on the type of relations between two synsets. If it is considered that the semantic relatedness value will be maximum 1, it is assumed that the relations between the two can be expressed in real numbers between 0 and 1. The first approach is based on the fact that the semantic proximity can be calculated by multiplying each relations weights on the path as shown in Eq. (3.3).

$$rel(synset1, synset2) = \prod_{i=1}^{len(R)} \mathcal{R}_i \qquad (3.3)$$

It is assumed that the semantic relatedness/similarity between the two concepts can be obtained by multiplying each relation weight on the path $\mathcal{R}$. And $\mathcal{R}i$ shows the weights of relationships on the shortest path between the synset1, synset2.

In order to calculate semantic similarity, the weight of each type of semantic relationship on the path between concepts must first be determined first. Previous studies have not observed a weighting for all semantic relationship types on WordNet. Instead, certain semantic relationships were grouped and priorities were given by classifying them. Sblini and Koseim (2013), in their study on this subject, they stated that all of the 26 different types of the relationship on WordNet can not be not included into a group with the same weight and stated that it would not be possible anyway. As shown in Table 3.5, they have grouped the WordNet relations and given priority to the each group. Relation group at the top of the table has top priority.

Table 0.5. Categorization of semantic relationships degree from top to down (Sblini and Koseim, 2013)

Category	**Semantic Relation Types**
Similar	Antonym, cause, entailment, participle of verb, pertainym, similar to, verb group
Hypernym	Hypernym, instance hypernym, derivationally related
Part	Holonyms (part, member, substance), inverse gloss, meronomy (part, member, substance)
Instance	İnstance hyponym, hyponym
Other	Also see, attribute, the domain of synset (topic, region, usage), member of this domain (topic, region, usage)

From this point on, the following approach has been presented in order to give weight to semantic relationship types in this study;

- Human judged real life semantic word similarity dataset is taken.
- Each word on this dataset, the corresponding synset in the WordNet graph structure is found.
- For each word pair in the data set, corresponding synsets in the WordNet graph structure is found.

- The relations on this path will be determined one by one, and since semantic similarity will be obtained by multiplying the detected relations as seen in (3.4), the transformation of semantic similarity into a linear regression problem will be as in Eq. (3.5);

$$S_{xy} = Sim(X,Y) = relX * relY * relK * relZ \quad (3.4)$$

$$\ln(S_{xy}) = t_1 \ln(rel_1) + t_2 \ln(rel_2) + t_3 \ln(rel_3) \ldots . + t_n \ln(rel_n) \quad (3.5)$$

If we generalize the expression above;

$$\ln(S_{xy}) = \sum_{i=1}^{n} t_i \ln(Rel_i) \quad (3.6)$$

In Eq. (3.6), Rel_i represents each type of semantic relationship defined on WordNet, so it should be taken as 26 because there are 26 different types of relationship. t_i is the number of times the Rel_i the relationship has found on the shortest path between (X, Y). If it has not found, zero is taken. As can be seen, the problem has turned into a linear regression problem with many variables. As a result of this problem, the numerical value of the index Rel_i will give the weight of the related relationship on WordNet as shown in Eq. (3.7).

$$\text{weight}(Rel_i) = e^{Rel_i} \quad (3.7)$$

The mathematical representation of the multivariate regression problem is as follows;

[Feature matrix]X[Relation Vector]=[output vector]

$$\begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{bmatrix} X \begin{bmatrix} a \\ b \\ . \\ . \\ z \end{bmatrix} = \begin{bmatrix} \ln(S_1) \\ \ln(S_2) \\ . \\ . \\ \ln(S_n) \end{bmatrix} \tag{3.8}$$

With the solution of the multivariate linear regression problem formed by the mathematical model given by the equation (3.8), weights are obtained. The semantic relatedness of any two concepts given on WordNet after obtaining the weights is obtained by the equation (3.3).

Algorithm of the the method is shown in Algorithm 1. Real life dataset, set WordNet relations (as R={"synonym", "Hypernym", "Hyponym"…} total 26 relations) and full WordNet graph Db are given to the algorithm as input. And weights of the relations are calculated with help of multivariate linear regression model.

Algoritm 1 *Determination of WordNet Relation Weights using Multi Variate Linear Regression Method*

Input: *T is a real life similarity dataset (like, RG65, MEN3000, Ws353), G WordNet as Graph DB, R is a set of WordNet relations*
Output: *P, WordNet Relations Weights Vector*

```
1:   P[R.lenght] ←0, OV[T.lenght], FM[T.lenght][R.lenght], i ←0
2:   For(1) each record (w1,w2,SimVal) Є T
3:     OV[i] ←ln(SimVal), j ←0
4:     S ←GetShortestPathFromGraphWithPriority(G,w1,w2)
5:     For(2) each rel(r) in R
6:       rc[i,j] ←countRelation(S,r)
7:       j ←j+1
8:     End For(2)
9:     i ←i+1
10:  End For(1)
11:  LR[R.Lenght] ←getMultipleLinearRegression(rc,Ov)
12:  For(3) i ←0 to R.lenght
13:    P[i] ← exp(LR[i])
14:  End For(3)
15:  Return P
```

Using the semantic weights that will be obtained by the multivariate linear regression method given by the mathematical model above, the following steps have been accomplished one by one when designing the system which will perform the similarity calculation between the synsets on the graph structure.;

- English WordNet (3.0) is downloaded.
- All data files on the downloaded WordNet (data.noun, data.adj, data.verb, data.adv) are read one by one with the help of a java application, since each record on the files represents a synset, for each record, A node has been created in the Neo4J Graph database.
- The words of that synset contains were added to each node as a feature.

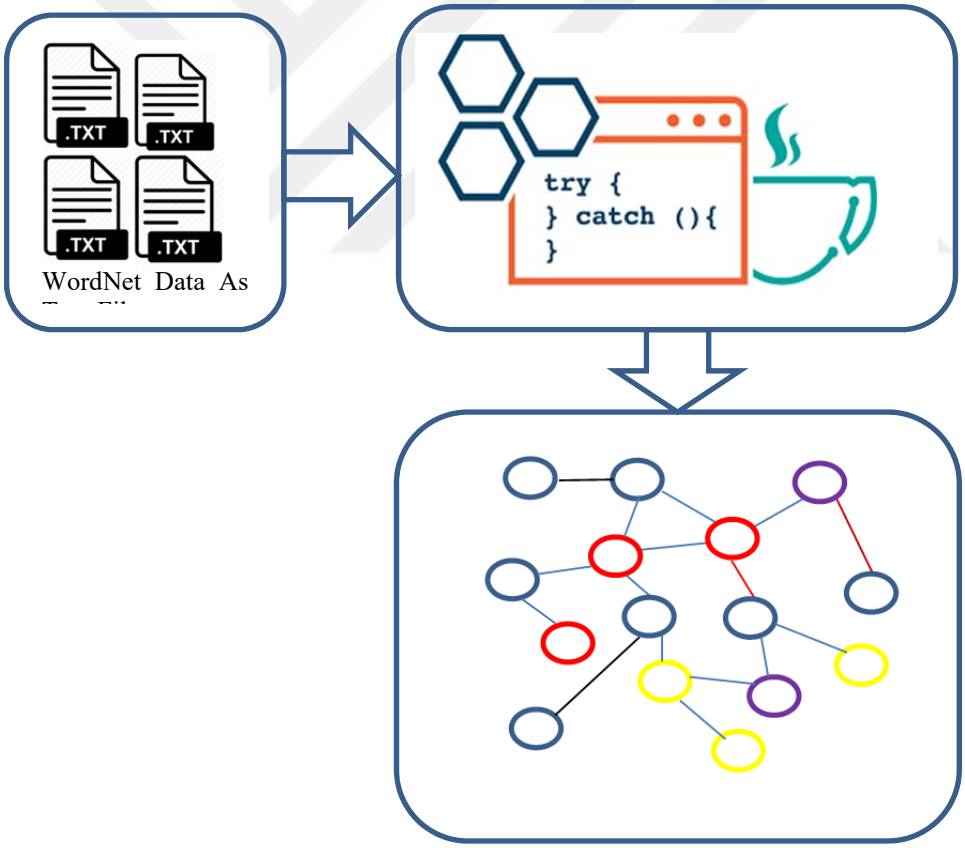


Figure 0.5. Transformation of WordNet into Graph Structure

Files with the .data extension that come with WordNet are read row by row via java application. Since each row represents a synset, a node is created in the graph database for that row. Synset id information and word information obtained from WordNet was added to each node. In this way, the node_id value of each node created on the Neo4J graph database is retrieved and kept on the variable of a list type on the applications in Table 3.6

Table 0.6. The structure of the list data (NodeList) where the synset_id of the nodes and id of the nodes are retained when creating the Neo4J Graph database

Node_id1	SynsetId_1
Node_id2	SynsetId_2
……	…….
…...	……
Node_idn	SynsetId_n

In the list data maintained by the NodeList variable, a list of data of a total length of 117K was created because a node was created for each synset. In order to establish the relations between the nodes, the records in the data files were read and by the way source, destination synsets and relation type were determined. In order to attract this information and to determine the types of relationships, shared reference documents were used with WordNet. For each relationship obtained from the source synset to the target synset, the directed relationship has been established. The node_id values provided by graphs of source and target synsets to generate directional relationship were obtained from NodeList list data. In this way, a total of 771K relations among nodes were formed on the graph database.

On generated WordNet graph db, all word pairs on the MEN3000 dataset are found on which word corresponds to synset, if any word in a given word pair is represented by more than one synset, then the ambiguity must be eliminated. ADW (Pilehvar and Navigli, 2015) method was used to solve semantic ambiguity. Accordingly, the synsets represented by each word in a matched word pair are

compared with each other, and semantically the most closely related synsets are assumed to represent the corresponding word pairs and accordingly the problem is resolved. In path-based methods using graph structure, the shorter the path length of the two concepts to each other means the more semantic relation to each other is assumed. Elimination of semantic ambiguity is performed using the approach detailed in Algorithm 2.

Algoritm 2 *Word Sense Disambiquation Algortihm*

Input: *Word1, Word2 and G WordNet as Graph DB,*
Output: *synsetID1, synsetID2*
1: Get SynSetList of Word1, Word2 from G as L1and L2
2: synsetID1 ←L1.get(0), synsetID2 ← L2.get(0)
3: Get shortest path using synsetID1and synsetID2 as SP
4: ***For(1)*** *each synset(s1) in L1*
5: ***For(2)*** *each synset(s2) in L2*
6: Get Shortest Path between s1 and s2 as CP
9: ***If(1)*** *CP <SP*
7: SP ←CP, ***synsetID1*** *←s1, synsetID2 ← s2*
8: ***End if(1)***
9: ***If(2)*** *CP =SP*
10: Calculate path priority of CP as p1 and SP as p2 using ***priority*** *function*
11: ***If(3)*** *p1>p2*
12: SP ←CP, synsetID1 ←s1, synsetID2 ← s2
13: ***End if (3)***
14: ***End if (2)***
15: ***End for(2)***
16: ***End for (1)***
17: ***Return*** *synsetID1, synsetID2*

One of the few points to be considered in the algorithm is that if the path length is equal for two different synset pairs, then the priority values of the paths are calculated and the path with the lowest priority value is taken. When calculating Priority, each relationship on the path is assigned a value to a relationship group and accordingly the priority value of a path is calculated. Priority calculation algorithm is shown in Algorithm 3.

Algoritm 3 *Priority determination of a path*

Input: *shortest path between two words as path*
Output: *priority value P*
1: *P←0*
2: ***For*** *each relation (r) in the path*
3: *P←p+getRelationPriority(r)*
4: ***End For***
5: ***Return*** *P*

The priority values for each type of relationship are given in Table 3.7. Accordingly, the relationPriortiy (rel) function returns the priority value for the corresponding relationship type. According to the relationship type priority assignment, Sblini and Kosseim (2013) proposed by the weight of the relationship types are made on the basis of the order. Relation types and corresponding priority are given on the Table 3.7, priority values are assigned using the study of the Sblini and Kosseim (2013).

Table 0.7 Relationship types and priority values defined on WordNet (Sblini and Kosseim, 2013)

Relation Type	**Priority**	**Relation Type**	**Priority**
Synonym	1	Substance_Holonym	4
Antonym	2	Substance_Meronym	4
Entailment	2	Part_Holonym	4
Cause	2	Part_meronym	4
Verb_Group	2	Hyponym	5
Similar_to	2	Instance_Hyponym	5
Participle_of_verb	2	Attribute	6
Pertainym	2	Domain_of_synset_TOPIC	6
PertainedBy	2	Member_of_this_domain_TOPIC	6
Hypernym	3	Domain_of_synset_REGION	6
Instance_Hypernym	3	Member_of_this_domain_REGION	6
Derivationally_related_form	3	Domain_of_synset_USAGE	6
Member_Holonym	4	Member_of_this_domain_USAGE	6
Member_Meronym	4	Also_see	6

The weights obtained when we run the Algorithm 1 with the MEN3000 dataset are shown in Table 3.8.

Table 0.8. Semantic Relationship Weights by Multivariate Linear Regression Method

Relation Type	Symbol	Weight
Antonym	a	0.70
Hypernym	b	0.82
Hyponym	c	0.85
Instance_Hypernym	d	0.85
Instance_Hyponym	e	1.07
Member_Holonym	f	0.88
Member_Meronym	g	0.83
Substance_Holonym	h	0.85
Substance_Meronym	i	0.87
Part_Holonym	j	0.85
Part_meronym	k	0.83
Attribute	l	0.86
Derivationally_related_form	m	0.79
Domain_of_synset_TOPIC	n	0.73
Member_of_this_domain_TOPIC	o	0.80
Domain_of_synset_REGION	p	0.74
Member_of_this_domain_REGION	q	0.78
Domain_of_synset_USAGE	r	0.85
Member_of_this_domain_USAGE	s	0.62
Entailment	t	0.65
Cause	u	0.84
Also_see	v	0.91
Verb_Group	x	0.65
Similar_to	w	0.85
Participle_of_verb	y	0.85
Pertainym	z	0.89

The weight value obtained for the Instance_Hyponym relationship type is 1.07. Although our expectation is below the value of 1, it is considered that the variables with a very low number of samples in multivariate linear regression problems might such an effect. Using the obtained weights, validation tests on the MEN3000 data were obtained with a value of 0.68 Spearman rank correlation, which is well below than the expected value. Similarly, 0.74 correlation was obtained in the comparison test performed on the RG65 dataset.

Although the results obtained from the tests look not successful, a few issues are noteworthy. The first of these is to calculate the similarity of the two concepts, taking into account their position on the graph by using the shortest path is not enough alone, perhaps all of the paths should be taken into account. It has been concluded that a new relationship weight determination method can be studied by taking into account all the types of the connecting paths and all the types of the relationship on the paths.

3.2.3. Determination of Semantic Relation Weights on WordNet (Tulu et al., 2018)

While the results obtained by the method described in 4.4 cannot achieve the expected success, in determining semantic relations weights and calculating the semantic similarity, it is necessary to take into account all the paths that link the two concepts together, so that it might be concluded that the shortest path alone cannot be sufficient. As the operational cost of handling all the paths in the semantic similarity measurement be very costly, the idea of handling all the paths of certain lengths linking the two concepts on the WordNet graph structure is found to be more feasible and applicable. In this study, WordNet 3.0 and MEN3000 data set were used together and a new approach was given for the WordNet-based semantic relationship weights determination methods.

The main reasons for the use of MEN3000 dataset in determination semantic weights are as follows; List of the samples are created by experts, judgments of the word pairs are evaluated by native English speakers, it includes enough number of word pair samples in the dataset, it represents real life. On the other hand, the use of WordNet, a mature semantic well-defined word network developed by experts is the strength of the study in the semantic weight determination. In order to measure the success of the proposed method, RG65 similarity dataset was used for benchmark and 0.81 Spearman correlation was gained by the way. The use of a large dataset as the MEN3000 for development

and testing and the use of a common and important dataset as the RG65 for the purpose of benchmarking the weights of semantic relations and calculating the semantic relationship brings a different perspective to the calculation of semantic similarity and semantic relationship. Although the result obtained is not the best result in the literature, it is suggested that the method will bring a different perspective to the semantic similarity measurement method since the simplicity of the method includes rather statistical and arithmetic operations instead of complicated mathematical operations.

In the previous studies, it was observed that there are two important factors affecting the similarity in the WordNet based approaches. These are path length between two concepts and relation types that constitutes the path. And also it is widely known that path length and semantic similarity has a negative correlation. So this means that longer path length means a smaller similarity. It is known that the shortest path between concepts is an important parameter for similarity but this is not enough alone. It is thought of as a different aspect of this study that all paths can contribute to semantic similarity, not the shortest path, because of the daily uses of concepts and the reasons originating from cultural differences. It is supposed that human mind models the similarity of the two concepts by taking into account all the combinations links connection them. From this reason, the WordNet 3.0 graph is used in the study, graph consists of 117 nodes and over 700K edges. Initially, it was supposed to use of all the paths connection two pairs in the MEN3000 dataset. Since this will bring huge computational cost, then it was decided to get all the paths with a defined length. By the way, this approach was accepted as; firstly find the shortest path between two concepts then get all the paths with max length (shortest path)+3. +3 is the maximum length to collect the paths from WordNet between two synsets. This value can be choosen greater, but in this case query time is getting increased. Algorithm of this approach is shown in Algorithm 4.

Algoritm 4 *Get all the paths between word pairs in whole Dataset*

Input: *Dataset D, Max Path Lenght Pmax, WordNet Graph as G*

Output: *list of the paths PL*

1: *Initialize Path List PL←Ø*

4: ***For(1)*** *each record(W1,W2,SimVal) in D*

5: *c1,c2←GetNodesFromWordNet(G,W1,W2)*

6: *N←GetShortestPathLenght(G,c1,c2)*

9: *AllPath←FindAllPath(G,c1,c2,N+Pmax)*

7: *PL.add(AllPath,SimVal)*

8: ***End for(1)***

9: ***Return PL***

With algorithm 4, each path is assigned the same similarity score, as shown in the Figure 3.6, *acdfhij=0.74*, *acdgik=0.74* and *bdgj=0.74*.

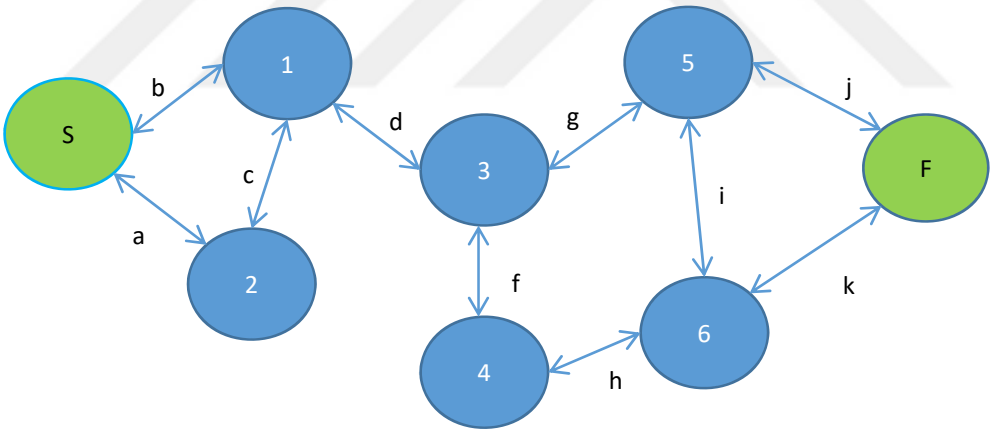


Figure 0.6. Visualisation of geting all paths between a word pair

With the function given above, 3000 path lists are obtained for 3000 pairs of words. As a result, 3000 similarity scores for 3000 pairs of words and 400K different paths to which these scores are assigned are obtained. The same path and different scores of these paths are combined and grouped together with the

algorithm given below. For example, *bdgj=0.74* for a word pair, *bdgj=0.60* in another word pair, *bdgj=0.55* in another word pair, and values of these paths are converted to a single value by taking its mod or median.

On the other hand, although the shortest path is not sufficient in calculating the semantic similarity, it is clear that the importance of short paths in the similarity of the two concepts is much more than the long paths. In addition, the number of short paths between the two concepts is very small, although a large number of the long path is obtained. For this reason, short paths are more than long paths, and similarity values should be considered in this case. For this reason, the Importance factor is proposed to give more importance to the shorter paths.

$$ImF_i\big(getAllPath(w_1, w_2)\big) = n^{n-k_i} \tag{3.9}$$

In Eq. (3.9), ImF (Importance factor), the path linking the two concepts can be considered as a weight value given to each path in the path list. *n* is the length of the longest path in the path list of the given words. *k* is the length of the current path. For example, w_1 = *"bedroom"*, w_2 = *"kitchen"* the whole path list is obtained as *findAllPath ("bedroom", "kitchen")=[bcjk, bcjk, bcjk, bcjk, bcjk, bcjk, jkbc, jkbc, jkbc, jkbc, jkbc, bc, jk]*. According to MEN3000 data, the similarity of these two words is 0.6. All the paths obtained here are assigned a value of 0.6.

Table 0.9. As an example, w_1 = "bedroom", w_2 = "kitchen" for the words obtained and the importance factor of these paths with the assignment of similarity value (Tulu et al., 2018)

Path	**Occurrence**	**Impartance Factor**	**Value**
bcjk	6	1	0.6
jkbc	6	1	0.6
bc	1	16	0.6
jk	1	16	0.6

When creating a path list for each word pair, the path list is added up to the importance factor value of the relevant path as shown in Table 3.9. The aim here is to take into account the importance of the short paths after the addition, which contributes to the greater similarity than the longer paths. Algorithm 4 is used to group paths and give weight to each path;

Algoritm 5*Calculate Weights of the Each Path Retreived from All Word Pairs in Dataset*

Input: *All paths retrieved from dataset as AP*

Output: *Calculated Path Weights according to Mean and Median as RWmean and RWmedian*

```
Initialize Path list PL ←Ø
For(1) each record(PathList, PathVal) in AP
    For(2) each path in PathList
        If path exist in PL update PL(path)
        If there is no path in PL, create PL(path)
    End for(2)
End For (1)
Initialize Weight list RWmean ←Ø, RWmedian ← Ø
For(3) each record(path,valList) in PL
        Calculate mean and median of the values in the valList as mn and mdn
        RWmean.add(path, mn),RWmedian.add(path, mdn)
End for(3)
Return RWmean, RWmedian
```

According to the algorithm 5, the paths obtained for each word pair on the MEN3000 dataset is handled one by one, and the values given to each path are assigned to a separate list and the value of each path is determined by subjecting the mean and median operations on this list.

In the result path list, path length = 1 (single-length paths) are the values we now obtain for the weight of each relationship. Not all of the WordNet relationship weights were obtained in this way. Because the singular paths of some semantic relationship types have never been seen on the MEN3000 data, below explained method has been applied to overcome this situation.

The shortest paths or paths containing any *x* relationship can be determined, and the weight type of the relationship type which cannot be weighted

is obtained by using the relationship types. For example, the shortest path *xa* with *x* relationship is *xa*, if the resulting weight is 0.5 and the known weight of *a* is 0.7, then *x=0.5/0.7=0.71* is obtained. If a relationship type was never seen on the MEN3000, then, in this case, the assumption of 0.85 is given for the relevant weight based on the edge counting technique developed by Yang and Powers (2005). As a second step on the method, the obtained weights were applied to all the paths of each word on the MEN3000 data and the values obtained were taken as closest to the MEN3000 real human assessment. For the weights which were taken from these paths, initially unique weights were obtained by the weighting of the method mentioned above and the obtained weights were shown in Table 3.10 for both the first and second stages.

Table 0.10. Relation weights retrieved in first two steps (Tulu et al., 2018).

Relation Type	Rel. Code	Determined Weight	
		Step1	Step2
Antonym	a	0.70	0.70
Hypernym	b	0.80	0.82
Hyponym	c	0.76	0.76
Instance_Hypernym	d	1.39	1.36
Instance_Hyponym	e	0.85	0.85
Member_Holonym	f	0.97	1.14
Member_Meronym	g	1.05	0.85
Substance_Holonym	h	1.07	1.10
Substance_Meronym	i	0.99	0.85
Part_Holonym	j	0.83	0.84
Part_meronym	k	0.78	0.80
Attribute	l	0.95	0.98
Derivationally_related_form	m	0.84	0.86
Domain_of_synset_TOPIC	n	0.90	1.02
Domain_of_synset_REGION	p	0.75	0.90
Member_of_this_domain_REGION	q	0.85	0.85
Domain_of_synset_USAGE	r	0.77	0.86
Member_of_this_domain_USAGE	s	0.85	0.85
Entailment	t	0.79	0.79
Cause	u	0.78	0.72
Also_see	v	0.82	0.95
Verb_Group	x	1.16	1.12
Similar_to	w	0.88	0.89
Participle_of_verb	y	1.44	0.85
Pertainym	z	1.48	1.30
No_Path_Determined*	%	0.21	0.21
Synonym*	&	0.84	0.79
PertainedBy*	!	0.85	0.85

Here, No_Path_Determined means that minimum path length is more than 10 between those two concepts. Synonym relationship, there is no such a relationship type defined in WordNet, it represents two concepts that are identical to each other and this relation type is generated by the algorithm. The PertainedBy relationship was also produced in reverse of the Pertainym relationship by an algorithm based on the principle that there was also an absolute inverse of each relationship, although not found in WordNet. Some rarely used relation types have weights more than 1.0, because they are used rarely in the long relational paths.

In the development of the proposed system, Neo4J was used as graph db and Java was used as a development environment. WordNet 3.0 data was transferred to graph db with prepared java program, raw data was converted from WordNet 3.0 to Graph Db then the nodes and edges were created by this program. The cypher query language on the Neo4J graph db is used to determine which of the nodes correspond to the synset and to find the desired length and/or shortest paths between the nodes. After determining the weight of each type of relationship, tests were performed on the RG65 benchmark dataset. RG65 data on each pair of the word belonging to the nodes were determined by querying the paths between them, then the paths were grouped and sorted according to importance factors. Afterwards, the weighting of the paths obtained for each word pair and their consolidation were performed.

Table 0.11. Retrieved spearman correlation values for RG65 in both step 1 and 2 according to the mean and median of the path (Tulu et al., 2018)

	Correlation using means of the path value list	**Correlation using the median of the path value list**
Relation weights determined in first step	0.81	0.78
Relation weights determined in second step	0.79	0.74

As shown in Table 3.11, we have obtained 0.81 Spearman correlation values by applying the weights obtained in the first step to the paths obtained by taking the mean of the path value list. The value obtained is above the average of the literature, although not high, and is noteworthy. Because the method is statistical and mathematical complexity and the processing load of the method is less. As a result of using the weights obtained at the second level, 0.79 Spearman correlation was obtained. The Spearman correlation values obtained with the statistical mean for the first and second stages were higher than those obtained with the median. The method is apparently a simple method, the only time consuming and processing part is the process of obtaining all the lengths of a certain length between two concepts, which is not a continuous process, it is enough to be done once. Especially, it is easy to obtain all paths of the length on the graph database or to obtain the shortest paths. The process of obtaining all the paths of a certain length of 3000 pieces lasted an average of 10 hours. Another result is the assumption that, together with other real-life data, these results can be synthesized and more optimum weights and better benchmarking values can be obtained. In addition, not only the path length but also other graph metrics can be used on the graphical network, such as the lowest common subsumer (LCS). The values obtained on the benchmarking data, the simplicity of the method, the low cost of processing, and the fact that this method can be improved further while testing the data on different datasets can also be performed by the way.

3.3. Proposed Method: SemSpace

In this study, word representations are emphasized and a new word representation method is studied. In the literature review chapter, we have seen the methods that create dense word vectors called word embedding's from the large corpuses using feed forward neural networks these methods can be Word2vec (Mikolov, 2013), GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017). SemSpace method is based on the dense representation of words in low-

dimensional space. The use of both WordNet and word representations in the approach and not using the corpus is among the most unique aspects of this method. The process can be summarized as the Euclidean distance based learning approach in obtaining the word representation vectors. While WordNet 3.0 was used for the training part of the method, the RG65, WS353, and MEN3000 benchmark datasets were used for the test. In addition, another important aspect of the method is that it allows new word pairs and similarity scores to be obtained in WordNet. By this method, WordNet is transferred to the graph structure, all words are considered as independent nodes out of synset structure. This study also represents that WordNet relations can be divided into three different categories and weighted accordingly.

The term prototype, defined as an example representing any model, is referred to as vectors representing the problem or data samples in some machine learning methods. In our study, each word is considered as a prototype in the hyperspace, and the relations which represent the similarity between two words are counted as the equation with two unknowns. When the relationship between the two words as an equation between the two unknowns can be said that an infinite number of solution can be found. To show this as a graph model, we can evaluate each variable as a node and the relationship between both variables as an edge like in Figure 3.7.

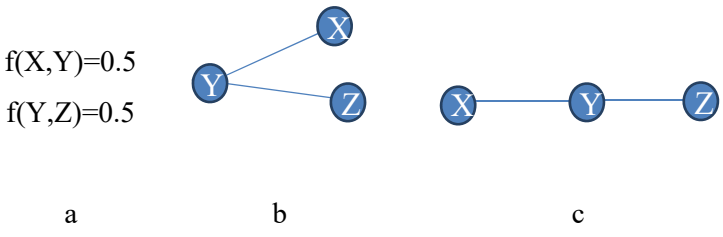


Figure 0.7. Graph model of two possible solutions with two unknowns

Accordingly, both graphical models presented in (Figure 3.7.b and 3.7.c) may be the solution for the equation given in Figure 3.7.a. *F(A,B)* shown in Figure 3.7 represents the relatedness, i.e. the relatedness between the words *A* and *B* as a function. By the way, two of the possible equations in two-dimensional space is chosen for the two equations given in Figure 3.7.a. Difference on solutions in Figure 3.7.b and Figure 3.7.c are quite remarkable, but both of them are possible solutions. In order to talk about the unique solution in this problem, it is necessary to form a full K3 graph with equations. Otherwise, even if we ignore rotation versions, we can say that there are numerous solutions in only two-dimensional space. On the other hand, even when the size of the hyper-space increases or geometric angles between the nodes is changed, solution models can be rotated in space to create an infinite number of solutions. To be able to use the expression "the best single solution" in such problems, it is necessary to create a full graph with each node in the system, which is not possible in many real-world problems. Therefore, the best thing to do without a sufficient number of equations is to seek a successful solution according to the objective function. In order to develop proposed algorithm, it is inspired from Learning Vector Quantization method (Kohonen, 1995). General steps of the proposed iterative learning algorithm can be found in Figure 3.8.

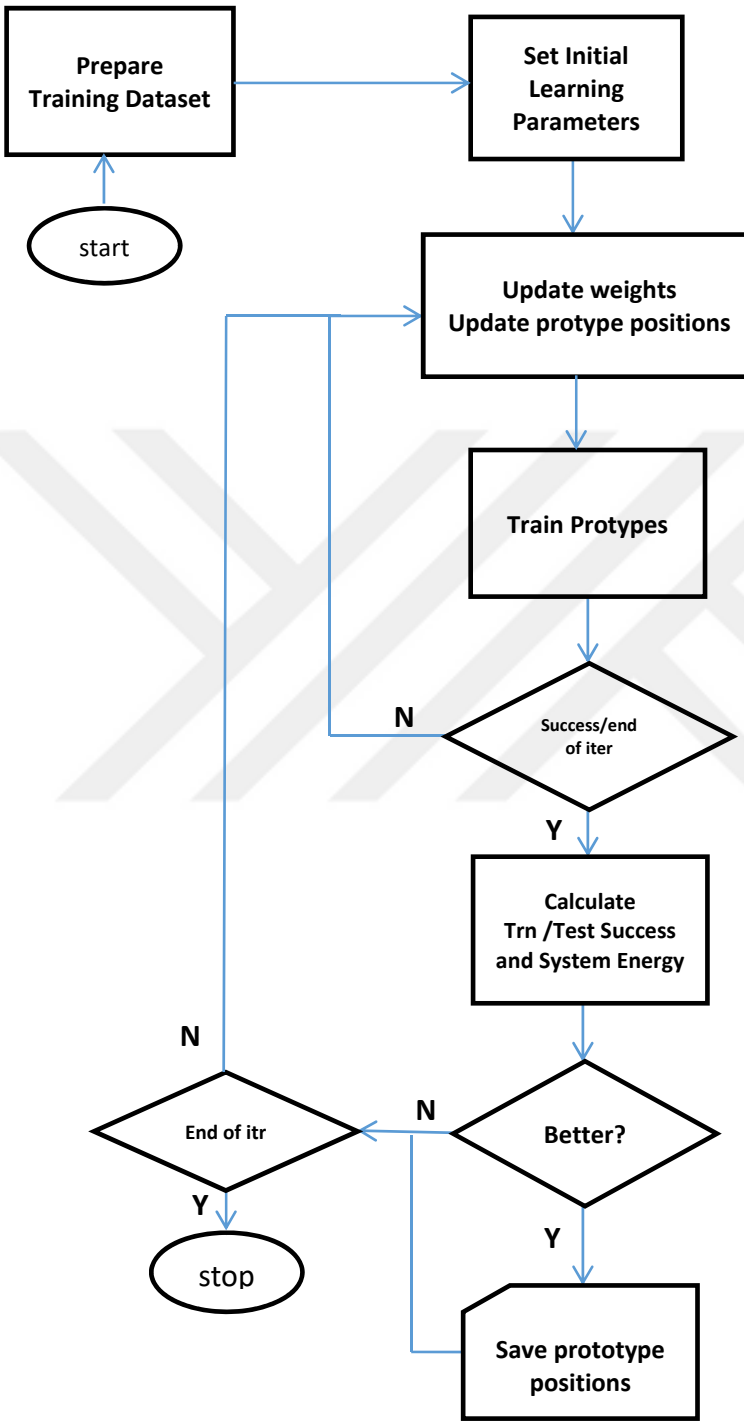


Figure 0.8. The flowchart of SemSpace Method

The algorithm acts as a search algorithm (like genetic or bee colony algorithms) also by random additions and therefore needs to be repeated many times. For machine learning, data mining, and many more, there are many methods that treat each instance of the current data set as a prototype and act according to the distance between prototypes. Theoretically, it is assumed that the system will produce the highest correlation and the lowest deviation when the prototypes find the most suitable position. The algorithm needs training and test sets separately. For the test set, data sets such as RG65, WS353 and MEN3000, which are frequently used as benchmarks in the literature, were used. WordNet relations are planned for the training set. However, the word pairs obtained from WordNet cannot be used directly in the study because they are not identified with each other as a numerical similarity but with directed relations. This problem has been tried to be overcome by defining the weights between [0 1] and the relationship types defined in WordNet. In order to determine the optimum values of these weights, a search approach was used.

3.3.1. Training Dataset Generation

The first step of the algorithm, initially, ID of the words belongs to each word pairs in the real-life test datasets (RG65, WS353, MEN3000) are found and placed into the vocabulary list. The binary combinations of all ID values in the generated vocabulary list are used to determine whether these two words are in relation to WordNet. All available pairs are added to the training set with the type of relationship between them (the codes are specified in Table 3.12). In this process, ambiguity may occur in a few places, but these problems are solved by the following approaches. Training dataset generation overview can be found in figure 3.9.

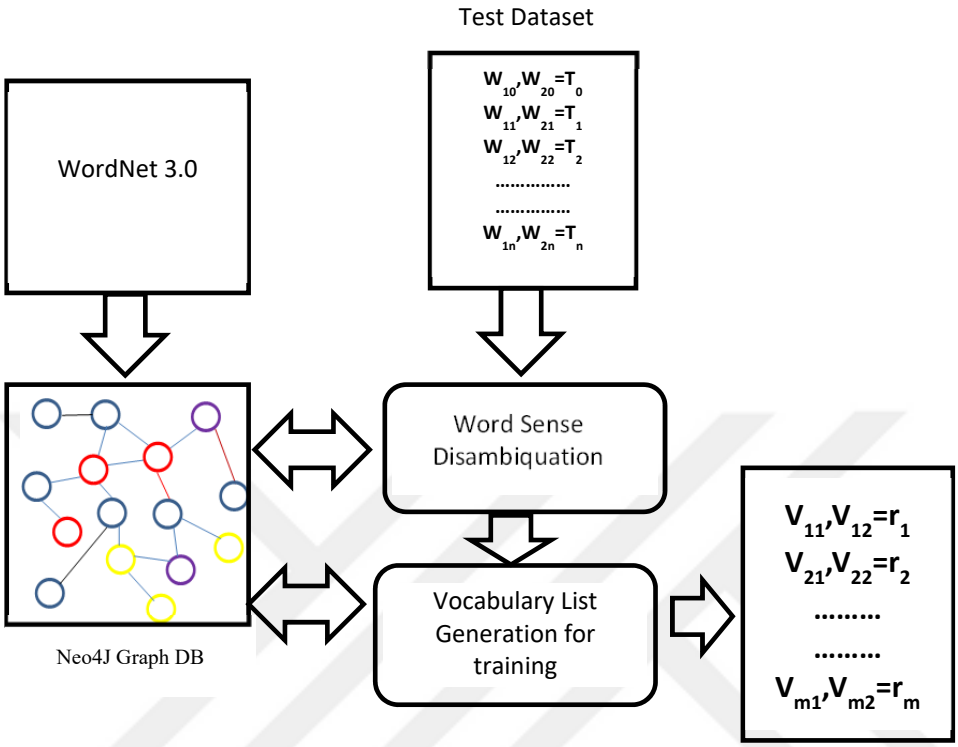


Figure 0.9. Training dataset generation overview

WordNet 3.0 is downloaded as raw text, it is converted to Neo4J using a java application developed by us. Any dataset for the benchmark purposes can be used, word pairs given in the test dataset are taken as word1 and word2, and this words helps us to obtain vocabulary list exist between them in the WordNet structure. If any of the words in a pair has more than one node in the graph then WSD procedure is applied.

After WordNet 3.0 is created as a network in a graph-based database and the relationship types are coded as shown in Table 3.12, all possible binary relationships in WordNet for the words in the vocabulary list. However, the word pairs in the test set will have normalized numerical outputs in the range of [0 1], while the word pairs semantic relations are coded as in Table 3.12.

Table 0.12. The relationship types and corresponding codes for SemSpace algorithm in WordNet 3.0

Relation Type	**Rel. Code**	**Relation Type**	**Rel. Code**
Synonym	&	Derivationally_related_form	m
PertainedBy	!	Domain_of_synset_TOPIC	n
Antonym	a	Member_of_this_domain_TOPIC	o
Hypernym	b	Domain_of_synset_REGION	p
Hyponym	c	Member_of_this_domain_REGION	q
Instance_Hypernym	d	Domain_of_synset_USAGE	r
Instance_Hyponym	e	Member_of_this_domain_USAGE	s
Member_Holonym	f	Entailment	t
Member_Meronym	g	Cause	u
Substance_Holonym	h	Also_see	v
Substance_Meronym	i	Similar_to	w
Part_Holonym	j	Verb_Group	x
Part_meronym	k	Participle_of_verb	y
Attribute	l	Pertainym	z

Although the two relationship types (Synonym, PertainedBy) listed in Table 3.12 are not defined on WordNet, the approach in our study assumes that there should be a relationship in the opposite direction of each type of relationship.

In order to solve ambiguity problem among the word pairs in the datasets, we have offered an approach that is called WSD approach. The first case of the WSD approach, it is supposed that two words can be in a synonym relation in the WordNet. Since the Synonym relationship is the most prioritized relationship type according to Sblini and Kosseim (2013) study, these two words are assumed to express the same concept regardless of other meanings. In the vocabulary list, they are represented as different words with the same ID, the synonym relationship between these words is added to the training set. On the other hand, some words in the test sets can be found on more than one node in the WordNet network. In order to find the correct nodes, all the nodes of the ambiguity words are found on the WordNet and they compare by means of the assumption that “semantically closer concepts are taken in case of ambiguity” (Pilehvar et al., 2013). By the way, semantically closeness also has a positive correlation with the shortest paths

between the nodes. For example, in order to find the correct words represented in the pairs “book” and “paper” given in WS353, firstly all the nodes with “paper” and “book” are found on the WordNet. 13 nodes for “book” and 9 nodes for “paper” are found. Then Total 13X9=117 word pars are compared to find the shortest path between nodes and the one with the shortest path is assumed to be the correct nodes representing the words “book” and “paper”.

When implementing this approach on WordNet, if more than one shortest path is specified with the same length, then the relationship type priority is considered. Sblini and Kosseim (2013) divided semantic relations into six different severity levels. If there is more than one shortest path of the same length for the two words being compared, a significance level is calculated for each path using the relationship priorities shown in Table 3.7. The absolute severity of each path is calculated by multiplying the priority values of relation types. When the selection is made, the principle with the lowest absolute importance wins is applied. If more than one way wins means has the same severity, one of them are randomly selected. Although one of our goals is to determine the weights of the WordNet relations, we can say that we use the priority of a relationship that we do not know whether it will coincide with the results of our study. However, considering that this method will be used to solve a rare ambiguity, its effect on the intended results will be as low as possible. For example, if there is a subgraph as given in Figure 4.X and if we are looking get correct nodes for the W1 and W2. According to our approach, red circled nodes are taken as shown in Figure 3.10.

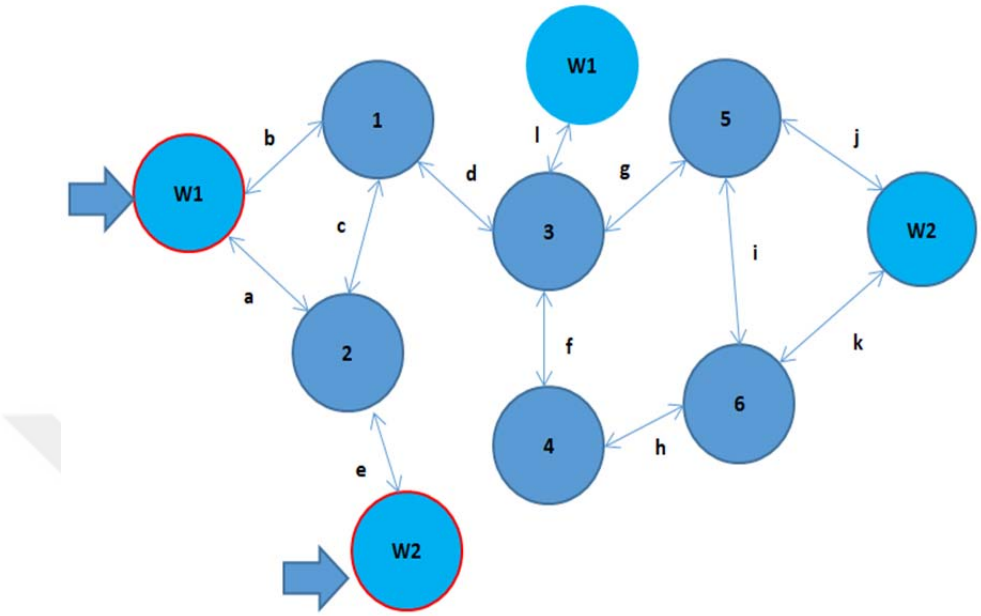


Figure 0.10. A Sample subgraph to show how to solve WSD

As seen in the Figure 3.10, WSD is solved using the shortest paths between ambiguous nodes and red circled nodes are chosen as a solution.

After solving the WSD, next step is to get vocabulary list that will be used as an input to the iterative learning algorithm of the proposed approach. Algorithm 6 is used to generate vocabulary list for training.

Algoritm 6 *Vocabulary List Generation for Training*

Input: *Real life dataset DS and WordNet graph G*
Output: *vocabulary list as object list as VL*

```
1:    Initialize Object list PL(node1,node2,relation) ←Ø
2:    For(1) each word pair(w1, w2) in DS
3:        P ←findShortestPathwithWSD(w1,w2,G)
4:        For(2) each neighbour(n1,n2) in P
5:            Rel ←getRelation(n1,n2)
6:            VL.add(n1,n2,rel)
7:        End for(2)
8:    End For (1)
9:    Return VL
```

In Algorithm 6, first real life dataset to be test is taken, these can be any dataset consist of the word pairs list, but generally real life datasets is preferred in order to make benchmark evaluations. And WordNet as graph db is used to get corresponding node list as vocabulary list from graph db. By iterating all the dataset in the algorithm, shortest path with WSD is applied and shortestpath is retrieved. After taking the shortest path then it is iterated and every adjacent nodes and their relations are added to the vocabulary list. This procedure iterated for all word pairs in the test data and vocabulary list s generated.

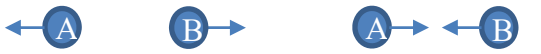
3.3.2. The Algorithm of SemSpace

In the proposed iterative learning algorithm, firstly training dataset is prepared which is explained in the previous subsection. Then initial learning paraters are set, these learning parameters are; Protoype dimension (exp:2…100), number of iteration of learning process (exp, 1.000.000), update amount of the relation weights (exp, 0.001), since they will be starting with random values, it will be increased or decreased according to defined amount here. Update amount of the prototype positions (exp, 0.01). Stop threashold for training, when the difference between Euclidian distance based similarity of the protoypes and corresponding relation weights are smaller than this amount, training will be stopped, this means training is successful. Number of iterations will be used in the training, this will be also used to stop the training (exp, 100). Dataset to be used in the to test the system

energy (RG65, WS353, MEN3000 etc.). Randomization of the prototype positions is also made here in this step. And WordNet relation weights are randomly generated between [0..1] in this step. Next step is the update of the prototype positions and randomly generated WordNet weights. Training of the prototypes are done by updating the protype positions until it converges with the randomly assigned relation values. The update of the prototype positions is done by the Eq. (3.10) in each training iterations.

$$\Delta P = \eta\,(A - B) \tag{3.10}$$

In Equation 3.9, the term η refers to the coefficient of learning, *(A-B)*, the directional distance between prototypes A and B, and *ΔP* represents the updating amount the distance of both prototypes, learing coefficient and update amount is ste in the first step of the algorithm. In the training process, protoype positions are updated The direction of the updated amount applied to each prototype is a different way. If the similarity value calculated between A and B prototypes *(Ysim (A, B))* is higher than the desired similarity *(Dsim (A, B))*, both prototypes are moved out from each other, otherwise, they are moved in which means bring them closer. This is shown in Figure 3.11.



$Y_{sim(A,B)} > D_{sim(A,B)}$ $Y_{sim(A,B)} < D_{sim(A,B)}$

a b

Figure 0.11. As a supervised updating of prototype positions in hyperspace

As shown in Figure 3.11, the positions of the prototypes are located by adjusting the distance depending on the desired similarity. Therefore, a similarity-based distance function should be used in similarity calculation. In the study, similarity function based on Euclidean distance which is traditionally used in literature has been preferred. This function is given by Equation (3.10).

$$Y_{sim(A,B)} = \frac{1}{1 + \|A - B\|} \tag{3.11}$$

In Eq. (3.11), $\|A - B\|$ the expression represents the distance between prototypes. All distance values in the study were calculated with Euclidean distance. On the other hand, it is considered useful to keep the iterative training time of the prototype positions short in order to use the random search contribution of the algorithm effectively. Therefore, the convergence threshold value used in almost all machine learning methods were kept high in this study ([0.1 0.01]) and the maximum number of iterations were kept low ([25 50]). Training process is shown as an algotihm in Algorithm 7.

Algoritm 7 *Training Algorithm*

Input: *Vocabulary List to Train* ***TS****, training parameters* ***R1, R2,*** *Training iteration count* ***ItrTr***

Output: *Trained positions of the vocabularies in hyperspace as list* ***NodePosList*** *(node,pos)*

```
NodePosList←Ø
For(1) each node pair(N1, N2) in TS
    V1,V2←get prototype positions of n1,n2
    R←Get relations between n1,n2
    W←Get randomly assigned weight for R
    For(2) (I=1 to TrItr)
      D← Calculate Euclidean distance between V1,V2
      S← Calculate similarity(1/(1+D))
      If(1) abs(W-S)<R1
        break
      Else
        If(2) W>S
          V1←V1-R2
          V2←V2-R2
        Else
          V1←V1+R2
          V2←V2+R2
        End if(2)
      End if(1)
    End For (2)
    NodePosList.add(N1,V1), NodePosList.add(N2,V2)
End For (1)
Return NodePosList
```

The energy values calculated separately for each set were determined by taking the difference of Spearman Correlation (SC) and Mean Absolute Error (MAE) values. Prototype locations and WordNet relationship weights are stored those belong to the highest energy value. Higher values of E_{mean} measure obtained from form the learning process, the developed algorithm is assumed to be successful. The energies of the train-set (E_{tr}), the test-set (E_{ts}) and their average (E_{mean}) are followed during the training process, and vector positions and the WordNet's relation-weights, which support the maximum E_{mean} value, are saved. The mean energy is computed on train and test sets as shown in Eq.(3.12).

$$E_{mean} = \frac{SC_{tr} + SC_{ts} - MAE_{tr} - MAE_{ts}}{2} \tag{3.12}$$

The problem of weight determination in WordNet requires resolution at the same time as the optimization of the prototype position. At this stage, a search-based approach was used and "the success of the system on test data" was used as the objective function. This step of the algorithm can be described as "Keep the current WordNet relationship weights and prototype locations for the situation that leads the system to the highest success". The last step of the algorithm can be explained as "Add random values as α to the best prototype positions on hand and add β random values to the WordNet weights and create new start positions and weights, and iterate it until desired iteration value then go to training step".

4. RESULTS AND DISCUSSIONS

In this section, experimental studies performed on SemSpace algorithm is given in details and the result of the experiments are discussed. In SemSpace method, RG65, WS353, and MEN3000 similarity sets were used as test data sets. As a result of the processing of the WS353 dataset, it was found that the word "Maradona" is not contained in WordNet, and therefore word pair contains this word was removed from the dataset. In the shortest path analysis on WordNet for RG65, WS353 and MEN3000 datasets, 76, 536 and 1837 different synset were detected and the ID values of the synset were taken into vocabulary lists. When the first-degree neighborhoods of the syntaxes among these vocabulary lists were examined, 28, 85 and 1,112 neighborhoods were determined for RG65, WS353 and MEN3000, respectively, and these synset pairs were stored as training sets. According to this study, three experiments were conducted which evaluated RG65, WS353, and MEN3000 datasets as a test set. All the vocabulary lists related to each experiment were queried in WordNet. As a result of the query, gathered words and their relations each other has been used as a training set.

4.1. Analysis of Dimension Effects

In the experiments with RG65 and WS353 data sets are sufficient to work only in two-dimensional space, while a large number of vocabulary and train sets such as the MEN3000 have a high level of training and test success, with a higher dimensional size of prototypes. According to the prototype dimensions, MAE and SC values obtained from the train and test sets are given in Table 4.1.

Table 0.1. MAE and SC values obtained on trainset and test set

Dataset	N	SC_{tr}	MAE_{tr}	E_{tr}	SC_{ts}	MAE_{ts}	E_{ts}	E_{mean}
RG65	2	93.29	8.80	84.49	94.43	7.80	86.63	85.56
RG65	49	93.29	0.01	93.28	92.55	11.17	81.38	87.33
RG65*	49	93.29	0.01	93.28	**94.46**	5.61	88.85	**91.07**
WS353	2	93.31	2.30	91.01	92.47	3.00	89.47	90.24
WS353	49	93.50	0.36	93.14	95.31	3.30	92.01	92.58
WS353*	49	94.17	0.01	94.16	**94.13**	2.06	92.07	**93.12**
MEN3000	2	94.84	0.50	94.34	74.17	13.10	61.07	77.71
MEN3000	49	95.29	0.00	95.29	78.77	10.75	68.02	81.66
MEN3000	55	94.53	0.40	94.13	80.13	11.50	68.63	81.38
MEN3000	99	94.70	0.40	94.30	79.08	11.20	67.88	81.09
MEN3000*	55	94.62	0.45	94.17	**88.77**	5.24	83.53	**88.85**

*: Relationship weights greater than 0.25 are used in the test set.

According to Table 4.1, N shows the prototype dimension, SC_{tr} is the training success according to Spearman Correlation, MAE_{tr} is the mean absolute error in training, E_{tr} shows the energy of the training, SC_{ts} is the test success to Spearman Correlation, MAE_{ts} is the mean absolute error in test, E_{ts} is the test energy of the system and E_{mean} is the average system energy of the proposed dimension in the experiment.

It is observed that the higher size selection of prototypes dimension has contributed to an increase in both *SC* success and decrease in *MAE* deviation. However, in order to define this contribution in clear terms, it can be said that more data sets should be experimented. In the study, different values of N (dimension) parameter were tried in algorithm which indicated the size of word vector space. As the representation of words in high-dimensional space with one-hot-vector is known to cause both memory problem and computational complexity due to high sparsity, it is generally aimed to choose the smallest space that provides similar performance. However, in the experiments performed with SemSpace algorithm, many N values have been determined which provide almost the same performance. As a second control point, therefore, visual control of the discoveries was provided. At the beginning of the training, the higher dimensional size was chosen, this caused large sparsity and prototypes were placed into the longer distances that

shows the less similarity. Using the word prototypes that has defined relations in the training set bring them closer that caused successful relation discovery by the way. We accept semantic relations which are greater than 0.25 as strong, and their experiment results (the lines marked by *) show that the using of both large vector dimensions and strong semantic relations improve the energy of SemSpace algorithm.

4.2. Benchmark Test Results and Comparison

SemSpace method's benchmark test results with popular semantic word relatedness datasets are also compared with widely accepted successful word relatedness methods used in the literature. As shown in the Table 4.2, SemSpace method achieves best results according to spearman rank correlation values. SemSpace method is the only knowledge-based method that uses purely Princeton English WordNet. Other successful approaches are using not only different type of the Knowledge-bases like Wiktionary or using the large corpora consist of billions of the words.

Table 0.2. Comparison of the benchmark test results with SemSpace method and other popular methods

		Spearman Rank Correlation		
Method	**Approach**	**RG65 DataSet**	**WS353 Dataset**	**MEN3000 Dataset**
SemSpace Method	Knowledge-based (WordNet)	0.94	0.94	0.88
ADW	Knowledge-based (Wiktionary)	0.92	0.75	
ConceptNet Numberbatch	Hybrid		0.828	0.87
Y&Q	Corpus Based	0.89	0.81	
GloVe	Corpus Based(NNLM)	0.83	0.76	
Word2Vec	Corpus Based(NNLM)	0.70	0.63	

4.3. Analysis of the WordNet's Relation-Weights

In order to reduce the number of unknowns in our study, which appears to be a problem of many unknowns, the types of relationships given in Table 3.12 are grouped based on the principle of "mutual relations should have the same weight" (bc, de, fg, hi, jk, no, pq, rs, z-!). With this assumption, the number of relations is reduced to 19. 3 different relationship types (&, b, m) in the train-set prepared from WordNet for RG65, 7 relationship types for WS353 (&,a,b,g,k,m,n) and 16 relationship types for MEN3000 (&,a,b,d,g,h,k,l,m,n,t,u,v,w,x,z) were determined. It was observed that the WordNet relation weights obtained in the experiments were consistent with each other (when only two digits were used after the comma). These weights are presented in Table 4.3.

Table 0.3. WordNet relationship weights that provide the optimum values obtained in the study

Dataset	&	a	b	d	g	h	k	l	m	n	t	u	v	w	x	z
RG65* (N=49)	.84	-	.74	-	-	-	-	-	.50	-	-	-	-	-	-	-
WS353* (N=49)	.81	.71	.73	-	.50	-	.50	-	.50	.50	-	-	-	-	-	-
MEN3000* (N=55)	.81	.72	.71	.50	.50	.50	.51	.51	.50	.50	.50	.50	.50	.50	.50	.50

*: Relationship weights greater than 0.25 are used in the test set.

According to Table 4.3, the highest correlation and the lowest error in the weights in terms of synonyms, antonym, and hypernym- hyponym relations were determined with high importance while others remained at the level of 0.5. From another point of view, if people find the words close enough to be synonymous with each other they give similarity value as [0.8 1], if they find it with relations antonym, hypernym-hyponym relations they give similarity value as with [0.7 0.8]. And other relations except those above is measured with [0.5 0.7]. A general expression cannot be given because there is no consistency in the similarity values outside ranges above. In the study, it was found that the synonym relation was the

highest and the antonym-hyponym-hypernym relations had the second most valuable weight values. All other types of relationships were placed in the third group with almost the same weight. In this way, it can be said that WordNet relations are consistent and also divided into three important semantic groups.

4.4. Discovery of New Relations

In the last analysis of the study, all possible pairs of words are listed for all words in the vocabulary list using the best values for each test set. Similarity-based on distance is calculated for each word pairs are calculated and top 10 relations with higher similarity was filtered. Accordingly, word pairs discovered in each test set in Table 4.4 are presented with similarity value.

Table 0.4. Similarity pairs discovered in the test sets

RG65 (N=49)	**WS353 (N=49)**	**MEN3000 (N=55)**
mound, hill, 0.84	psychiatry, Freud, 0.84	color, colour, 0.89
smile, grin, 0.74	mind, psychiatry, 0.81	painting, picture, 0.85
crane, tool, 0.53	category, family, 0.81	tulip, rose, 0.83
cock, crane, 0.53	cemetery, graveyard, 0.81	frog, lizard, 0.83
	forest, woodland, 0.81	apple, strawberry, 0.82
	mind, Freud, 0.79	owl, swan, 0.82
	Freud, health, 0.78	owl, parrot, 0.81
	mind, depression, 0.78	ski, jump, 0.81
	cognition, mind, 0.78	post, office, 0.81
	mind, science, 0.78	…
	…	

Several word pairs and corresponding relations are shown in Table 4.4, and when all the relationships and word pairs obtained in the analysis are examined visually, a significant relevance has been observed in all discovered relationships. While some of these are synonymous words like “color, colour”, others are indirectly related to “mind, science”. By increasing the Sparsity value with increasing the dimension, it was observed that the similarity values of the word pairs (like mind-science) which were not directly related decreased. Specifically, for the N = 2 value, it was observed that the discoveries for all three test sets did

not have an acceptable consistency and that many unrelated words pair had a high similarity. Using the values N = 49 and N = 55, the expansion of the space was ensured, the sparsity value increased, so that the unrelated words that started with the random distribution did not have to be adjacent to each other in a multidimensional space.

For the vocabulary list of the RG65 cluster, the similarity of only 4 pairs in the N = 49 space exceeded 0.5. Two of them were larger than 0.7 and only one was greater than 0.8. The numbers of valuable word pairs for each of the three test sets are presented in detail in Table 4.5.

Table 0.5. Similarity statistics of new word pairs discovered in test sets

Relatedness Score	**RG65 (N=49)**	**WS353 (N=49)**	**MEN3000 (N=55)**
≥ 0.8	1	5	54
≥ 0.7	2	101	677
≥ 0.5	4	400	19,460

In Table 4.5, if we look at the ratio of the most valuable (≥0.8) number of relations to the number of relations in the medium value (≥0.5), the growth of the vocabulary list led to a decrease in this ratio.

5. CONCLUSIONS

In this thesis, which was carried out in the form of graph-based approaches using WordNet in the measurement of word-level semantic relatedness in natural language processing; firstly, word level relatedness, numerical measurement of this relatedness, the studies of the mathematical models that to realize it, and the studies on the computation of relatedness, have been carried out and experimental studies have been conducted which resulted in a new model development. WordNet, developed by linguists and computer experts, thanks to its planned and systematic structure, it can be fully mapped to the graph database and it has the potential to be successful in graph-based studies, these are the main reasons of the WordNet have been preferred in this study. Words are expressed in low-dimensional dense space with word representations and the successes of the corpus-based approaches are considered, by the way, the importance of word representations and the use of vectors derived from word representations and the weighted distance obtained from graphs tend to be combined to form a new model. With SemSpace method, it is observed that the distance between the words on the graph network via WordNet and the similarity obtained by the human evaluations of the same words can create representations of a certain size in the vector space and these generated vectors give successful results when tested with known benchmarking data.

Until reaching to SemSpace, many different approaches have been tried. First of all, experimental studies of some of the popular methods have been conducted and those experiments motivated us to go deep into the discovery of the new models. Initially, WordNet was transformed into a graphical network, and various similarity scenarios were applied using graph metrics on this graphical network. Using the UKB method (Agirre et al., 2009), it was observed that how PageRank vectors were formed and how these PageRank vectors were used effectively by ADW (Pilehvar et al., 2013). The study of the semantic relatedness in Turkish WordNet (Tulu and Orhan, 2017), especially with the UKB method, is

the first WordNet-based study in Turkish. On the other hand, examining the shortest path and the length of all paths generated in order to measure the semantic similarity of the two concepts on the graph network, that proposes importance on semantic relations (Tulu et al., 2018) is another important output of these studies. It is one of the important elements of the fact that the MEN3000 data is created by the composed by fifty different native speakers of English, including all kinds of POS's and the use of the semantically of words in these studies. Although the results obtained in the study of the determination of the relationships consisting of 3000 rows and 26 different relationship types by weight using multiple linear regression method are not very successful, but maybe it is very important to give some ideas about that better results can be obtained by trying different approaches.

In SemSpace method, the new iterative learning algorithm is introduced; the words were represented as prototypes in multidimensional space. At the same time, the evaluation of some benchmark datasets by using WordNet relations was performed with high correlation and low error. Weights are determined in WordNet relations, the highly-correlated word pairs not specified in benchmark datasets are also discovered in this approach. It is thought that the study will contribute to the literature in several aspects. The first one is the iterative learning algorithm, which forms the basis of the study. With this algorithm, the words were taken out of the synset pattern in WordNet and gained their identities again. Thus, the relationship with other words could be expressed differently. In addition, in the evaluations made through the semantic relatedness measurement, a higher rate of success is achieved than the other approaches in the literature that convert the words into a vector. Another important finding of the study was the fact that WordNet verification was done by using WordNet relations prepared by scientists together with the numerical values given by people who participated in conceptual similarity questionnaires and also WordNet relations were tried to be determined.

It is also noteworthy that the results obtained with SemSpace method are above the existing methods in the literature. It is likely that this method will not

only inspire semantic similarity, but also a source of inspiration for researchers who want to find solutions to different natural language applications with graph-based approaches. Another indicator of the method being a successful and preferable method is that the computational costs are low according to other successful methods. For example, methods such as GloVe and word2vec, which convert the corpora with millions of words into low-dimensional dense word vectors, these are methods that have a certain cost of calculation even though they are fast, and this cost is very low for SemSpace method. In terms of using only MEN3000 and WordNet 3.0 as input, as compared to the other methods that use the very large gigabyte level corpora and examining the n-grams obtained from them, the advantage of SemSpace method is seen in every aspect

REFERENCES

Abdalraouf, H., Ausif M., 2017. “Deep learning for sentence classification”, IEEE Long Island Systems, Applications and Technology Conference (LISAT), NY, pp. 1-5 DOI:10.1109/LISAT.2017.8001979.

Agirre E., Soroa A., 2009. "Personalizing PageRank for Word Sense Disambiguation", Proceedings of the 12th Conference of the European Chapter of the ACL, pages 33–41, Athens, Greece

Agirre E., Alfonseca, E., Keith, H., Strakova, J., Pasca, M., Soroa, A., 2009. “A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches”, Proceedings of NAACL-HLT. PP:19-27, DOI: 10.3115/1620754.1620758.

Altınel, B., Ganiz, M. C., 2018. “Semantic text classification: A survey of past and recent advances”, Information Processing, Management, 54(6), 1129-1153.

Beiswenger, K., Calcutt A., Mizisin A., 2008. “Dissociation of thermal hypoalgesia and epidermal denervation in streptozotocin-diabetic mice”, Neuroscience letters, 442. 267-72. 10.1016/j.neulet.2008.06.079.

Bengio Y., Ducharme R., Pascal V., Christian J., 2003. “A Neural Probabilistic Language Model, Journal of Machine Learning Research”, 3, (2003) 1137–1155

Bentivogli, L., Forner, P., Magnini, B., Pianta, E., 2004. “Revising the WordNet domains hierarchy”. MLR '04 Proceedings of the Workshop on Multilingual Linguistic Ressources Pages 101-108, Geneva, Switzerland, DOI: 101. 10.3115/1706238.1706254.

Bilgin, O., Çetinoglu O., Oflazer, K., 2004). “Building a WordNet for Turkish”, ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY Volume 7, 163-172.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. “Enriching Word Vectors with Subword Information”, Transactions of the Association for Computational Linguistics. DOI:5. 10.1162/tacl_a_00051.

Brin, S. and Page, L., 1998. “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Bruni, E., Tran, N.K.,, Baroni, M., 2014. “Multimodal Distributional Semantics”. Journal of Artificial Intelligence Research. DOI:10.1613/jair.4135.

Camacho-Collados J., Pilehvar M.T., Navigli R., 2015. “NASARI: a Novel Approach to a Semantically-Aware Representation of Items”. NAACL 2015, Denver, USA, pp. 567-577.

Collins, M.A, Loftus, E., 1975. “A Spreading Activation Theory of Semantic Processing”. Psychological Review, 82. 407-428 DOI:10.1037/0033-295X.82.6.407

Collobert, R., Weston, J., 2008. “A unified architecture for natural language processing: Deep neural networks with multitask learning”. Proceedings of the 25th International Conference on Machine Learning. 160-167 DOI: 10.1145/1390156.1390177.

Dehkharghani, R., Saygin, Y., Yanikoglu, B., Oflazer, K., 2016. “SentiTurkNet: a Turkish polarity lexicon for sentiment analysis”, Language Resources and Evaluation, Pages: 1–19, Spinger, DOI:10.1007/s10579-015-9307-6.

Dragoni M., Petrucci G., 2017. “A neural word embeddings approach for multi-domain sentiment analysis”, IEEE Transactions on Affective Computing 8 (4), 457-470

Dumais T., Furnas S., Landauer G., Thomas, Scott T.D., 1988. “Using latent semantic analysis to improve information retrieval”, CHI '88 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Pages 281-285, Washington, D.C., USA

Elavarasi S., Saravanan K., Renuka C., 2014. “A systematic review on medicinal plants used to treat diabetes mellitus”. International Journal of Pharmaceutical, Chemical And Biological Sciences. 3. 983-992.

Faruqui M., Dodge J., Jauhar S., Dyer C., Hovy E., Noah S., 2015. “Retrofitting Word Vectors to Semantic Lexicons”, NAACL, 2015

Fellbaum, C., 1998. “A Semantic Network of English: The Mother of All WordNets.” Computers and the Humanities, 32, 209-220 DOI:10.1023/A:1001181927857.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E., 2001. “Placing search in context: The concept revisited.” ACM Transactions on Information Systems - TOIS. 20. 406-414 DOI:10.1145/503104.503110.

Francis N., Kucera W., Henry, W., Mackie, A., 1982. “Frequency Analysis of English Usage”, Lexicon and Grammar. 18. 64-70.

Gabrilovich E., Markovitch S., 2009,"Wikipedia-based Semantic Interpretation for Natural Language Processing", Journal of Artificial Intelligence Research, vol.34, pp. 443-498

Gomaa, W., Fahmy, A., 2014. “Arabic Short Answer Scoring with Effective Feedback for Students”, International Journal of Computer Applications, 86, DOI:10.5120/14961-3177.

Guessoum, D., Miraoui, M., 2016. “A modification of Wu and Palmer Semantic Similarity Measure”, The Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies

Gujarati, D.N., Porter, D.C. (2009). "How to Measure Elasticity: The Log-Linear Model". Basic Econometrics. New York: McGraw-Hill/Irwin. pp. 159–162

Harris, S., 1954. “Distributional Structure”, Word, 10, PP:146-162 DOI:10.1007/978-94-009-8467-7_1.

Hassan, S., Mihalcea, R., 2011. “Semantic Relatedness Using Salient Semantic Analysis”, Twenty-Fifth AAAI Conference on Artificial Intelligence

Haveliwala H., Taher, G., Aristides, K., Dan, I.P., 2002. “Evaluating Strategies for Similarity Search on the Web.” Proceedings of the 11th International Conference on World Wide Web, WWW DOI:02. 10.1145/511446.511502.

Hirst H., St-Onge D., 1998. “Lexical chains as representations of context for the detection and correction of malapropisms”. In Fellbaum 1998, 305–332.

Hughes T., Ramage D., 2007. “Lexical semantic relatedness with random graph walks”, Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)

Islam A., Inkpen, D., 2006. “Second order co-occurrence PMI for determining the semantic similarity of words”, Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006).

Jiang J.J., Conrath, D.W., 1997. “Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy”, Proceedings of the International Conference on Research in Computational Linguistics, 10.

Kohonen, T. (1995), "Learning vector quantization", in M.A. Arbib (ed.), The Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press, pp. 537–540

Kozima, H., Furugori, T., 1996. “Similarity between Words Computed by Spreading Activation on an English Dictionary”, In Proceedings of the European Association for Computational Linguistics, DOI:10.3115/976744.976772.

Lacobacci I., Pilehvar M.T., Navigli R., 2016. “Embeddings for Word Sense Disambiguation: An Evaluation Study”, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 897–907, Berlin, Germany, August 7-12, 2016

Leacock, C., Chodorow, M., 1998. Combining Local Context and WordNet Similarity for Word Sense Identification.

Lesk, M., 1986. “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone”, Proceedings of SIGDOC-86: 5th International Conference on Systems Documentation, 24–26, DOI:10.1145/318723.318728.

Levy O., Goldberg Y., Dagan I., 2015. “Improving Distributional Similarity with Lessons Learned from Word Embeddings”, Transactions of the Association for Computational Linguistics, vol. 3, pp. 211–225.

Lin D., 1998. “An Information-Theoretic Definition of Similarity”. In 15th International Conference of Machine Learning, Madison,WI: 1998:296-304.

Lund, K., Burgess, C., 1996. “Producing high-dimensional semantic space from lexical co-occurrence”, Behavior Research Methods Instruments, Computers. 28. 203-208. DOI:10.3758/BF03204766.

Mikolov, T., Sutskever, I., Chen, K. Corrado, G., Dean, J., 2013. “Distributed Representations of Words and Phrases and their Compositionality”, Advances in Neural Information Processing Systems, 26,

Mikolov T., 2007. “Language modeling of Czech using neural networks”, Proc. 13th Conference, STUDENT EEICT 2007

Mikolov T., Kopecky K., Burget L., Glembek O., 2009. “Neural network based language models for highly inflective languages”, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing

Miller A., George, G. Charles, Walter., 1991. Contextual Correlates of Semantic Similarity, Language and Cognitive Processes, 6, 1-28. DOI:10.1080/01690969108406936.

Morris, J., Hirst, G., 2004. “Non-classical lexical semantic relations”, Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics, Pages 46-51, DOI: 10.3115/1596431.1596438

Pennington, J., Socher, R., Manning, C., 2014. “Glove: Global Vectors for Word Representation”, EMNLP, 14, 1532-1543, DOI:10.3115/v1/D14-1162.

Pilehvar M.T., Navigli R., 2015. “From Senses to Texts: An All-in-one Graph-based Approach for Measuring Semantic Similarity”, Artificial Intelligence, 228, pp. 95-128, Elsevier

Pilehvar M.T., Jurgens D., Navigli R., 2013. “Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity.”, ACL 2013, Sofia, Bulgaria

Prakash M.N., Lucila O.M., Chapman, W., 2011. “Natural language processing: An introduction”, Journal of the American Medical Informatics Association : JAMIA. 18. 544-51. 10.1136/amiajnl-2011-000464.

Quillian, M.R., 1967. “Word concepts: A theory and simulation of some basic semantic capabilities, Behavioral Science 12(5), 410-430, DOI:10.1002/bs.3830120511.

Rada, R., Mili, H., Bicknell, E., Blettner, M., 1989. “Development and application of a metric on semantic nets. Systems”, Man and Cybernetics, IEEE Transactions on. 19, 17 - 30, DOI:10.1109/21.24528.

Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S., 2011. “A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis”, Proceedings of the 20th International Conference on World Wide Web, WWW 2011, 337-346, DOI:10.1145/1963405.1963455.

Resnik, D., 1995. “Developmental constraints and patterns: Some pertinent distinctions”, Journal of Theoretical Biology, J THEOR BIOL, 173, 231-240. DOI:10.1006/jtbi.1995.0059.

Resnik, P., 1999. “Semantic Similarity in a Taxonomy: An Information Based Measure and Its Application to Problems of Ambiguity in Natural Language”, Journal of Artificial Intelligence Research, 11, 95-130, DOI:10.1613/jair.514.

Ross, S., 1976. “The Arbitrage Theory of Capital Asset Pricing”, Journal of Economic Theory, 13. 341-360, DOI:10.1016/0022-0531(76)90046-6.

Rubenstein, H., Goodenough, J., 1965. “Contextual correlates of synonymy” Commun. ACM, 8, 627-633., DOI:10.1145/365628.365657.

Saedi C., Branco A., Rodrigues J.A., Silva J.R., 2018, "WordNet Embeddings", Proceedings of the 3rd Workshop on Representation Learning for NLP, pages 122–131, Melbourne, Australia

Siblini, R., Kosseim, L., 2013. “Using a weighted semantic network for lexical semantic relatedness”, International Conference Recent Advances in Natural Language Processing, RANLP, 610-618.

Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L., 2002. “Open Mind Common Sense: Knowledge acquisition from the general public”, Lecture Notes in Computer Science, 2519, 1223-1237

Smith, Barry, Fellbaum, Christiane., 2004. Medical WordNet: a new methodology for the construction and validation of information resources for consumer health. 10.3115/1220355.1220409.

Speer, R., Chin, J., Havasi, C.., 2017. “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge”, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence

Sultan M.D., Salazar C., Sumner T., 2016. “Fast and Easy Short Answer Grading with High Accuracy”, Proceedings of NAACL-HLT 2016, pages 1070–1075

Sussna, M., 1993. “Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network.”. Proc 2nd Int Conf on Information and Knowledge Management, 67-74, DOI:10.1145/170088.170106.

Tsatsaronis, G., Varlamis, I., Nørvåg, K., 2010. “An Experimental Study on Unsupervised Graph-based Word Sense Disambiguation”, Computational Linguistics and Intelligent Text Processing, 184-198. DOI:10.1007/978-3-642-12116-6_16

Tulu, C., Orhan, U., 2017. "PageRank based semantic similarity measure on a graph based Turkish WordNet", International Conference on Computer Science and Engineering (UBMK), pp. 468-473, Antalya, DOI: 10.1109/UBMK.2017.8093438

Tulu, C., Orhan, U., Turan, E., 2018. "Semantic Relation's Weight Determination on a Graph Based WordNet", GUFBED CMES (2018) 67-78, DOI:10.17714/gumusfenbil.432582

Vossen V., 1998. EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer, Dordrecht, The Netherlands.

Wojtinnek, P., Pulman, S., 2011. "Semantic relatedness from automatically generated semantic networks", IWCS '11 Proceedings of the Ninth International Conference on Computational Semantics, Pages 390-394

Wu, Z., Palmer, M., 1994. "Verbs Semantics and Lexical Selection", Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, 133-138, DOI:10.3115/981732.981751.

Yang, D., Powers, D., 2006. Word similarity on the taxonomy of WordNet.

Yih, W., Qazvinian, V., 2012. "Measuring Word Relatedness Using Heterogeneous Vector Space Models", Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2012), pp. 616–620

Yousefi-Azar, M., Hamey, L., 2016. "Text Summarization Using Unsupervised Deep Learning", Expert Systems with Applications, 68, DOI:10.1016/j.eswa.2016.10.017

Yucesoy V., 2007. "Turkce Dokumanlar icin Anlamsal Benzerlik Hesaplama Yontemi", MSC Thesis, Istanbul Technical University

Zesch, T., Gurevych, I., 2010. "Wisdom of crowds versus wisdom of linguists—Measuring the semantic relatedness of words", Natural Language Engineering, 16, 25-59 DOI:10.1017/S1351324909990167.

Zhang, Z., Gentile, A., Ciravegna, F., 2013. “Recent advances in methods of lexical semantic relatedness - A survey”, Natural Language Engineering, 19, DOI: 10.1017/S1351324912000125.

Rada, R., Mili, H., Bicknell, E., Blettner, M., 1989. “Development and application of a metric on semantic nets. Systems”, Man and Cybernetics, IEEE Transactions on. 19, 17 - 30, DOI:10.1109/21.24528.

Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S., 2011. “A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis”, Proceedings of the 20th International Conference on World Wide Web, WWW 2011, 337-346, DOI:10.1145/1963405.1963455.

Resnik, D., 1995. “Developmental constraints and patterns: Some pertinent distinctions”, Journal of Theoretical Biology, J THEOR BIOL, 173, 231-240. DOI:10.1006/jtbi.1995.0059.

Resnik, P., 1999. “Semantic Similarity in a Taxonomy: An Information Based Measure and Its Application to Problems of Ambiguity in Natural Language”, Journal of Artificial Intelligence Research, 11, 95-130, DOI:10.1613/jair.514.

Ross, S., 1976. “The Arbitrage Theory of Capital Asset Pricing”, Journal of Economic Theory, 13. 341-360, DOI:10.1016/0022-0531(76)90046-6.

Rubenstein, H., Goodenough, J., 1965. “Contextual correlates of synonymy” Commun. ACM, 8, 627-633., DOI:10.1145/365628.365657.

Siblini, R., Kosseim, L., 2013. “Using a weighted semantic network for lexical semantic relatedness”, International Conference Recent Advances in Natural Language Processing, RANLP, 610-618.

Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L., 2002. “Open Mind Common Sense: Knowledge acquisition from the general public”, Lecture Notes in Computer Science, 2519, 1223-1237

Smith, Barry, Fellbaum, Christiane., 2004. Medical WordNet: a new methodology for the construction and validation of information resources for consumer health. 10.3115/1220355.1220409.

Speer, R., Chin, J., Havasi, C.., 2017. “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge”, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence

Sultan M.D., Salazar C., Sumner T., 2016. “Fast and Easy Short Answer Grading with High Accuracy”, Proceedings of NAACL-HLT 2016, pages 1070–1075

Sussna, M., 1993. “Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network.”. Proc 2nd Int Conf on Information and Knowledge Management, 67-74, DOI:10.1145/170088.170106.

Toutanova, K., Klein, D., Manning, C., Singer, Y., 2004. “Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network”, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology—NAACL 03, DOI:10.3115/1073445.1073478.

Tsatsaronis, G., Varlamis, I., Nørvåg, K., 2010. “An Experimental Study on Unsupervised Graph-based Word Sense Disambiguation”, Computational Linguistics and Intelligent Text Processing, 184-198. DOI:10.1007/978-3-642-12116-6_16

Tulu, C., Orhan, U., 2017. "PageRank based semantic similarity measure on a graph based Turkish WordNet", International Conference on Computer Science and Engineering (UBMK), pp. 468-473, Antalya, DOI: 10.1109/UBMK.2017.8093438

Tulu, C., Orhan, U., Turan, E., 2018. "Semantic Relation's Weight Determination on a Graph Based WordNet", GUFBED CMES (2018) 67-78, DOI:10.17714/gumusfenbil.432582

Wojtinnek, P., Pulman, S., 2011. “Semantic relatedness from automatically generated semantic networks”, IWCS '11 Proceedings of the Ninth International Conference on Computational Semantics, Pages 390-394

Wu, Z., Palmer, M., 1994). “Verbs Semantics and Lexical Selection”, Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, 133-138, DOI:10.3115/981732.981751.

Yih, W., Qazvinian, V., 2012. “Measuring Word Relatedness Using Heterogeneous Vector Space Models”, Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2012), pp. 616–620

Yousefi-Azar, M., Hamey, L., 2016. “Text Summarization Using Unsupervised Deep Learning”, Expert Systems with Applications, 68, DOI:10.1016/j.eswa.2016.10.017

Yucesoy V., 2007. “Turkce Dokumanlar icin Anlamsal Benzerlik Hesaplama Yontemi”, MSC Thesis, Istanbul Technical University

Zesch, T., Gurevych, I., 2010. “Wisdom of crowds versus wisdom of linguists—Measuring the semantic relatedness of words”, Natural Language Engineering, 16, 25-59 DOI:10.1017/S1351324909990167.

Zhang, Z., Gentile, A., Ciravegna, F., 2013. “Recent advances in methods of lexical semantic relatedness - A survey”, Natural Language Engineering, 19, DOI: 10.1017/S1351324912000125.

BIOGRAPHY

Çağatay Neftali TÜLÜ was born in Adana, Turkey. He received his BSc. from Computer Engineering Department of Sakarya University in 2000, MSc from from Computer Engineering in 2003, from Sakarya University. Now, he has been working as Information Technologies Specialist in Adana Science and Technology University since 2012. His research interests include Natural Language Processing Applications, Semantic Similarity, Word Embedding's, Knowledge Based Systems, WordNet, Graph Based Systems, Distributional Approaches.