T.R.
EGE UNIVERSITY
Graduate School of Applied and Natural Science

PARAPHRASE IDENTIFICATION IN TURKISH USING DISTRIBUTED REPRESENTATIONS OF WORDS AND SENTENCES

MSc THESIS

Hakkı Engin YORGANCIOĞLU

International Computer

İzmir
2019

T.R.
EGE UNIVERSITY
Graduate School of Applied and Natural Science

PARAPHRASE IDENTIFICATION IN TURKISH USING DISTRIBUTED REPRESENTATIONS OF WORDS AND SENTENCES

Hakkı Engin YORGANCIOĞLU

Supervisor : Prof. Dr. Bahar KARAOĞLAN

International Computer
M.Sc. in Information Technologies

İzmir
2019

Hakkı Engin YORGANCIOĞLU tarafından YÜKSEK LİSANS TEZİ olarak sunulan **"Paraphrase Identification in Turkish Using Distributed Representations of Words and Sentences"** başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 27/11/19 tarihinde yapılan tez savunma sınavında aday oybirliği/~~oyçokluğu~~ ile başarılı bulunmuştur.

Jüri Üyeleri:		**İmza**
Jüri Başkanı	: Prof. Dr. Bahar KARAOĞLAN	....
Raportör Üye	: Doç. Dr. Tarık KIŞLA	....
Üye	: Doç. Dr. Senem Kumova Metin	....

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “Paraphrase Identification in Turkish Using Distributed Representations of Words and Sentences” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

27/12/2019

Hakkı Engin YORGANCIOĞLU

ÖZET

DAĞITIK KELİME VE CÜMLE TEMSİLLERİ KULLANILARAK TÜRKÇE EŞANLATIM TESPİTİ

YORGANCIOĞLU, Hakkı Engin

Yüksek Lisans Tezi, Uluslararası Bilgisayar Anabilim Dalı
Tez Danışmanı: Prof. Dr. Bahar KARAOĞLAN
Aralık 2019, 77 sayfa

Eşanlatım tespiti, doğal dil işleme alanında iki cümlenin aynı anlamda olup olmadığının belirlenmesi işlemidir. Makine çevirisi, otomatik metin özetleme ve intihal tespiti gibi çeşitli konulardaki uygulamalarıyla merkezi bir konudur. Son dönemde, kelime temsilleri kullanan yöntemler bu alanda başarılı sonuçlar almıştır. Kelime temsilleri, büyük derlemlerden yapay sinir ağları yöntemiyle öğrenilen yoğun vektör temsilleridir.

Bu tezde; kelime temsilleri yöntemi, üzerine kısıtlı çalışma bulunan Türkçe eşanlatım tespiti problemine uygulanmıştır. Yöntemin kural tabanlı yöntemlere kıyasla katkı sağlayıp sağlamadığının sınanması hedeflenmiştir. Ana fikir, kelime temsilleri kullanılarak cümle benzerliğinin hesaplanması ve bundan yola çıkarak eşanlatım tahmini yapılmasıdır. Türkçe metinlerden yeni bir derlem hazırlanmış ve yapay sinir ağları kullanılarak kelime temsilleri öğrenilmiştir. Kelime temsillerinden cümle benzerliğini hesaplayan çeşitli yöntemler, mevcut Türkçe eşanlatım derlemleri üzerinde sınanmıştır. Daha yüksek benzerlik puanına sahip cümle çiftleri beklendiği gibi eşanlatımlı olmaya daha yatkın bulunmuştur. Sonuçlar, Türkçe eşanlatım tespiti için güncel başarım oranlarına yakındır. Özet olarak, kelime temsillerinin Türkçe eşanlatım tespiti yöntemine uygulanması yaklaşımı olumlu sonuç vermiştir. İyileştirme ve araştırma önerileri belirtilmiştir.

Anahtar sözcükler: Eşanlatım tespiti, kelime temsilleri, kelime vektörleri, metin benzerliği, cümle benzerliği, doğal dil işleme

ABSTRACT

PARAPHRASE IDENTIFICATION IN TURKISH USING DISTRIBUTED REPRESENTATIONS OF WORDS AND SENTENCES

YORGANCIOĞLU, Hakkı Engin

MSc in International Computer Institute
Supervisor: Prof. Dr. Bahar KARAOĞLAN
December 2019, 77 pages

Paraphrase identification in natural language processing is the task of deciding whether two sentences have the same meaning. It is a central topic with applications in many tasks such as machine translation, automatic text summarisation and plagiarism detection. Recent studies in this field report successful results using word embeddings, which are vector representations of words learned from large corpora using neural networks.

In this thesis, we apply word embedding methods to Turkish paraphrase identification problem, which has limited research. The objective is to test if any improvement can be achieved over rule based methods. The main idea is to predict the paraphrase relation between a pair of sentences based on the similarity score calculated from word embeddings. A new corpus is created from Turkish texts and word embeddings are learned using neural networks. Several sentence similarity methods that make use of these word embeddings are evaluated against existing Turkish paraphrase corpora. Sentence pairs with a higher similarity score are found more likely to have paraphrase relation as expected. The results are close to the state-of-the-art in Turkish paraphrase identification. In summary, application of word embeddings to Turkish paraphrase identification is found a useful approach. Suggestions for improvements and further research directions are provided.

Keywords: Paraphrase identification, word embeddings, word vectors, text similarity, sentence similarity, natural language processing

PREFACE

(In Turkish)

Ailem, iş ve okul arasında mekik dokumaya razı olarak başlayıp bitirdiğim bu tez çalışmasının okuyanlara faydalı olması dileğiyle...

İZMİR
.../.../2019

Hakkı Engin YORGANCIOĞLU

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
NLP	Natural Language Processing
PI	Paraphrase Identification
MT	Machine Translation
LSA	Latent Semantic Analysis
SVD	Singular Value Decomposition
OOV	Out-of-Vocabulary

1. INTRODUCTION

1.1 Motivation

To paraphrase the dictionary definitions, a paraphrase is a rewording that gives the same meaning. In reality, a different choice of words inevitably alters the meaning but the human mind tolerates a certain level of variation. Machines, on the other hand, need to be programmed to handle ambiguities in natural language, which is a concern of the Natural Language Processing (NLP) field. Paraphrase identification in NLP is the task of automatically detecting whether two sentences are paraphrases.

In recent years, dense vector representations of words or “word embeddings” have been successfully used in solving natural language problems, outperforming traditional methods and achieving state-of-the-art in many tasks including paraphrase identification in English. In this thesis, we aim to apply the word embedding methods to Turkish paraphrase identification task. As far as we are aware, there is no existing study that uses word embeddings for this task in Turkish and we hope to contribute to this area.

Paraphrase identification has applications in many areas such as text summarization, question answering, information retrieval, text classification, automatic evaluation of machine translation and plagiarism detection (Mihalcea et al., 2006). Earlier studies employed various measures such as string similarity and lexical overlap (e.g. length of common word sequences) as well as semantic similarity measures based on corpus co-occurrence statistics or semantic networks. Some studies also took a hybrid approach, combining several of those measures as part of a machine learning setup. Although these traditional methods are still part of active research, word embeddings have been increasingly popular in the last decade.

1.2 Research Goal

In this thesis our main goal is to apply vector embedding methods to test if any improvement can be achieved over rule based methods for automatic paraphrase identification in Turkish. We focus on the classic task of sentence-level paraphrase identification. Specifically, we target the use of word embeddings and leave sentence embeddings out of the scope. A complementary objective, therefore, is to investigate measures to compute sentence similarity from individual word embeddings. In this sense, the main research question can be stated as "*How well can sentence similarity measures based on word embeddings predict paraphrase relation in Turkish?*".

1.3 Contribution

To our knowledge, this thesis is the first study that applies word embedding methods for Turkish paraphrase identification task. In this respect, we hope it contributes to the limited studies for Turkish natural language. We evaluate three sentence similarity measures and report a positive correlation between paraphrase relation and sentence similarity scores. In the process, we train and test several new word embeddings and report the impact of different training algorithms and parameters that may be useful for further research in Turkish.

This work also acts as a link between studies related to paraphrase identification in Turkish: word embeddings and analogy question sets (Şen and Erdoğan, 2014; Güngör and Yıldız, 2017); paraphrase corpora and identification methods (Karaoğlan et al., 2016; Eyecioglu Ozmutlu, 2016; Karaoğlan et al., 2018) and BilCol-2005 corpus (Can et al., 2005) used for training embeddings.

The work can be considered as an extension to the project number 114E126 "Türkçe Kısa Metin Benzerlik Ölçümü İçin Eminlik Faktörü Yaklaşımı ve Eş-Anlatım Derlemi Oluşturma" supported by TUBITAK, as it investigates whether word embedding methods can offer an improvement over rule based methods.

Part of the work here was also presented as a conference paper (Karaoğlan et al., 2019).

1.4 Paraphrase and Meaning

Meaning is central to paraphrase identification as well as many natural language tasks. For further discussion, a number of dictionary definitions are given below. Looking closely, each definition expresses the meaning of "paraphrase" with different words; as this sentence does:

- "A rewording of something spoken or written, usually for the purpose of making its meaning clearer." (Collins Online Dictionary, 2019)
- "A restatement of a text or passage in another form or other words, often to clarify meaning." (The American Heritage Dictionary, 2019)
- "A restatement of a text, passage, or work giving the meaning in another form." (Dictionary.com Online Dictionary, 2019)

In other words, each definition *is* a paraphrase of the other definitions. This is made possible by the fact that all definitions target the same meaning of the word 'paraphrase'.

Meaning is an elusive concept subject to philosophical discussions. A fundamental question is where it "emerges in the relationship between readers and texts" (Chandler, 1995). Objectivist approach argues meaning is entirely contained in text and reader remains in a passive role. Subjectivist approach, in contrast, maintains that meaning is entirely re-created by reader (Chandler, 1995). Constructivist approach is positioned between objectivism and subjectivism. In this point of view, meaning emerges as the result of a dialogue or construction process that takes place between writer and reader. In other words, meaning is not completely contained in text but constructed by reader to a degree (Chandler, 1995). The following excerpt from (Hofstadter, 2000) on haikus can be viewed as example of constructivism:

> "*Tortoise*: Such compressed poems with seventeen syllables can't have much meaning…
>
> *Achilles*: Meaning lies as much in the mind of the reader as in the haiku."

Following from the subjectivist or constructivist approach, one may ask: How is it possible to agree on common definitions of words if there is no objective meaning? A possible explanation is from another dichotomy. Meaning of a word can be analysed in two parts: Denotation, the literal meaning as provided by dictionaries, and connotations which are additional or subjective meanings (Chandler, 1995). In the opening paragraph, the definitions taken from different dictionaries are targeting the same denotative meaning of the word; hence we are able to assume they are paraphrases. In the strict sense of paraphrase, a pair of words cannot have identical meanings as each word has its own connotations. However, there exists an acceptable level of equivalence in meaning where two words can be identified as synonyms. The same reasoning also applies to sentence pairs where a paraphrase can be identified if sentence meanings are sufficiently similar.

Finally, the relation between context and meaning is quite relevant for this study. Firth argued that the real-life context of words contributed to the meaning of a statement as much as the actual words in it, also known as his "contextual theory of meaning" (Chapman and Routledge, 2005). As in many natural language tasks, the task of paraphrase identification usually works on sentence pairs stripped off from context. Also, mainstream text processing systems lack any audio/visual clues that may be present in the environment. On the other hand, it is possible to "embed" textual contexts from different texts into "word embeddings". This makes statistical information available for each word in accordance with Firth's famous saying "You shall know a word by the company it keeps" (Firth, 1957).

1.5 Challenges

As in many natural language tasks, paraphrase detection needs to capture meaning at a certain level and can only go so far with string and lexical similarity methods. Consider the following Turkish sentence pairs composed to illustrate edge cases. In the first example below, the given sentence pair is not paraphrase although one sentence entails the other lexically (English translations are provided in brackets):

"Dün gece İzmir'de yağmur yağdı." (It rained in Izmir yesterday.)

"Dün gece İzmir'de yağmur yağdı diyemeyiz." (We cannot say it rained in Izmir yesterday.)

In the second example, the pair is not paraphrase even though the sentences are the same except for their word order:

"Sınava az çok çalıştım." (I studied more or less for the exam.)

"Sınava çok az çalıştım." (I studied very little for the exam.)

Finally, the third pair is paraphrase despite having no common words:

"Zorluklardan kesinlikle yılmayan bir yapısı vardı." (He was definitely unyielding by nature.)

"Azimli biri olduğu su götürmezdi." (It was unquestionable that he was a determined character.)

As discussed in previous sections, the definition of paraphrase is dependent on the definition of meaning and therefore is relative. In linguistics literature a non-strict interpretation of paraphrase is favoured where it is sufficient to keep roughly the same meaning (Bhagat and Hovy, 2013) and information loss may be acceptable (Bartlett, 2006). On the other hand the strict sense of paraphrase would require an exact match which is arguably more convenient for automation but less useful. Finally, it is possible to identify some of the lexical transformations that generate paraphrases (Bhagat and Hovy, 2013) however it is not easy to do the same for semantics due the endless possibilities of expression in natural language text. The meaning can vary greatly by small changes in text or remain the same despite large changes as illustrated in the examples above.

In this thesis, our approach is to investigate if word embeddings can provide an accurate approximation of corpus statistics that performs well in the paraphrase identification task despite the mentioned challenges.

1.6 Organisation of the Thesis

This section provides a brief overview of the structure of the thesis. In the first chapter, we started by describing the motivation and the goal of the thesis. Then, we defined the research question and listed the contributions. After a discussion on meaning, we discussed the challenges of the task and provided this section on the organisation of the thesis. In Chapter 2, we provide an account of studies on paraphrase identification, paraphrase corpora and word embeddings. For English, we present an overview of significant works. For Turkish, we provide a more detailed account of the limited studies. In Chapter 3 we explain statistical language models such as N-gram and neural network language models. We describe word2vec algorithms. In Chapter 4 we describe our methodology, specifically how the new word embeddings were trained and the sentence similarity was calculated. In Chapter 5 we report the results of these experiments and provide an analysis. Finally, in Chapter 6 we compare the results to previous baselines and discuss future work.

2. LITERATURE REVIEW

This chapter presents an overview of studies on paraphrase identification as well as word embeddings and their intersection, i.e. methods that use word embeddings for paraphrase identification. We aim to cover the most relevant studies for both English and Turkish as we cannot do justice to each broad category.

2.1 Paraphrase Identification

In this section, we first discuss natural language tasks related to paraphrase identification. Then we provide a chronological summary and follow with a thematic account of various methods. Finally, we describe the studies for Turkish.

2.1.1 Related tasks

Studies relevant to paraphrase identification (PI) are also found under different keywords such as “paraphrase recognition” and “paraphrase detection”, which are used interchangeably with paraphrase identification (Ul-Qayyum and Altaf, 2012). In addition, studies on closely related natural language tasks such as Semantic Textual Similarity and Recognising Textual Entailment are relevant to PI. Semantic Textual Similarity (STS) aims to calculate a graded similarity score from a pair of texts, e.g. an integer in the range [1, 5]. In comparison, the PI task produces a binary classification label (Mihalcea et al., 2006). STS score can be used in paraphrase identification. For example, it can act as a feature in machine learning methods or it can be converted to a binary paraphrase label by comparison to a defined threshold. Indeed, the latter is the approach taken in this thesis. Recognising Textual Entailment (RTE) aims to decide whether the meaning of one text can be inferred from another text (Dagan and Glickman, 2004). A binary label is produced as output as in the PI task, however textual entailment represents a unidirectional relation whereas paraphrase is a bidirectional relation (Agirre et al., 2012). If a pair of texts entails each other, it can be logically argued that they have paraphrase relation. A detailed account of the relation between paraphrase and textual entailment relations can be found in (Androutsopoulos and Malakasiotis, 2010).

2.1.2 English paraphrase identification

From a thematic perspective, we assume paraphrase relation is a specific form of textual similarity, which can be categorised into string-based, corpus-based and knowledge-based similarity approaches (Gomaa and Fahmy, 2013). String-based similarity is a type of lexical similarity, which takes into account the character sequences of strings, whereas corpus-based and knowledge-based similarities measure semantic similarity. Other studies that classify PI approaches are (Islam and Inkpen, 2008), which is a previous study also on textual similarity, and (Androutsopoulos and Malakasiotis, 2010) that gives an extensive account of works on PI.

From a chronological perspective, Microsoft Research Paraphrase Corpus (MSRP) (Dolan et al., 2004) can be seen as a milestone in paraphrase identification even though prior research exists. This work supplied a standard dataset for evaluation and comparison of different methods, contributing greatly to the progress in the area. The earlier approaches in the same period mostly relied on string similarity and lexical overlap (Mihalcea et al., 2006). Semantic similarity was measured through semantic networks (e.g. WordNet), and corpus-based measures such as LSA (Latent Semantic Analysis) or PMI (Point-wise Mutual Information) based on word co-occurrence counts (Islam and Inkpen, 2008). Paraphrase identification studies that use word embeddings are covered in more detail in section 2.3. Briefly, word embeddings started gathering interest with (Collobert and Weston, 2008), which has its roots in (Bengio et al., 2003), and grew in popularity after the studies of Mikolov et al. (2013a; 2013b) in many natural language tasks including PI. In the following paragraphs, we provide reviews of relevant studies.

The initial approaches to the task of paraphrase identification were mostly based on lexical methods, such as measuring the lexical match between texts as discussed in (Mihalcea et al., 2006), finding overlapping vocabulary or calculating edit distance (Dolan et al., 2004). Finch et al. (2005) consider the PI task as similar to automatic machine translation (MT) evaluation in which the sentences in MT output are compared against reference translations to determine if there is sufficient similarity. They train a machine learning classifier that incorporates several MT evaluation methods, using two corpora including MSRP. The approach achieves comparable results to state of the art at the time even though MT evaluation methods are all syntactic and lexical, in other words not semantic.

Mihalcea et al. (2006) differ from previous research in that they focus on short texts instead of large documents, and take the semantic similarity between words into account rather than follow a pure lexical approach. This study calculates a similarity score between short text segments based on a combination of what is categorized as corpus-based and knowledge-based word similarity measures. Corpus-based measures use the information obtained from large corpora such as word co-occurrence counts; knowledge-based measures use the similarity information from semantic networks such as WordNet. The similarity score is calculated from six knowledge-based and two corpus-based measures, namely pointwise mutual information (PMI) and latent semantic analysis (LSA). This metric is evaluated in a paraphrase classification task using the MSRP corpus. Labels are obtained by comparing each similarity score to a threshold of 0.5, however why this value is selected is not explained. The metric achieves better F-score performance than each of the measures used independently as well as two other baselines computed by the authors due to lack of existing baselines at the time. This study itself establishes a baseline for future work.

Islam and Inkpen (2008) base their study on semantic text similarity arguing that paraphrase relation is the result of similarity or difference relation between two texts. This study also focuses on short texts rather than documents as in (Mihalcea et al., 2006). It proposes a new method, called Semantic Text Similarity or STS, to measure text similarity of short text segments in contrast with previous studies targeting large documents. STS combines string similarity as well as semantic similarity, and optionally common-word order similarity. It is argued that string similarity can capture similarity where knowledge-based or corpus-based similarity methods cannot, such as in the case of simple typographical errors. STS uses a modified version of longest common subsequence (LCS) measure for string similarity, and a modified version of PMI for semantic similarity. Even though STS is an unsupervised method on its own, it is considered supervised for the paraphrase classification task. This is because a threshold value is needed to decide whether a similarity score represents paraphrase or not. STS scores are tested with 11 similarity threshold scores against the MSRP corpus training data set. The best threshold value found here also yields the best results in the test set. It is worth adding that this value is specific to the STS method and were the method's formulation changed, tests to find the optimal threshold would need to be repeated. This study does not report a performance improvement over (Mihalcea et al., 2006) in the paraphrase classification task, however claims to have better time complexity as it uses a single corpus-based measure compared to several different measures.

Guo and Diab (2012) test their sentence similarity method on the PI task. It is argued that sentences are too short to get a good representation of the context and therefore the set of missing words are explicitly taken into account in their model, called weighted textual matrix factorization, in such a way that missing words are not overpowered in comparison to constituent words. The similarity metric is also tested in the PI task using the MSRP corpus. Similarity score is converted to a classification label using a threshold derived from the training data only. The result does not advance state-of-the-art which is (Islam and Inkpen, 2008) at the time, but it is quite close.

Ji and Eisenstein (2013) reports state-of-the-art results in the PI task against the MSRP corpus. Firstly, they introduce the TF-KLD term weighting method that emphasizes discriminative features using labelled training data. Secondly, they follow a supervised classification approach where additional lexical features (such as BLEU and n-gram overlap) can be used as well as the feature vectors resulting from factorization. Finally, these additional features are concatenated to the distributional feature vectors. Paraphrase label is decided by applying a threshold to the cosine distance between final vector representations. The threshold is tuned only from the training data. The experiments are conducted in inductive and transductive settings where the latter implies access to test data as well as training data. To our knowledge, the results in inductive setting are still state-of-the-art.

A relatively new study (Issa et al., 2018) builds upon (Ji and Eisenstein, 2013) making use of SVMs, rather than mainstream neural network approaches, owing to its efficiency in their specific case. For the transductive setting as explained above, a significant improvement is reported in state-of-the-art, while in the more commonly used inductive setting the results are not competitive.

Issa et al. (2018) also describe a category of “distributional methods for paraphrase identification” for studies where the semantic similarity of sentence pairs is measured by distributional representations. In this category several previous studies are mentioned. (Turney and Pantel, 2010) is a survey on vector space models (VSM) used for semantic processing of text. It is a detailed study that starts with the fundamentals of the VSM and examines, among other topics, the link between VSMs and the distributional hypothesis. Other relevant studies already discussed elsewhere in this chapter are (Mihalcea et al., 2006; Mitchell and Lapata, 2010; Guo and Diab, 2012; Ji and Eisenstein, 2013). The work done in our thesis can also be categorized under this category as we make use of word embeddings that are distributional representations.

2.1.3 Turkish paraphrase identification

The research on PI for the Turkish language is limited and existing studies are focused on traditional methods. We will first focus on the paraphrase corpora and then the paraphrase methods.

The first paraphrase corpus for Turkish is created by Demir et al. (2012). It contains 1270 sentence pairs collected from multiple translations of English sources, multilingual parallel corpora for machine translation and news articles paraphrased by native speakers. To the best of our knowledge it is not publicly available.

PARDER (Karaoğlan et al., 2016) is the main Turkish paraphrase corpus used in this thesis work. It includes a total of 6510 sentences pairs with 4421 paraphrase and 2089 non-paraphrase instances collected from a diverse set of sources such as news corpora and translated sentence pairs as well as user generated pairs. Paraphrase candidates are found from these corpora using various sentence distance metrics and presented to human annotators for review. For each sentence pair, annotators can supply binary paraphrase label, encapsulation relation and opposite flag. In comparison with other Turkish corpora, it includes more sentence pairs and reports a higher agreement rate between annotators. Fleiss' Kappa score is a reliability measure for annotator agreement that can work for more than 2 annotators (Fleiss, 1971), where scores are calculated in the unit range [0, 1]. For this study, the Fleiss' Kappa score is reported as 0.634 which may be interpreted as a high agreement rate. The corpus was also augmented further and used as part of (Karaoğlan et al., 2018).

The Turkish Paraphrase Corpus, TuPC, is another Turkish paraphrase corpus developed by Eyecioglu Ozmutlu (2016). It includes a total of 1002 sentence pairs with 563 paraphrase and 285 non-paraphrase instances. 154 of sentence pairs with an unclear similarity score are classified as debatable. The data is collected from several Turkish news sites. Graded similarity scores are provided instead of binary paraphrase labels to make the data available for sentential semantic similarity tasks as well as paraphrase identification. For annotators, however, providing a graded similarity score is reported to be more difficult than providing a binary label, resulting in an average inter-rater agreement rate (Fleiss Kappa: 0.42). With respect to process; sentence pairs are collected from newspapers, filtered based on lexical overlap and difference length and finally handpicked. The corpus is available publicly.

In terms of paraphrase identification methods, (Karaoğlan et al., 2018) is the most recent relevant work. In this study, a Bayesian classifier is trained from PARDER corpus (Karaoğlan et al., 2016) with syntactic and semantic features to identify paraphrases. The syntactic features are based on overlapping words and word order. The tense and polarity of sentence are added as semantic features, which are extracted using natural language processing methods such as part-of-speech tagging and morphological analysis. The study shows that semantic features contribute to the overall classification performance.

In the study by Eyecioglu Ozmutlu (2016), a support vector machine (SVM) classifier is trained from the TuPC paraphrase corpus described in the same study using character and word n-gram features. Overlapping character bigrams are found to be the best performing feature among other combinations of character and word n-grams including larger values of n, as well as similarity measures employed.

2.2 Word Embeddings

2.2.1 Definition and history

Word embedding is a method to map words to vectors of real. Resulting vector representations are called “word embeddings”. This allows performing vector operations on word representations which is useful to exploit available national language processing methods. In this section we provide a brief history.

Tracing back the origins of the idea, one can find that the concept of distributional word representations is based on the distributional hypothesis in linguistics, stating that words that occur in similar contexts have similar meanings. This is famously expressed by the saying “You shall know a word by the company it keeps” (Firth, 1957). The idea in general can be further traced back to the works of Wittgenstein (Turney et al., 2010). It can be categorised within statistical NLP methods, which take an empirical approach to language rather than the rule-based methods that follow a rationalist approach (Manning and Schütze, 1999).

The procedure to learn representations using intermediate layers of neural networks is first proposed in (Rumelhart et al., 1986). Bengio et al. (2003) makes use of this procedure in order to learn distributed feature vectors for words as a solution to the “curse of dimensionality” problem. The issue is basically that the vector space becomes sparse due to high-dimensionality, which we provide more details in the next chapter. Continuous representations can avoid this problem as they need only a fixed number of feature dimensions and the vector space does not grow as fast. Feature vectors learned in (Bengio et al., 2003) are a type of distributed representation where each word is represented across many feature vectors, and each feature is used in representing many words. Feature vectors use continuous real values. It is also worth noting that it is an expectation set in this paper that words with similar syntactic and semantic roles should have a similar vector.

The work of Collobert et al. (2011) plays an important role in the adoption of neural network based approaches in NLP building on (Bengio et al., 2003). In this study, a multi-layer, or deep, neural network is trained so as to create a "unified architecture" which can be used in a multitude of NLP tasks, largely in contrast with the studies at the time focusing on hand-engineered features. Word embeddings, constituting the first layer of the deep network, are learned from unlabeled data of English Wikipedia with the only pre-processing step being conversion to lower case. The training of these embeddings is reported to have taken one week. An important difference to Bengio et al.'s method is in the selection of context words during training. Collobert et al. select the words before and after the target word, whereas Bengio et al. select only the words before the target word. This difference is due to the fact that Bengio et al. build a language model which by definition predicts the next word given a previous list of words, while Collobert et al. aim to improve the representation of words. In the experiments, words that are out of vocabulary are mapped to a special word. An evaluation of resulting embeddings is to find the closest words in vector space using simple Euclidian distance. Reported results seem quite accurate, for example giving "Spain" for "France" and, when tested with a much rarer word, "smashed" for "scratched".

Mikolov et al. (2013a) were influential in making word embeddings popular by achieving state-of-the-art results in several tasks. They mainly build upon (Bengio et al., 2003) and (Collobert et al., 2011) to create a probabilistic model that learns distributed representations of words, also known as word embeddings, in a highly efficient manner. The model produces remarkably good results in analogy tasks as the word embeddings can capture semantic properties of words in addition to syntactic properties, as suggested by Bengio et al. (2003). Details of the method are described in the Background chapter.

GloVe, short for global vectors, is another set of distributional word embeddings proposed in (Pennington et al. 2014), which can be considered one of the studies prompted by Mikolov et al.'s work (2013a). Unlike CBOW and Skip-Gram, GloVe uses a global co-occurrence matrix which can detect words with very high frequency. Indeed, Mikolov et al. use sub-sampling to balance the impact of most frequent words as an improvement (2013b). Although GloVe was reported to outperform CBOW (Pennington et al. 2014), later studies showed these two approaches do not have a decisive advantage over each other (Levy et al., 2015) and the differences in performance were mainly explained by peripheral concerns such as parameter selection.

2.2.2 Turkish word embeddings

There are several studies on word embeddings learned from Turkish corpora. (Şen and Erdoğan, 2014) is the first such study that is known to us. In this study, training corpus is built from Turkish Wikipedia and online Turkish news sources. Before rare words are removed, less frequent words are split into stem and suffix, which is argued to improve the quality of embeddings in Turkish. For evaluation, question sets are prepared for the analogy task similar to (Mikolov et al., 2013b) and another task to find the odd word among similar words. In terms of training algorithms, negative sampling is found to perform better than hierarchical soft-max. Also, stemming is found to give better results in the analogy task. Finally, it is observed that performance increases with larger vector dimensions up to a certain point, after which the effect is reversed.

In a similar study, Güngör and Yıldız (2017) create new Turkish word embeddings from a larger corpus. They also add new test sets of analogy questions to better evaluate these embeddings, building upon the work by Şen and Erdoğan (2014). These questions are aimed at testing Turkish words with derivational suffixes, inflectional suffixes and both type of suffixes at the same time. In order to create the questions sets, existing corpora are scanned and morphological analysis applied. Additional questions are added by starting from common roots and appending both types of suffixes. Results indicate that word embeddings work well with Turkish, but taking advantage of the morphologically-rich nature of Turkish can still help.

There are also studies on different tasks that make use of word embeddings. In these studies, researches have mostly built their own word embeddings. Pembeci (2016) learns embeddings from Turkish newspaper articles for use in ontology enrichment task. Bigram, trigram and four-gram phrases are captured based on the approach in (Mikolov et al., 2013b). No special pre-processing is applied. Although negative sampling is reported not to provide performance improvement in this task, it is still preferred owing to its training speed. Resulting word embeddings are found effective for the task. Similarly, Demir and Özgür (2014) investigate the use of word embeddings in Named Entity Recognition task, obtaining satisfactory results without any language-specific features.

Common opinion from the studies reviewed above is that, although there may be room for language-specific improvements, word embedding methods work well with Turkish in general. This is encouraging for the direction of the thesis which is to use word embeddings for the downstream task of paraphrase identification.

2.3 Paraphrase Identification using Word Embeddings

There are several methods that can be used for paraphrase identification from word embeddings. It is also possible to directly use sentence embeddings, however methods that learn sentence embeddings from corpora is a large topic that may deserve a separate study, therefore it is left out of the scope of this thesis. Still, it is essential to mention the Paragraph Vectors introduced in (Le and Mikolov, 2014) due to its relevance and similarity to Skip-Gram and CBOW architectures (Mikolov et al., 2013a; 2013b). In this study, fixed-length vector representations of texts, or paragraphs, are learned from variable-length text sources in a similar way to how the word embeddings are obtained in (Mikolov et al., 2013a). Paragraph vectors are tested in a variety of NLP tasks such as sentiment analysis and information retrieval and found to achieve state-of-the-art performance.

One of the approaches to decide if two sentences are paraphrases is to calculate their similarity. Here, it is assumed that paraphrase relation may be correlated with similarity score, which is an idea tested in this thesis. A basic approach to obtain a sentence representation is to average or concatenate individual word vectors. Both methods are simple to calculate and also provide favourable results as employed in this work. More details are provided in the Methodology chapter.

Another method for paraphrase identification is to calculate the similarity of sentences without building an explicit representation of sentences. "Word Mover's Distance", or WMD, is a method that calculates the distance between two texts based on word embeddings (Kusner et al., 2015). In essence, the algorithm treats text similarity as a transport problem where each word in document 1 needs to move the minimum distance to travel to document 2 (Figure 3.1). This method is also used in this thesis and achieves the best performance among tested sentence similarity measures as discussed in the Experiments chapter. A supervised version of this measure is also published (Huang et al. 2016) but is not covered in this thesis.

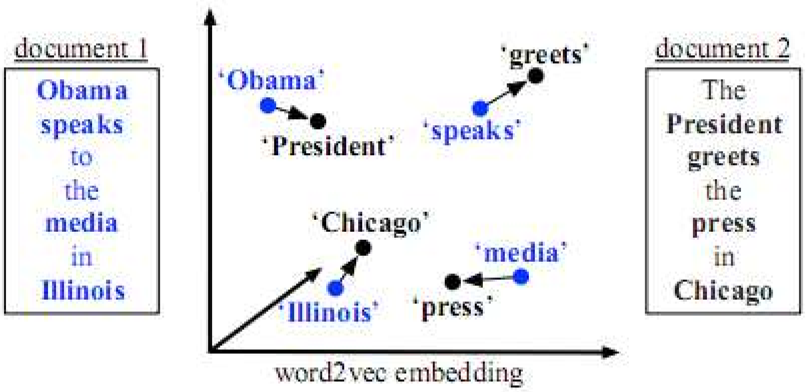

Figure 2.1. "An illustration of the word mover's distance" (Kusner et al. 2015)

Mitchell and Lapata (2010) investigate the compositionality of different functions to build a useful representation of phrases from individual word vectors. They categorise semantic representation methods into three groups: semantic networks, feature-based models and semantic spaces which also include probabilistic models. Semantic representations of two-word phrases are obtained using different composition functions that use vector addition and multiplication.

The similarity scores from these functions are then compared to ratings given by human judges. It is found that a different function can be suited for each type of word combination such as noun-noun or verb-object. The question of how to scale this approach to obtain a sentence level representation is left open, with the comment that a single function can be applied recursively to the binary parse tree of the phrase or sentence.

3. BACKGROUND

This chapter provides details on the concepts and significant technical methods referred to in this thesis.

3.1 Statistical Approach in NLP

The birth of statistical approach in NLP can be mostly attributed to the limitations of rule-based approaches versus the flexible and ambiguous nature of human language. Gradually, it was seen as an unmanageable task to maintain formal grammars of language and hand-written rules to capture the exceptions (Nadkarni et al, 2011). In this context, statistical approach has given rise to the more active usage of machine learning methods and large corpora.

As the thesis is focused on word embeddings, we will not attempt to cover all statistical methods and cover only closely related concepts, such as traditional n-gram models, their shortcomings and improvements from neural network models.

3.2 Language Models

Statistical language models aim to predict the probability of word sequences. Specifically, they are used to predict the next word given a previous set of words. This is usually a corpus-based method where the model learns the probabilities of word sequences from large corpora. A problem with this method is that it is not possible to cover *all* possible sequences in a language even if the input corpus is large, which causes data sparsity. Different approaches are taken to solve this problem as discussed in different language models below.

3.2.1 N-gram language models

In natural language, “n-gram” means a sequence of n items such as characters or words. N-gram language models focus on word sequences such as bigrams (2-word) or trigrams (3-word). They are typically used to estimate the probability of word sequences based on the count of n-grams in a given training corpus. For example, the bigram model estimates the next word given the previous word or the trigram model estimates the third word given previous two words.

N-gram models do an approximation by taking into account only the last N-1 words instead of *all* previous words. In the latter case, one would need an extremely large training data set in order to make a meaningful estimation of the probability of long word sequences. Even at this age of abundant data, the amount of available data would remain too small as the length of sequence increases. N-gram models try to alleviate the problem of data sparsity by making a simplifying assumption that the model carries the Markov property, which basically indicates that future states of the model do not depend on previous states. In the case of language models, this is an approximation that enables the model to make predictions on a much smaller set of previous words compared to all preceding words.

Despite this simplifying assumption, however, the size of the count matrix used in an n-gram model grows quickly as new dimensions are added. Available training data usually cannot grow at the same rate and this causes the matrix to become sparse. This is described as the curse of dimensionality in (Bengio et al., 2003). Neural network models offer a different approach to this problem.

3.2.2 Neural network language models

The basic idea of neural network language models (NNLM) is to have a hidden layer which learns the weights to represent each word through training over large a corpus. As discussed in the literature survey, implementation of this idea has evolved over time making it possible to create high-quality word representations in a shorter time. There are different types of NNLMs such as feed-forward neural network and recurrent neural networks. A detailed discussion of neural network models is provided in (Goldberg, 2016).

The end product of NNLMs is usually a matrix where rows are words and columns are dimensions. Dimensions can be thought of as implicit features auto-generated by NNLM. Values in this matrix are real numbers that represent the weights of features for each word. Thus, resulting word vectors are dense, continuous representations of words. In effect, a many-to-many relationship is established between words and dimensions where each word is represented by all dimensions and each dimension contributes to the representation of each word. In this way, the curse of dimensionality is eliminated as the data is not sparse and the number of dimensions is fixed.

Representations obtained by neural networks are referred to with different terms such as word vectors, word representations or word embeddings. We use these terms interchangeably unless specified otherwise.

3.3 Word2vec

Word2vec is the software library published by Mikolov et al. (2013a) that implements the proposed architectures which are continuous bag-of-words (CBOW) and Skip-Gram. The main idea of these architectures is to build a predictive neural network model that is trained by a large body of text, as discussed in the NNLM section above. The neural network proposed in this approach has a single hidden layer except input and output layers. During training, weights of this hidden layer are learned which finally become the word vectors. This method is primarily an improvement over n-grams, which suffer from the curse of dimensionality as explained in the previous section, as well as over "deep" neural works which have may have several hidden layers that increase the computational complexity of models and hence the training duration. A shallow neural network that uses a log-linear classifier proves to be more efficient in comparison and enables training of much larger corpora.

In the continuous bag-of-words architecture, neural network predicts the current word based on the words in the context window. In the skip-gram architecture, the prediction is reversed, in other words context words are predicted from the current word. Figure 3.1 and 3.2 show a diagram of these architectures, with a context window of size 4.

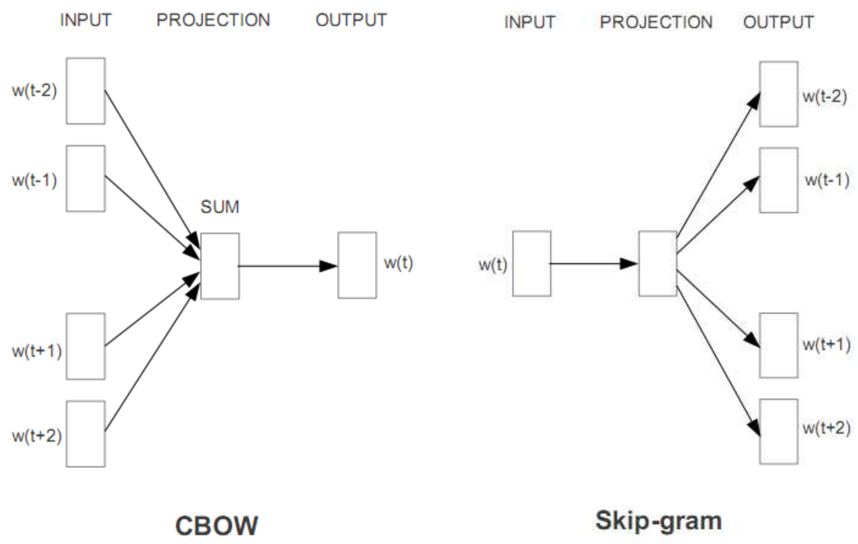

Figure 3.1. CBOW and Skip-gram architectures (Mikolov et al. 2013a)

The skip-gram model is further enhanced in (Mikolov et al. 2013b) by several improvements. First is the usage of hierarchical softmax algorithm in place of the full softmax. The purpose of softmax algorithms is to ensure a proper probability distribution in the output nodes of neural network, such that each node probability will be in the interval (0, 1) and the total probability will be equal to 1. Second is the usage of negative sampling which is an approximation that is applicable in the context of Skip-gram architecture. Third, subsampling of frequent words is a method to decrease the occurrences of the most frequent words. The actual step is to discard the word with a probability that increases with word frequency. This helps both the training speed and as argued, the accuracy.

As part of their research, Mikolov et al. (2013a) trained and shared high-quality word embeddings trained on Google News corpus that has 6 billion words, as well as the library that can be used to efficiently learn new embeddings. The availability of these resources led to more research in the field that makes use of word embeddings.

3.4 GloVe

GloVe, short for global vectors, is another set of distributional word embeddings. Although it is not directly used in our work, we include a brief description of this study (Pennington et al. 2014) as it can be considered one of the studies prompted by (Mikolov et al., 2013a). The main idea is the observation that Skip-gram and CBOW architectures do not make use of a global co-occurrence matrix, and update weights only by passing a window through the whole corpus,. However, there might be some crucial info that can be gathered from such a matrix such as the overall frequency of words which may be too high. Indeed, Mikolov et al. also needed to introduce subsampling for the most frequent words (2013b). Thus, they make use of global co-occurrence matrix as well as a local window. Finally, it is observed that the two models are mathematically quite similar. Although as initially reported GloVe seemed to outperform word2vec, later studies showed these two approaches do not have a decisive advantage over each other (Levy et al., 2015).

3.5 FastText

FastText is a model that represents each word with the n-grams it is made up of, rather than representing it as a single entity (Bojanowski et al. 2017). Each word vector in this model is the sum of its n-gram vectors. The model is able to compute word vectors for out-of-vocabulary words by summing its n-grams. This enables the model to cover rare words especially in languages with a large vocabulary such as Turkish, which has a very large number of distinct words due to its morphologically rich nature.

Pre-trained fastText embeddings are being distributed for 157 languages (Grave et al. 2018). The pre-trained embeddings for Turkish has been tested with Turkish analogy question sets and are found to achieve the best results among other embeddings.

4. METHODOLOGY

Our main goal is to investigate the performance of paraphrase identification in Turkish using sentence similarity measures based on word embeddings. The method can be summarised as follows. We identify several sentence similarity measures that calculate the distance between two sentences. Here, we use a geometric concept (distance) to express the dissimilarity in meaning because model words and sentences in the vector space and measure their similarity by cosine or Euclidean distance. The distance cannot determine paraphrase relation on its own; a threshold is needed to convert it to a binary label. Each measure may produce distance values in a different range and, thus, have a different threshold. To find the best threshold for a measure, we use k-fold cross validation. In this technique, dataset is split into training and test datasets in *(k-1):1* ratio and tests are repeated *k* times with a different dataset pair. At each split, the distance of all sentence pairs in the training set is calculated, and best threshold to maximise performance is found. We take the average of these *k* thresholds as the best threshold for the sentence similarity measure under test. Similarly, the performance is calculated as the average of *k* best performance values.

Our work can be described in three phases. First, we conducted a basic literature review on the usage of word embeddings in paraphrase identification inspired by their recent success in many natural language tasks. For English we found similar studies with successful results but this was not the case for Turkish. Therefore we decided to pursue this area of study, also as a continuation of the existing research project on Turkish paraphrase corpus mentioned in the introduction chapter.

In the second phase, we carried out a basic experiment to see whether the hypothesis can be quickly rejected. New Turkish word embeddings were learned from existing research corpora and other online resources. Then, vector representations of sentences were obtained by simple average of word embeddings and sentence distances were measured using cosine similarity. The results were promising even though the embeddings were not optimised in terms of training parameters or pre-processing steps. Therefore we continued with the study.

In the last phase we tested word embeddings trained with various parameters and used other sentence distance measures. Results were evaluated using cross validation techniques. Finally, we analysed the results and compared them to existing baselines.

4.1 Word Embeddings

Depending on the actual task, the optimal settings for learning word embeddings may change (Mikolov et al. 2013b). Therefore, we decided to learn new word embedding, which allowed us to control the selection of text sources, pre-processing steps and training parameters.

To learn the embeddings, the Turkish Wikipedia and the Bilcol-2005 corpus (Can et al., 2005) are used as the input texts. Both corpora contain a large amount of Turkish text and are available for research purposes. Combining two corpora increases the amount of training data, which is assumed to increase the word embedding quality.

The process to create the input file for training can be summarised as follows. Each corpus was first acquired; then cleaned up for the task; and then converted into a file that contains one sentence per line. Finally, the two files were combined into a single file, being ready for training. The details for each corpus are provided below.

The Turkish Wikipedia was obtained online as a WikiMedia dump file from 01/08/2019. The documents contain special syntax and keywords called “Wiki markup”. To obtain raw text, Wiki markup was removed using the “WikiExtractor” script (Attardi, 2019). Next, the text was split into into 4,052,263 sentences using the Python NLTK library (Bird et al. 2009). Letter case was not modified at this point.

The BilCol-2005 corpus (Can et al., 2005) was provided by the researchers upon our application. It contains 61,684 xml files, each of which contains several sentences. The content of each document was extracted and split into sentences using the Python NLTK library. This resulted in 689,862 sentences using Turkish language option. Finally, sentences shorter than 10 characters were mostly found to be errors in sentence splitting and therefore discarded. Letter case was not modified at this point.

Sentence splitting, or sentence segmentation, is a topic on its own. For both corpora, NLTK is used to tokenize texts into sentences. This is mainly due to the simplicity of development and runtime efficiency considering that the rest of the development is done in Python. As we are not aware of any data set for evaluation of Turkish sentence splitters, we inspected the results visually. We observed that the Turkish option did slightly better in handling abbreviations, although it still had issues handling abbreviations. Overall, results seemed equivalent, each option having its strengths and weaknesses.

The final input file contained a total of 4,742,125 sentences or approximately 64 million words. It was then fed into word2vec library to create the word vectors using Python library gensim (Řehůřek and Sojka, 2010). For pre-processing, we used gensim's "simple_preprocess" function with default parameters which converts the text into lowercase and discards tokens with shorter than 2 characters and longer than 15 characters. Table 4.1 shows the main parameters used for training word embeddings and their descriptions.

4.2 Paraphrase Corpora

Several paraphrase corpora were used to test our methods against. Table 4.1 below lists these corpora and a break down of sentence pairs in relation to paraphrase classification. The first column gives the short name of the corpus. Remaining columns display respectively the number of total sentence pairs, the number of paraphrase pairs and non-paraphrase sentence pairs and the ratio of paraphrase pairs to total number of pairs.

Table 4.1. Paraphrase corpora used in the experiments

Corpus	Pairs	Paraphrase	Non-paraphrase	Paraphrase ratio
PARDER	6510	4421	2089	68%
TuPC	1002	563	285	56%
COM	7512	4984	2374	67%
MSRP	5801	3900	1901	67%

In terms of corpora used, PARDER refers to (Karaoğlan et al., 2016), TuPC to (Eyecioglu Ozmutlu, 2016), COM to a combined version of the sentences pairs in PARDER and TuPC, and finally MSRP to (Dolan et al, 2004). Details of the first two corpora are provided in the literature review chapter. (Demir et al. 2012) is not included in our tests as it is not publicly accessible. The combined corpus is prepared in order to make a rich set of test targets and taken as the main corpus on which we base our tests. MSRP corpus was only for a basic check of whether our method obtained sensible results for English as well.

A basic pre-processing step was applied to sentence pairs read from paraphrase corpora. In this step, Gensim Python library (Řehůřek and Sojka, 2010) was used as done in training word vectors. Lowercase operation was customised to properly handle the "Turkish İ" character. Additionally, the apostrophe character was removed for consistency in handling suffixes appended to proper nouns and common nouns. This is because, in Turkish, suffixes to proper nouns need to be added with a leading apostrophe. There were also many sentences where opening and closing quotes did not match; these were left as is.

4.3 Sentence Similarity Measures

Measuring semantic similarity or sentence distance between sentences is a large task that is also related to sentence representations and compositionality of meaning. We focus on methods based on word vectors where a sentence is represented in the vector space as a function of its constituent word vectors. Once the vector representations are obtained for a pair of sentences, their similarity can be calculated based on the simple Euclidian distance of vectors.

The first method we used to obtain sentence distance is calculating the average of word vectors, which we will call in short AWE – average of word embeddings. Resulting sentence vectors are of a fixed size same as that of word vectors. Distance is found by first calculating the cosine similarity between sentence vectors and then subtracting it from 1 to inverse it. As cosine similarity is always in the unit interval [0, 1], so is the distance.

$$dist = 1 - similarity$$
$$dist = 1 - \frac{u \bullet v}{\| u \| \cdot \| v \|}$$

The second method we used is constructing a sentence vector by concatenating all constituent word vectors, which we will call CWE – concatenation of word embeddings. Since each sentence has an arbitrary number of words, resulting sentence vectors often have different sizes. To make these vectors of equal size, the shorter vector is "padded" as many times as needed with a word vector of choice. Distance is then calculated as the cosine distance between sentence vectors as in AWE. We chose one of the most common words in each language ("bir" in Turkish and "the" in English) as the padding word, so that it has minimum impact on sentence meaning. We also experimented with different padding words which can change the score visibly.

The third method we used is "Word Mover's Distance", or WMD, which is based on word vectors as described in (Kusner et al., 2015). It is different from AWE and CWE in that it does not create an explicit vector representation of sentences. Instead, it calculates the distance directly from the word vectors of each sentence. Assuming a pair of sentences A and B, this method calculates the distance of the closest word in B for each word in A. The sum of these distances gives the total distance.

Out-of-vocabulary (OOV) words are words that are not seen in training data set but appear in test data set. Consequently, the model does not have a word vector for those words. It is desirable to minimise the number of OOV words as the distance calculations may be quite inaccurate especially if the model misses highly representative words, i.e. words which can have an impact on the similarity scores. It is also an important design choice how to handle OOV words when they occur: They can be ignored or replaced with a very frequent “placeholder” word to the same effect. In our case this choice is also dependent on the type of vector arithmetic used by the sentence similarity method. For example, AWE method can either ignore an OOV word in the sentence or replace it with a placeholder word because the input is valid in both cases for average calculation. CWE, on the other hand, needs to use a placeholder word as it depends on two equal sized sentence vectors. The impact of vocabulary size and OOV words in our tests is discussed in the following chapter.

Stop word removal is a common NLP technique applied as part of pre-processing. However, we did not remove stop words during training as we wanted to see how well embeddings would perform as learned from raw text.

For verification of the distance calculations, we prepared test scenarios such as “identical sentences should report zero distance” and “distance calculation should be case-insensitive” based on additional sentence pairs we composed for the purpose. The impact of word repetitions and word order were also tested with sentence pairs.

5. EXPERIMENTAL RESULTS

In this section, we first describe the process of training new Turkish word embeddings and our approach to perform intrinsic evaluation of these embeddings. Next, we describe the sentence similarity measures used to predict the paraphrase relation. For each similarity measure, the distribution of similarity scores over the paraphrase pairs is analysed. In addition, sample paraphrase pairs and their distance scores are inspected.

5.1 Training Word Embeddings

A laptop PC with an average configuration (Intel Core i5, 4GB RAM) was used to learn the embeddings. All embeddings were trained from a combination of Turkish Wikipedia and BilCol-2005 corpus, using the Skip-Gram architecture with a window size of 10. We experimented with vector size, minimum count and iteration count which are described in Table 5.1.

Table 5.1. Gensim Training parameters (Řehůřek and Sojka, 2010)

Parameter	Description
Window	Max. distance between current and predicted word within a sentence
Size	Dimensionality of the word vectors”
Min.count	Ignores all words with total frequency lower than this.
Iterations	Number of iterations (epochs) over the corpus.

Table 5.2 displays the training parameters and the resulting vocabulary size and training time. Hierarchical softmax and negative sampling algorithms were both tested. Vocabulary size is dependent on the minimum count parameter which dictates the removal of words below the given frequency. Similarly, training time is to be largely dependent on the number of iterations. Considering the training corpus has approximately 64 million words and an average PC is used, training durations of 1 to 7 hours may be interpreted as an indicator of the efficiency of word2vec algorithm.

Table 5.2. Training statistics for word embeddings

Size	Min. count	Iterations	Vocabulary	Training Time
200	2	5	726539	7h
300	2	1	726539	1h
300	2	3	726539	3h
300	2	5	726539	7h
400	2	5	726539	7h
300	10	1	226151	1h
300	10	3	226151	3h
300	10	5	226151	7h

In our tests, the ratio of out-of-vocabulary (OOV) words is calculated as 14% with respect to the target paraphrase corpus when 'Min. count' parameter is set to 10, and drops to approximately 6% when 'Min. count' is 2. The embeddings with larger vocabulary and lower OOV ratios achieved better results with the same training data as explained in this chapter. We believe this should be highly related to the agglutinative nature of Turkish where suffixes can create many variations of the same word. For both AWE (average of word embeddings) and CWE (concatenation of word embeddings), ignoring OOV words gave better results than appending the most frequent word of the language. Similarly, WMD (Word Mover's Distance) implementation in Gensim library removes OOV words before the computation.

5.2 Evaluation of Word Embeddings

Word embeddings can be evaluated intrinsically and extrinsically. Intrinsic evaluation techniques aim to test the overall, inherent quality of word embeddings and provide a quicker way compared to extrinsic evaluation. As an example, Mikolov et al. (2013c) create a new dataset for testing various word embeddings on the analogy task. Synonym detection, semantic similarity and other intermediate tasks which supply test datasets can also be used for this purpose. Extrinsic evaluation techniques, on the other hand, measure the quality of word embeddings on the actual task at hand. To assist qualitative analysis, word embeddings can also be visualised on two- or three-dimensional plane using methods such as PCA, t-SNE (Maaten and Hinton, 2008) or UMAP (McInnes et al. 2018).

The following sub-sections discuss two intrinsic evaluations of the word embeddings and summarise our extrinsic evaluation method.

5.2.1 Evaluation of Most Similar Words

For a basic intrinsic evaluation of word embeddings, we use Gensim Python library to list most similar words and check if results are intuitive from the perspective of a native Turkish speaker. For several sample words below, top-10 most similar words are listed in order of similarity score from highest to lowest. For ease of reference, rank is given in a separate column. Approximate English translation of each word is also provided. Finally, we add our evaluation of the results for each sample word.

In these samples it is not possible to calculate precision and recall in the sense discussed in “Evaluation Measures” because this is a subjective evaluation. It can be roughly interpreted as high precision if native speakers agree that proposed similar words make sense. Recall is more difficult to measure because it would require native speakers to agree on the most relevant words for a given word, which is more opinion-based. Therefore we limit the scope of our evaluation to deciding whether produced list of similar words make sense.

As the first example, Table 5.3 lists the most similar words to “çay” (tea). “Kahve” (coffee), “demlenir”/“demlemek” (brewed/to brew), “şeker” (sugar), “içecekler” (drinks) and “çayın” (tea + possessive suffix) are correctly identified as similar words. Other suggested words are more distantly related and especially the word at rank 6 is unrelated. In general, however, the list can be argued to make sense. Inflectional forms of the same words (such as “çay”-“çayın” or “demlenir”-”demlemek”) could merge if word normalisation such as stemming or lemmatisation were applied. In this case, different words could make it into the list and recall performance would possibly increase.

Table 5.3. Most similar words to “çay” (tea)

Rank	Word	Translation	Similarity score
1	“kahve”	coffee	0.716
2	“demlenir”	~ brewed	0.652
3	“meyve”	fuit	0.650
4	“şeker”	sugar	0.636
5	“gazozdan”	~ soda	0.622
6	“preslenmesiyle”	~ with its being pressed	0.621
7	“içecekler”	drinks	0.621
8	“demlemek”	to brew	0.621
9	“çayın”	of tea	0.620
10	“qishr”	~ hot drink	0.612

Table 5.4 lists the most similar words for “çocuk” (child). In this sample, all reported words are relevant. Better precision in this sample may be linked to the higher frequency of the word in corpus. The word “çocuk” is present in 0.7% of all sentences in the training corpus whereas “çay” (tea) from the previous example is present in 0.1%. Again, 4 out of 10 words are inflections of the same lemma.

Table 5.4. Most similar words to “çocuk” (child)

Rank	Word	Translation	Similarity score
1	“çocuğun”	of the child	0.737
2	“kadın”	woman	0.700
3	“çocukların”	of children	0.695
4	“bebek”	baby	0.665
5	“kız”	girl	0.665
6	“anne”	mother	0.642
7	“çocuklar”	children	0.639
8	“çocukları”	~ children	0.630
9	“erkek”	man	0.622
10	“babası”	father of	0.614

Table 5.5 lists the most similar words for a less frequent word, “zürafa” (giraffe), which is present only in 0.002% of all training sentences. Interestingly, all reported words are related to the constellation sense of the word rather than the animal. We reason that this exposes a bias that originates from the training corpus. The word “giraffe” is not very frequently seen in news articles. However, each constellation article in Wikipedia includes a list of links to other constellation articles, which raises their frequency of being seen together. Consequently, the vectors related to the constellation sense of the word are found closer than the

animal sense. This sample also demonstrates word embeddings may fuse several senses of a word together; and more importantly, it might not be the most common sense of the word that is dominant in the embedding.

Table 5.5. Most similar words to ‘zürafa’ (giraffe)

Rank	Word	Translation	Similarity score
1	“küçüksuyılanı”	hydrus	0.671
2	“suyılanı”	hydra	0.653
3	“tekboynuz”	monoceros	0.650
4	“tilkicik”	vulpecula	0.648
5	“güneytacı”	corona australis	0.644
6	“vaşak”	lynx	0.642
7	“kanatlıat”	pegasus	0.639
8	“kılıçbalığı”	dorado	0.627
9	“tukan”	tucana	0.626
10	“camelopardalis”	camelopardalis	0.615

5.2.2 Evaluation of Linguistic Regularities

Another way to do an intrinsic evaluation of word embeddings is to test if they support “linguistic regularities” as demonstrated in (Mikolov et al., 2013c). In this section, we first describe the linguistic regularities and the “vector offset” method described in the same work. Then we test if our embeddings support such regularities in Turkish. Next, we use the Turkish question set from (Şen and Erdoğan 2014) to perform a more detailed evaluation of our embeddings and present the results.

The idea is first to find an analogy relation, such as singular/plural. Next, vector arithmetic is used to find out if the relationship between the vectors of two words is repeated among other couples of words. For Turkish, we use literal translations of English words as well as other Turkish words relevant for demonstration.

As a start, we check the singular/plural relation. Using a direct translation of the examples from (Mikolov et al., 2013c), this approximately holds. In the first equation below, the right most term ($x_{arabalar}$) is predicted as the 3rd most similar word to the result of vector arithmetic; and in the second, “aileler” is predicted as the 7th most similar word.

$$x_{elma} - x_{elmalar} \approx x_{araba} - x_{arabalar}$$

$$x_{elma} - x_{elmalar} \approx x_{aile} - x_{aileler}$$

We also tested our word embeddings against the Turkish syntactic and semantic analogy question sets prepared as part of (Sen and Erdogan 2014), which is also discussed in the Literature Review chapter. The question sets include a total of 26588 questions in several categories such as family relationships, capital cities and currencies. Each question is composed of four words where the last word is the expected answer with respect to the analogical relation between the first three words. The answer is expected to be found by linear vector arithmetic. In this study, question sets were prepared to test the linear algebraic relations between word vectors that are observed in (Mikolov et al., 2013a).

We only covered top-10 closest word tests because this study is primarily focused on paraphrase identification rather than a full intrinsic evaluation of word embeddings. Overall, Negative Sampling algorithm obtained better results but only with a slight margin. Although there are some differences for individual question sets, it can be argued that Hierarchical softmax algorithm performed better in semantic sets and Negative sampling in syntactic tests. Table 5.6 displays a break down of the results per individual question set.

In comparison to (Sen and Erdogan, 2014), there is a significant difference in the Negative Sampling algorithm results in favour of the compared work (53.43% vs. 63.03%). Hierarchical Softmax results are comparable in total (52.15% vs. 53.11%), with our study having better performance in semantic questions and the previous study in syntactic questions. Overall, the higher accuracy obtained in the previous study may be owing to the training corpus size (530 million vs. 64 million words in our study) as well as the several pre-processing steps such as stemming and independent representation of suffixes.

Table 5.6. Performance against the test data set in (Sen and Erdogan, 2014)

		Hierarchical Softmax		Negative Sampling	
Question Set	Questions	Top-10	%	Top-10	%
Semantic Tests	15902	9222	**57.99**	8971	56.41
Family relationships	132	31	**23.48**	30	22.72
Capital cities	2970	2377	**80.03**	2173	73.16
Synonyms	3422	902	**26.36**	812	23.73
Locations	6466	4911	75.95	5012	**77.51**
Currencies	156	27	17.30	45	**28.84**
Antonyms	2756	974	**35.34**	899	32.62
Syntactic Tests	10686	4644	43.46	5236	**49.00**
Plurals	4830	2031	42.05	2320	**48.03**
Negatives	756	250	33.06	350	**46.30**
Past tense	3540	1485	41.94	1707	**48.22**
Simple present tense	1560	878	**56.28**	859	55.06
Total	26588	13866	52.15	14207	**53.43**

Text sources are similar in type as both studies rely on Wikipedia and news sources. Vocabularies seem to be compatible despite the large difference between training corpus sizes. Only two words encountered in the original question set were out of the vocabulary of our embeddings: “o” (Turkish third person pronoun), which was discarded as it was shorter than 2 characters; and the verb “gerekmek” (to require), which never occurred in the infinitive form in our training corpus. Another word (“brazilya”) in the question set was interpreted as a typing error and fixed before use. Similar to the previous study, a dimensionality of 300 provided the best results in our tests.

5.2.3 Extrinsic Evaluation

For extrinsic evaluation, we tested different word embeddings on the paraphrase identification task. Certain training parameters had a noticeable impact on the task performance. First, a dimension size of 300 provided the best results in our test environment. Secondly, increasing the number of training iterations also increased the performance. Finally, negative sampling algorithm proved better than hierarchical softmax. Details are provided in the following sections. Although word embeddings is an important part of this study, optimising word embeddings deserves a study on its own. Therefore, we stopped experimenting with different training parameters at a point where the returns were diminishing.

5.3 Sentence Similarity Measures

Sentence similarity measures used in this study can be compared in several aspects. First, they have characteristic output ranges and consequently different “best” thresholds that determine the paraphrase relation. The thresholds are also dependent on the embeddings.

AWE (average of word embeddings) and CWE (concatenation of word embeddings) produce distance scores in the unit range [0, 1] while WMD (word mover’s distance) produces larger values roughly in the range [0, 30]. Best threshold dependent on this range are such as 0.192 for AWE, 0.700 for CWE and 10.418 for WMD. More details are provided in the following paragraphs. Second; repetition of words within a sentence impacts WMD and CWE scores, but not AWE scores. This is because the sentence vectors AWE creates are always of the same size, which can also be explained by AWE being a bag-of-words approach while the other two methods take word order into account. Indeed, WMD and CWE report significant distance where only the order of the words change between sentences. Third; with regard to runtime efficiency, AWE and CWE have roughly the same performance while WMD takes 5 times longer to complete the same task due to the complex calculations it carries out.

To give an overall idea about the distance scores and thresholds, Table 5.7 (on the next page) displays the average distance scores calculated over the combined corpus (PARDER and TuPC). Column “Avg P” stands for the average distance for paraphrase pairs, column “Avg NP” for non-paraphrase pairs and “All avg” column for all pairs. Threshold represents the limit below which a distance score is predicted as paraphrase. It is calculated as the average of threshold values that give the best F-score in training set in k-fold cross validation. The table verifies that average distance of non-paraphrase pairs is greater than that of paraphrase pairs. Another observation is that thresholds are quite close to the average distance of non-paraphrase pairs.

Table 5.7. Distance metrics of sentence similarity measures

Measure	Avg P	Avg NP	All avg	Threshold
AWE	0.119	0.193	0.146	0.192
CWE	0.642	0.701	0.663	0.700
WMD	6.902	10.481	8.223	10.418

Average distances given above present an overall picture however they do not give an idea about the distribution of distances over the set of sentence pairs. Histograms below (Fig. 5.1-5.6) illustrate the distribution of distance scores calculated by each measure on the combined corpus. Left-skewed histograms represent lower distance scores and right-skewed histograms higher distance scores. A noticeable shift to the right is observed for non-paraphrase pairs for AWE (Fig. 5.1, 5.2) and WMD (Fig. 5.3, 5.4) methods which produce relatively better results. In contrast, CWE histograms (Fig. 5.5, 5.6) are right-skewed and the distribution profile does not change much between paraphrase and non-paraphrase sentence pairs, which indicates a lesser classification capability.

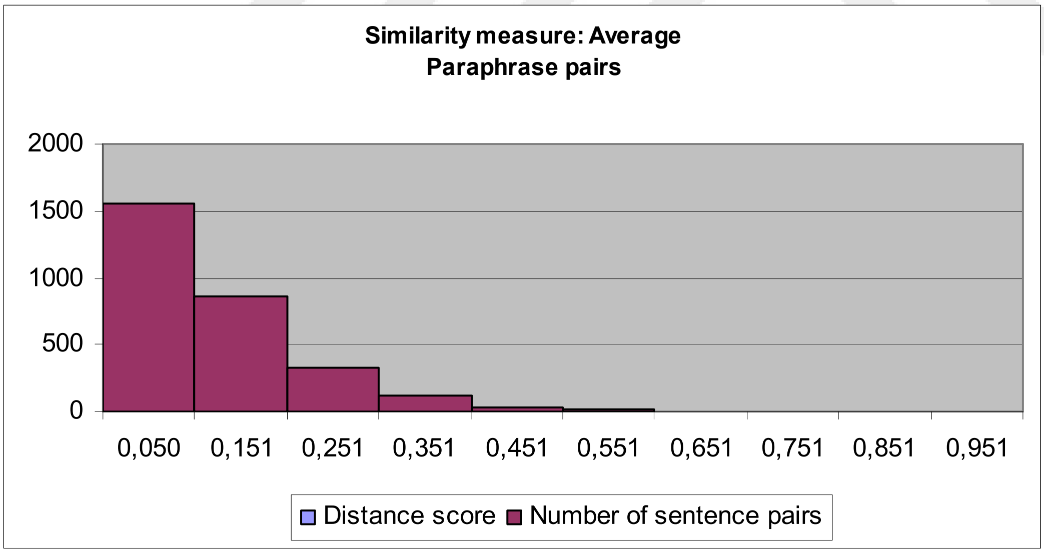

Figure 5.1. AWE measure score histogram for paraphrase sentence pairs

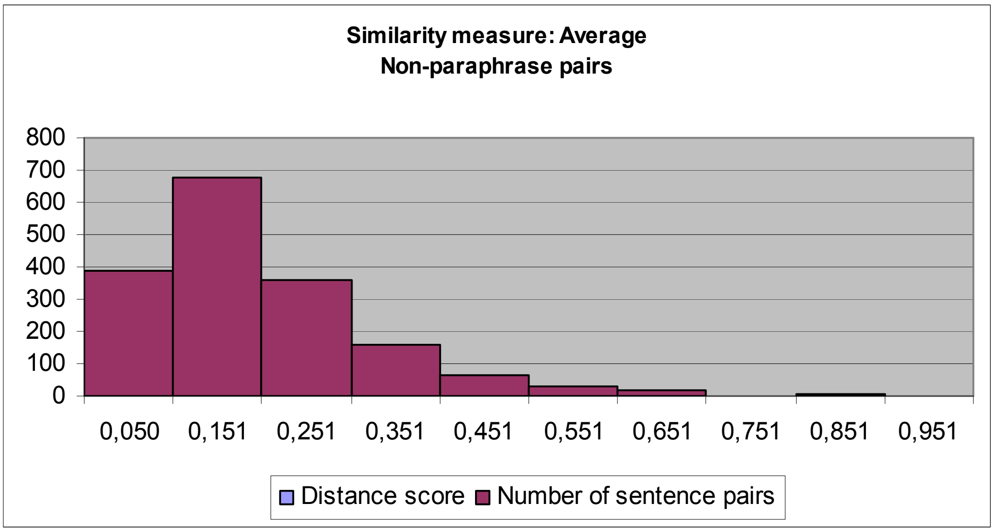

Figure 5.2. AWE measure score histogram for non-paraphrase sentence pairs

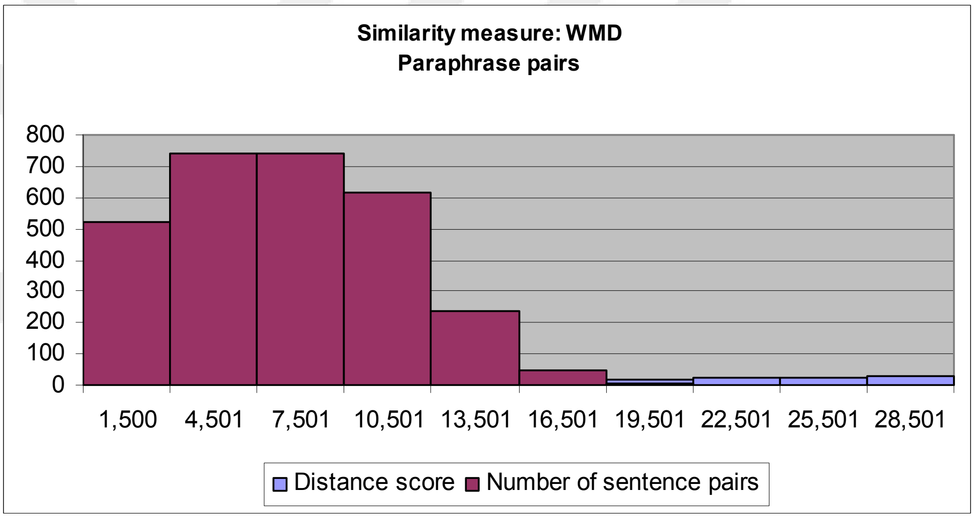

Figure 5.3. WMD measure score histogram for paraphrase sentence pairs

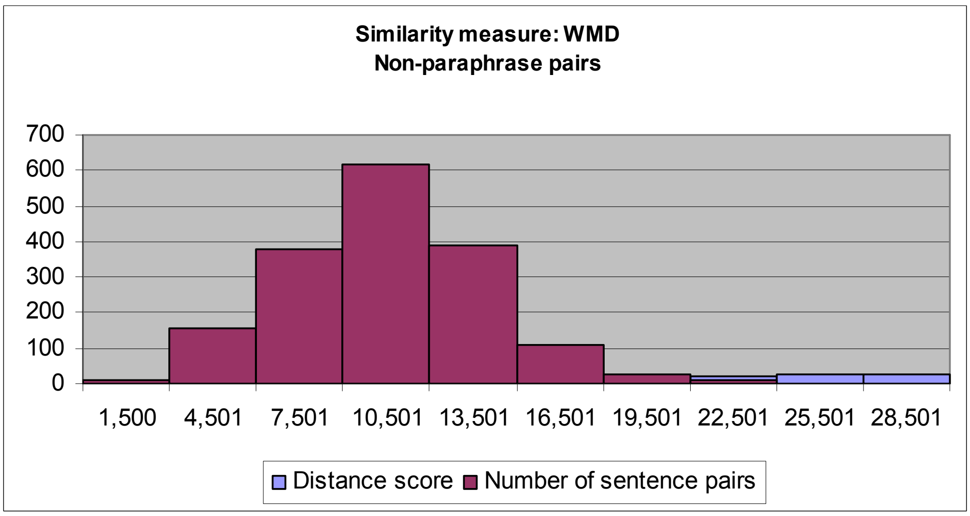

Figure 5.4. WMD measure score histogram for non-paraphrase sentence pairs

In CWE method, the shorter sentence of a pair is "padded" with a neutral word so as to obtain sentence vectors of equal size. We choose the most frequent word in the language as the padding word so it does not interfere with the meaning of the sentence. In a sentence where many words are OOV, the sentence vector might move away to an inaccurate direction. We suspect this may be the reason CWE performs below other methods. Table 5.5 and 5.6 show the histograms for CWE:

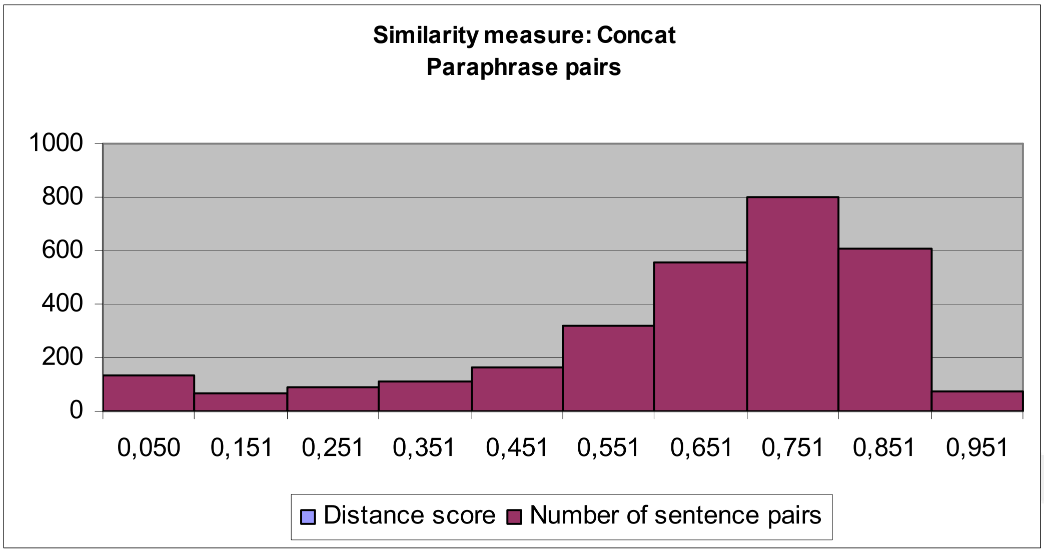

Figure 5.5. CWE measure score histogram for paraphrase sentence pairs

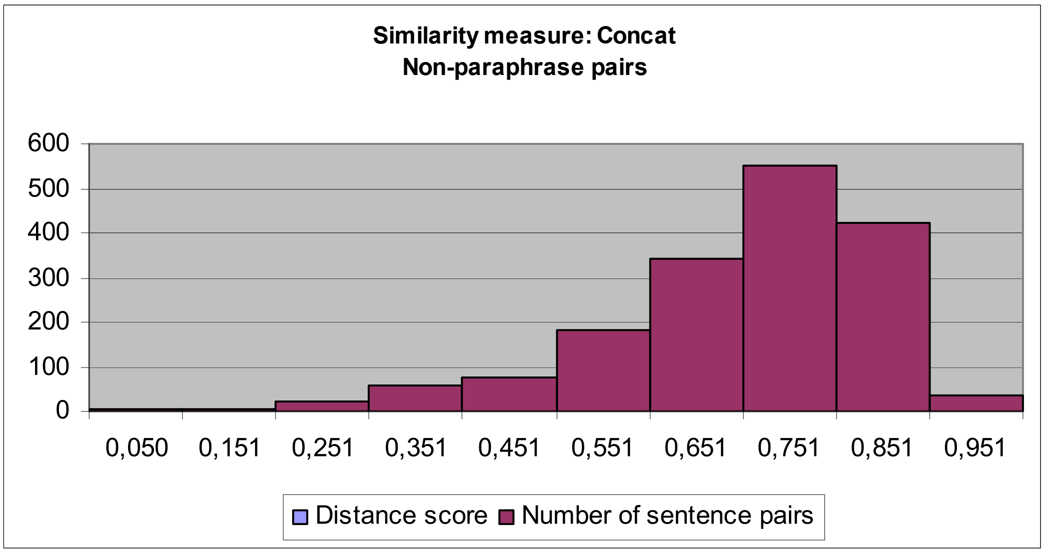

Figure 5.6. CWE measure score histogram for non-paraphrase sentence pairs

Tables 5.8-5.11 below show sample sentence pairs from PARDER and the labels predicted by WMD distance ranging between 0 to roughly 30.000. The threshold is 10.418 and distance scores below this threshold are marked as paraphrase. Sentence pairs are grouped by their respective categories in the confusion matrix (TP, TN, FP, FN). Labels are abbreviated as P for paraphrase and NP for non-paraphrase. Actual label is taken from the corpus while predicted label is decided by comparing distance to threshold.

Following tables show True Positive instances (hits). The sample in Table 5.8 is labelled correctly despite the difference in sentence lengths. There are two common words ("hakkını", "birbirimizin") which should have a big role in making the sentence vectors similar. Indeed, if we remove these words from the second sentence in our experiments, the distance is calculated noticeably higher.

Table 5.8. TP sample 1

Sentence1	"Birbirimizin hakkını hukukunu bileceğiz." We shall respect each others' rights.
Sentence2	"Birbirimizi inkar ederek, birbirimizin hakkını yok sayarak bir yere varmamız mümkün değildir." We cannot possibly get anywhere by denying or ignoring each others' rights.
Label	P
Prediction	P (8.315)

Table 5.9. TP sample 2

Sentence1	"Ricky Clemons'un kısa, problemli Missouri basketbol kariyeri sona erdi." Ricky Clemons' brief and troubled basketball career in Missouri ended.
Sentence2	"Missouri Ricky Clemons'u takımdan kovarak oradaki problemli kariyerini sona erdirdi." Missouri kicked Ricky Clemons off the team, ending his troubled career.
Label	P
Prediction	P (6.094)

Following tables show False Negative instances (misses). Table 5.10 contains two very short sentences. Calculated distance is high although it is simple for a native Turkish speaker to label it as paraphrase. It is also surprising because there are several overlapping words. An explanation could be that, since the sentence is short the distance score may be impacted by even a small number of non-overlapping words in a sufficiently distant location in the vector space. Table 5.11 combines a very long sentence with a very short one. In this case, with the same rationale mentioned above, the longer sentence has a higher probability to contain words located in a distant position in the vector space to the shorter sentence.

Table 5.10. FN sample 1

Sentence1	“Ama bunun ayda yüzde 2 faizi var.” But this has a 2% interest rate per month.
Sentence2	“Tabii bir de yüzde 2'lik faiz var.” Of course, there is also the interest rate of 2%.
Label	P
Prediction	NP (12.288)

Table 5.11. FN sample 2

Sentence1	“"Londra saldırılarından sonra terörizme karşı mücadele konusuna yaklaşımımızı tekrar gözden geçirmemiz ve yeni önlemler almamız gerekiyor" diyen Barroso, sınır polisleri arasında işbirliği, silah ve patlayıcı maddelere ilişkin ulusal yasaların uyumlu hale getirilmesi, telekomünikasyon verilerinin stoklanması ve paylaşımı, uluslararası somut işbirliği sağlanması, terör örgütlerinin mali olanaklarının kontrol edilmesi, istihbarat alışverişinin kordinasyonu gibi alanlarda yeni uygulama önerileri getireceklerini kaydetti.” ‘In the wake of the explosions in London, we need to review our approach to counter-terrorism and take new precautions’, Barroso said, stating that they will make new proposals in areas such as cross border police co-operation, improving legislation on tracing, stocking and marketing explosives, retention of telecoms data, establishing concrete international co-operation, combating terrorist financing, coordination of intelligence exchange.
Sentence2	“AB Komisyonu, Londra'da düzenlenen saldırılardan sonra terörizme karşı mücadele alanında yeni önlem önerileri getireceğini açıkladı.” In the wake of the explosions in London, the European Commission will propose a new mechanism to counter terrorist attacks.
Label	P
Prediction	NP (11.546)

5.4 Evaluation Measures

For evaluation of our approach to the PI task, we adopt accuracy and f-measure which are based on the confusion matrix elements as can be seen in Table 5.12:

Table 5.12. Confusion matrix

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Actual class</td></tr>
<tr><td>Positive</td><td>Negative</td></tr>
<tr><td rowspan="2">Predicted class</td><td>Positive</td><td>TP
True Positive</td><td>FP
False
Positive</td></tr>
<tr><td>Negative</td><td>FN
False
Negative</td><td>TN
True
Negative</td></tr>
</table>

In this table, True Positive represents hits, or the number of positive instances predicted as positive. False Positive represents false alarms, being the number of negative instances predicted as positive. False Negative represents misses, or positive instances predicted as negative. Finally, True Negative represents negative instances correctly predicted as negative, in other words correct rejections. Based on these elements, accuracy is a measure of accurate predictions over the whole set of predictions.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

F-measure, also called F-score or F1 score, is calculated as the harmonic mean of precision and recall. Precision is defined as the ratio of correct predictions among all positive predictions. Recall is defined as the ratio of correct predictions over all possible positive instances.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f\text{-}measure = 2 * \frac{precision * recall}{precision + recall}$$

5.5 Results

In our tests, the ratio of OOV words is calculated as 14% with respect to the target paraphrase corpora when “minimum count” parameter is set to 10, and drops to approximately 6% when minimum count is 2. The embeddings with lower OOV ratios achieved better results in our tests as discussed in the following chapter. For AWE (average of word embeddings) and CWE (concatenation of word embeddings), ignoring OOV words gave better results than appending the most frequent word of the language. Similarly, WMD (word mover’s distance) implementation in Gensim library removes OOV words before the computation.

The best experiment results we obtained are listed in Table 5.13 and 5.14 for Hierarchical Softmax and Negative Sampling algorithms respectively. The embeddings for this experiment were trained using Wikipedia and Bilcol-2005 corpora with 300 dimensions and 2 as minimum count in 5 iterations. Best results are highlighted in bold.

Table 5.13. F-scores of sentence distance measures (Hierarchical Softmax)

Measure	PARDER	TuPC	Combined
AWE	0.761	0.763	0.763
CWE	0.651	0.572	0.568
WMD	**0.802**	**0.816**	**0.804**

Table 5.14. F-scores of sentence distance measures (Negative Sampling)

Measure	PARDER	TuPC	Combined
AWE	0.769	0.780	0.771
CWE	0.583	0.549	0.579
WMD	**0.815**	**0.813**	**0.816**

Best performing methods use WMD sentence similarity measure and Negative Sampling algorithm. It is worth noting that AWE and WMD have comparable results in each setup, while CWE scores significantly lower in all setups. The results are consistent across different corpora and languages.

The score of WMD seems to be slightly higher for the combined corpus than an average of PARDER and TuPC, which is counter-intuitive. This can be explained by different accuracy and precision values and the F1 score calculation being a harmonic mean. Table 5.15 breaks down WMD F-scores in Table 5.13 into their components. It can be seen that the precision, recall and accuracy figures for the combined corpus are determined correctly between individual corpus figures for PARDER and TuPC:

Table 5.15. Break down of F-scores for WMD (Negative Sampling)

Paraphrase Corpus	P	R	A	F1
PARDER	0.790	0.844	0.742	0.815
TuPC	0.804	0.830	0.748	0.813
Combined	0.796	0.838	0.744	0.816

Additionally, a test was done for English paraphrase identification in the initial experimentation phase to verify that our method provides consistent results across languages. In this test, word2vec pre-trained "GoogleNews" embeddings was used against MSRP corpus (Dolan et al., 2004). Although later scores in Turkish are higher, the test served its purpose by reporting comparable results (AWE:0.798, CWE:0.541, WMD: 0.784).

Table 5.16 illustrates the impact of using different embeddings we created. It refers to results previously obtained with Hierarchical Softmax algorithm which are consistent with latest results. For each embedding, 5 training iterations were done, and Turkish combined corpus (PARDER and TuPC) was used for evaluation. A previous version of PARDER was used.

Table 5.16. F-scores of sentence distance measures

		AWE		CWE		WMD	
Dimensions	Min. Count	F1	A	F1	A	F1	A
200	2	**0.784**	0.693	**0.621**	0.576	**0.800**	**0.738**
400	2	0.781	0.691	0.609	0.575	0.799	0.736
300	2	0.781	0.691	0.616	0.578	**0.800**	**0.738**
300	10	0.783	**0.696**	0.601	0.569	0.799	0.737

It is interesting to note that WMD results are very close across various embeddings. This may suggest that, even if the dimensionality changes, the total travel distance calculated by WMD is not significantly affected. Embeddings with 200 dimensions achieved the best F-scores overall even if the differences may be too small to be meaningful. Lastly, it can be seen that lower Minimum Count obtained slightly better values. A possible explanation for this may be Turkish, being an agglutinative language, benefits from a larger vocabulary that includes many rare words.

6. CONCLUSION

In our research, we evaluated three different sentence similarity measures based on word embeddings for Turkish paraphrase identification. Revisiting the research question "*How well can sentence similarity measures based on word embeddings predict paraphrase relation in Turkish?*", Word Mover's Distance method achieved the highest score with 81.6% for Turkish paraphrase corpora we tested. This is an improvement on 74.0% of (Karaoğlan et al., 2018) but behind 83.7% obtained by the character bigram method in (Eyecioglu Ozmutlu, 2016). State-of-the-art for English is, to our knowledge, is 85.9% (Ji and Eisenstein, 2013). Using stop words as recommended for WMD did not bring any improvements. Averaging word vectors also achieved acceptable results. Given its speed in processing, it can be a promising method. Concatenation of word vectors performed visibly lower than other two measures.

We also created several word embeddings using different training parameters in the process. Negative Sampling algorithm proved to be more successful than Hierarchical Softmax as found out in (Sen and Erdogan, 2014). Best results were obtained with a dimensionality of 300. Increasing iterations (epochs) proved to have a significant positive effect at least up to 5 iterations. This also increases the training time substantially, therefore larger values was not tested. The "minimum count" parameter effectively governs the vocabulary size of the model. For the paraphrase identification task, a minimum count of 2 had noticeably better performance than a value of 10. When the vocabulary is inspected visually, it can be seen that there are many common words that are rarely seen in the training corpus. As expected due to do morphologically rich nature of Turkish, a large vocabulary is needed for good performance.

For intrinsic evaluation, we reused the analogy questions sets prepared by (Sen and Erdogan, 2014). Our embeddings obtained lower scores in the analogy task but better scores in the paraphrase identification task in comparison to the embeddings created in this work. Betters results obtained by the target study in analogy task may be owing to its representing suffixes as separate vectors. The size of training corpus could be another factor, but this was ruled out in our further tests with embeddings trained on larger corpora. Better results obtained by our study in the paraphrase identification task may indicate that analogy task may not be sufficiently representative of the real task. Also considering that testing on the paraphrase identification task is simple and fast, it can be argued that analogy tasks did not provide much value except giving visibility into the embeddings. On the other hand, inspecting most similar words to individual words was beneficial despite being a manual activity. Finding anomalous words at this stage usually triggered an inspection of the whole model vocabulary and sometimes pointed to errors or improvements in pre-processing stage.

Our approach is promising as we obtained a satisfactory result using only the sentence distance based on word embeddings. There may be many directions to pursue as future work. As a start, other sentence similarity measures may be explored and tested. A machine learning classifier can be trained using other measures such as string similarity or lexical similarity as well as sentence distance. As word embeddings are still very popular, new methods to obtain higher quality word embeddings may be found. Phrase compositionality approach (Mikolov et al., 2013b) can be tested to increase embeddings quality. Morphologically-rich nature of Turkish can be exploited by learning embeddings from morphemes instead of words. Sentence embedding models working well in low-resource languages can also be a good option.

In conclusion, word embeddings are found to be a successful device for Turkish paraphrase identification. As more and more data becomes available for statistical methods, computational methods also develop in a fast pace to handle this abundance. We hope to have made a small contribution for Turkish to keep up with this pace.

REFERENCES

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A., 2012, SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity, 9p.

Androutsopoulos, I., and Malakasiotis, P., 2010, A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Artificial Intelligence Research* 38 (May): 135–87. https://doi.org/10.1613/jair.2985.

Attardi, G., 2019, *WikiExtractor*. https://github.com/attardi/wikiextractor. (Accessed 27 November 2019)

Baroni, M., Dinu, G., and Kruszewski, G., 2014, Don't Count, Predict! A Systematic Comparison of Context-Counting vs. Context-Predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 238–47. Baltimore, Maryland: Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1023.

Bartlett, B., 2007, Finding Paraphrases Using PNrule. Ottawa: Library and Archives Canada = Bibliothèque et Archives Canada.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C., 2003, A Neural Probabilistic Language Model, 19p.

Bhagat, R., and Hovy, E., 2013, What Is a Paraphrase? *Computational Linguistics* 39 (3): 463–72. https://doi.org/10.1162/COLI_a_00166.

Bird, S., Klein, E., and Loper, E., 2009, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T., 2017, Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (December): 135–46. https://doi.org/10.1162/tacl_a_00051.

Can, Ç., Koşaner, Ö., and Aktaş, Ö., 2016, A Survey to Text Summarization Methods for Turkish. *International Journal of Computer Applications* 144 (6): 23–28. https://doi.org/10.5120/ijca2016910358.

Can, F., Kocberber, S., Baglioglu, O., Kardas, S., Ocalan, H.C., and Uyar, E., 2010, New Event Detection and Topic Tracking in Turkish. *Journal of the American Society for Information Science and Technology*, n/a-n/ap. https://doi.org/10.1002/asi.21264.

Chandler, D., 1995, *"Texts and the construction of meaning"*. http://visual-memory.co.uk/daniel/Documents/short/texts.html. (Accessed 27 November 2019)

Chapman, S., and Routledge, C., 2005, *Key Thinkers in Linguistics and the Philosophy of Language*. Oxford University Press on Demand.

Collins Online Dictionary, *"Paraphrase."*. https://www.collinsdictionary.com/dictionary/english/paraphrase. (Accessed 27 November 2019)

REFERENCES (continued)

Collobert, R., and Weston, J., 2008, A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In , 160–67. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011, Natural Language Processing (Almost) from Scratch. *NATURAL LANGUAGE PROCESSING*, 45p.

Dagan, I., and Glickman, O., 2004, Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability. *Learning Methods for Text Understanding and Mining* 2004: 26–29.

Demir, H., and Ozgur, A., 2014, Improving Named Entity Recognition for Morphologically Rich Languages Using Word Embeddings. In *2014 13th International Conference on Machine Learning and Applications*, 117–22. Detroit, MI, USA: IEEE. https://doi.org/10.1109/ICMLA.2014.24.

Demir, S., 2012, Turkish Paraphrase Corpus, 5p.

Dictionary.com Online Dictionary, *"Paraphrase."*. https://www.dictionary.com/browse/paraphrase. (Accessed 27 November 2019)

Dolan, B., Quirk, C., and Brockett, C., 2004, Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources, 7p.

Eyecioglu Ozmutlu, A., 2016, Paraphrase Identification Using Knowledge-Lean Techniques, 173p.

Finch, A., Hwang, Y.-S., and Sumita, E., 2005, Using Machine Translation Evaluation Techniques to Determine Sentence-Level Semantic Equivalence, 8p.

Firth, J.R., 1957, A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*.

Fleiss, J.L., 1971, Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin* 76 (5): 378.

Goldberg, Y., 2016, A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research* 57 (November): 345–420. https://doi.org/10.1613/jair.4992.

Gomaa, W.H., and Fahmy, A.A., 2013, A Survey of Text Similarity Approaches. *International Journal of Computer Applications* 68 (13): 13–18.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T., 2018, Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Gungor, O., and Yildiz, E., 2017, Linguistic features in Turkish word representations. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, 1–4. Antalya, Turkey: IEEE. https://doi.org/10.1109/SIU.2017.7960223.

REFERENCES (continued)

Guo, W., and Diab, M., 2012, Modeling Sentences in the Latent Space, 9p.

Hofstadter, D.R., 2000, *Gödel, Escher, Bach*. Penguin.

Huang, G., Guo, C., Kusner, M.J., Sun, Y., Sha, F., and Weinberger, K.Q., 2016, Supervised Word Mover's Distance. In *Advances in Neural Information Processing Systems*, 4862–4870.

I. Celenli, H., Talha Ozturk, S., Sahin, G., Gerek, A., and C. Ganiz, M., 2018, Document Embedding Based Supervised Methods for Turkish Text Classification. In , 477–82. https://doi.org/10.1109/UBMK.2018.8566326.

Islam, A., and Inkpen, D., 2008, Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity. *ACM Transactions on Knowledge Discovery from Data* 2 (2): 1–25. https://doi.org/10.1145/1376815.1376819.

Issa, F., Damonte, M., Cohen, S.B., Yan, X., and Chang, Y., 2018, Abstract Meaning Representation for Paraphrase Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 442–52. New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1041.

Ji, Y., and Eisenstein, J., 2013, Discriminative Improvements to Distributional Sentence Similarity, 6p.

Jing, K., and Xu, J., 2019, A Survey on Neural Network Language Models. *ArXiv:1906.03591 [Cs]*, June. http://arxiv.org/abs/1906.03591.

Karaoglan, B., Kışla, T., and Metin, S.K., 2016, Description of Turkish Paraphrase Corpus Structure and Generation Method. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 208–217. Springer.

Karaoğlan, B., Kışla, T., and Kumova Metin, S., 2018, Contribution of syntactic and semantic attributes in paraphrase identification. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 1–4. Izmir: IEEE. https://doi.org/10.1109/SIU.2018.8404204.

Karaoğlan, B., Yorgancioğlu, H.E., Kişla, T., and Metın, S.K., 2019, The Impact of Sentence Embeddings in Turkish Paraphrase Detection. In , 1–4. IEEE.

Kusner, M.J., Sun, Y., Kolkin, N.I., and Weinberger, K.Q., 2015, From Word Embeddings To Document Distances, 10p.

Le, Q., and Mikolov, T., 2014, Distributed Representations of Sentences and Documents, 9p.

Levy, O., Goldberg, Y., and Dagan, I., 2015, Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics* 3 (December): 211–25. https://doi.org/10.1162/tacl_a_00134.

Maaten, L. van der, and Hinton, G., 2008, Visualizing Data Using T-SNE. *Journal of Machine Learning Research* 9 (Nov): 2579–2605.

REFERENCES (continued)

Manning, C.D., and Schiitze, H., 1999, Foundations of Statistical Natural Language Processing, 704p.

McInnes, L., Healy, J., and Melville, J., 2018, Umap: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv Preprint ArXiv:1802.03426*.

Mihalcea, R., 2006, Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity, 6p.

Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013a, Efficient Estimation of Word Representations in Vector Space. *ArXiv:1301.3781 [Cs]*, January. http://arxiv.org/abs/1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J., 2013b, Distributed Representations of Words and Phrases and Their Compositionality, 9p.

Mikolov, T., Yih, W., and Zweig, G., 2013c, Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751.

Mitchell, J., and Lapata, M., 2010, Composition in Distributional Models of Semantics. *Cognitive Science* 34 (8): 1388–1429. https://doi.org/10.1111/j.1551-6709.2010.01106.x.

Nadkarni, P.M., Ohno-Machado, L., and Chapman, W.W., 2011, Natural Language Processing: An Introduction. *Journal of the American Medical Informatics Association* 18 (5): 544–51. https://doi.org/10.1136/amiajnl-2011-000464.

Pembeci, İ., 2016, Using Word Embeddings for Ontology Enrichment. *International Journal of Intelligent Systems and Applications in Engineering* 4 (3): 49. https://doi.org/10.18201/ijisae.58806.

Pennington, J., Socher, R., and Manning, C., 2014, Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–43. Doha, Qatar: Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162.

Řehůřek, R., and Sojka, P., 2010, Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., and others, 1988, Learning Representations by Back-Propagating Errors. *Cognitive Modeling* 5 (3): 1.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T., 2015, Evaluation Methods for Unsupervised Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 298–307. Lisbon, Portugal: Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1036.

REFERENCES (continued)

Sen, M.U., and Erdogan, H., 2014, Learning word representations for Turkish. In *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, 1742–45. Trabzon, Turkey: IEEE. https://doi.org/10.1109/SIU.2014.6830586.

The American Heritage Dictionary, *"Paraphrase."*. https://ahdictionary.com/word/search.html?q=paraphrase. (Accessed 27 November 2019)

Turney, P.D., and Pantel, P., 2010, From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research* 37 (February): 141–88. https://doi.org/10.1613/jair.2934.

Ul-Qayyum, Z., and Altaf, W., 2012, Paraphrase Identification Using Semantic Heuristic Features, 12p.

ACKNOWLEDGEMENT

(In Turkish)

Yüksek lisans eğitimimin ilk gününden itibaren bana sabırla destek olan, tez çalışmalarımda deneyimi ve bilgi birikimiyle kılavuzluk eden kıymetli tez danışmanım Prof. Dr. Bahar KARAOĞLAN'a teşekkürü borç sayıyor, saygılarımı sunuyorum.

Tez süreci boyunca anlayışları, değerli geribildirimleri ve tez çalışmasına temel teşkil eden makalenin hazırlanmasındaki büyük katkıları için Doç. Dr. Senem KUMOVA METİN ve Doç. Dr. Tarık KIŞLA'ya içten teşekkürlerimi sunuyorum.

Değerli Bilcol-2005 derlemi için kullanım izni veren Prof. Dr. Fazlı CAN'a teşekkür ederim.

Yüksek lisans eğitimimi destekleyen işyerim Delphi Technologies ve yöneticilerim Gülşah MURAT ve Işıl YAPALI'ya teşekkür ediyorum.

Ve tüm bu uzun süren uğraşlarımda desteğini hiç esirgemeyen sevgili annem, babam ve kardeşime, kendi yüksek lisansını sürdürürken bile daima yanımda olan sevgili eşim Nilay'a, son olarak da neşe kaynağım biricik Elif'ime yürekten sevgilerimi sunuyorum.

.... / .... / 2019

İmza

Hakkı Engin YORGANCIOĞLU

CURRICULUM VITAE

Hakkı Engin YORGANCIĞLU

email : henginy@alumni.bilkent.edu.tr

Personal Information

Nationality:	Turkish
Place of Birth:	Denizli, Turkey
Date of Birth:	1982

Education

2013 – 2019	Ege University, Izmir, Turkey International Computer Graduate School Information Technologies Department
2000 – 2004	Bilkent University, Ankara, Turkey Engineering Faculty Computer Science Department

Work Experience

2014 – ...	Delphi Technologies, İzmir, Turkey Team Leader
2010 – 2014	ASELSAN A.Ş., İzmir, Turkey Expert Engineer
2009 – 2010	ETİ Gıda A.Ş., Eskişehir, Turkey Software Engineer
2007 – 2008	Keytorc Technology Services, İzmir, Turkey Technology Consultant
2004 – 2005	AGMLab Information Technologies, Ankara, Turkey Software Engineer

Languages

Turkish (native)
English (advanced)
French (beginner)