INTRINSIC AND EXTRINSIC EVALUATION OF WORD EMBEDDING MODELS

by

Gökçe Yeşiltaş

B.S., Computer Engineering, Boğaziçi University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2019

INTRINSIC AND EXTRINSIC EVALUATION OF WORD EMBEDDING MODELS

APPROVED BY:

Prof. Tunga Güngör (Thesis Supervisor)	.....
Prof. Fikret Gürgen	.....
Prof. Şule Gündüz Öğüdücü	.....

DATE OF APPROVAL: 22.07.2019

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my thesis supervisor Prof. Tunga Güngör. He was very supportive to me all the time. I would like to appreciate his guidance during my research period.

I am very grateful to my family for their support and believing me in all circumstances. They have always motivated me to do better.

ABSTRACT

INTRINSIC AND EXTRINSIC EVALUATION OF WORD EMBEDDING MODELS

In natural language processing tasks, representing a word is an important issue. After Bengio et al. introduced a simple neural network language model that learns word vector representations in 2003, representing words in continuous vector space has become more popular. Mikolov et al. introduced a method named word2vec and showed that word embedding could capture meaningful syntactic and semantic similarities in 2013. Many methods and implementations have been proposed for English since then. However, there are only a few studies on word representations in Turkish. In this study, we aimed to understand and analyze how word embedding models work on both Turkish and English. We focused on the word2vec word embedding model and tried to modify it to improve the quality of word representations. Additionally, we trained many models with different window sizes and dimensions. The impact of different configurations on the quality of word representations was analyzed both intrinsically and extrinsically. We reported the accuracy on word analogy tasks for intrinsic evaluation and word similarity tasks for extrinsic evaluation. Our results show that our proposed models perform better on most of the word analogy task categories for Turkish. We also showed that increasing window sizes and dimensions does not always affect the accuracy in a positive direction. For some analogy and word similarity tasks, it affects negatively.

ÖZET

KELİME GÖMEVLERİNİN İÇSEL VE DIŞSAL DEĞERLENDİRMESİ

Bir kelimeyi matematiksel olarak temsil etmek doğal dil işleme uygulamarında önemli bir konudur. Bengio ve arkadaşlarının 2003'te basit sinir ağları kullanarak kelime vektör temsilleri elde etmelerinin ardından, kelimeleri sürekli vektör uzayında temsil etmek daha popüler hale gelmiştir. Mikolov ve arkadaşları 2013'te, word2vec adında yeni bir yöntem öne sürerek, kelime gömevlerinin sözdizimsel ve anlamsal benzerlikleri yakalayabildiğini gösterdi. O zamandan beri İngilizce için birçok yöntem geliştirildi ve uygulamalar yapıldı. Ancak, Türkçe'de kelime temsilleri üzerine yapılan sadece birkaç çalışma vardır. Bu çalışmada kelime gömevi yöntemlerinin hem Türkçe hem de İngilizce'de nasıl çalıştığını analiz etmeyi amaçladık. Word2vec kelime gömevi modeline odaklandık ve kelime temsillerinin kalitesini artırmak için bu modeli geliştirmeye çalıştık. Ek olarak, farklı pencere ve vektör boyutlarına sahip birçok model eğittik. Farklı konfigürasyonların kelime temsillerinin kalitesi üzerindeki etkisini hem içsel hem de dışsal olarak analiz ettik. İçsel değerlendirme için kelime benzeşim görevlerini ve dışsal değerlendirme için ise kelime benzerlik görevlerini kullandık. Sonuç olarak, önerilen modellerimizin Türkçe için, çoğu benzeşim kategorisinde, orijinal word2vec modeline göre daha iyi performans sergilediği gözlemlendi. Ayrıca, pencere ve vektör boyutlarının arttırılmasının, farklı benzeşim kategorilerinde farklı sonuçlar verdiğini gözlemledik. Pencere ve vektör boyutundaki artışın her zaman olumlu sonuçlanmadığını gördük. Bazı kelime benzeşim ve kelime benzerliği görevleri için pencere ve vektör boyutu arttıkça sonuçların kötüleştiğini gözlemledik.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiv
1. INTRODUCTION . . . 1
2. BACKGROUND . . . 3
2.1. Related Work in English . . . 3
2.1.1. Word2Vec . . . 3
2.1.2. GloVe . . . 5
2.1.3. fastText . . . 5
2.1.4. Morphological RNNs . . . 6
2.2. Related Work in Turkish . . . 6
3. CORPORA AND DATASETS . . . 8
3.1. Corpora . . . 8
3.1.1. Turkish Corpus . . . 8
3.1.2. English Corpus . . . 8
3.2. Analogy Tasks . . . 8
3.2.1. Turkish Analogy Task . . . 9
3.2.2. English Analogy Task . . . 9
3.3. Word Similarity Tasks . . . 11
3.3.1. Turkish Word Similarity Task . . . 11
3.3.2. English Word Similarity Tasks: WordSim353 . . . 14
3.3.3. English Word Similarity Tasks: MC . . . 14
3.3.4. English Word Similarity Tasks: RG . . . 14
3.3.5. English Word Similarity Tasks: RW . . . 15
3.3.6. English Word Similarity Tasks: SCWS . . . 15
4. METHODOLOGY AND EXPERIMENTAL RESULTS . . . 17

4.1. Methodology and Experimental Setup 17
4.2. Intrinsic Evaluation of Word Representations 21
4.2.1. Evaluation of Turkish Word Representations 22
4.2.2. Evaluation of English Word Representations 34
4.3. Extrinsic Evaluation of Word Representations 48
4.3.1. Evaluation of Turkish Word Representations 48
4.3.2. Evaluation of English Word Representations 51
5. CONCLUSIONS 65
5.1. Conclusion 65
5.2. Future Work 67
REFERENCES 68

LIST OF FIGURES

Figure 4.1. The original architecture of the Skip-gram where window size is 2. [1] 18

Figure 4.2. The modified architecture of the Skip-gram where the context window is chosen from only left side of the target word and window size is 4. 18

Figure 4.3. The modified architecture of the Skip-gram where the context window is chosen from only right side of the target word and window size is 4. 19

Figure 4.4. Windowing operation with different context orientations. 20

Figure 4.5. Accuracy (%) results on "district-city" analogy task questions. . . 31

Figure 4.6. Accuracy (%) results on "present tense" analogy task questions. . 32

Figure 4.7. Accuracy (%) results on "capital-country" analogy task questions. 33

Figure 4.8. Accuracy (%) results of the original Skip-gram models on "kinship" analogy task questions. Accuracy on this task is in a negative proportional relationship with window size and vector dimension. . 43

Figure 4.9. Accuracy (%) results of the original Skip-gram models on "city-state" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size and vector dimension. . 44

Figure 4.10. Accuracy (%) results of the original Skip-gram models on "nationality adjective" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size and vector dimension. 45

Figure 4.11. Accuracy (%) results of the original Skip-gram models on "comparative" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size but a negative relationship with vector dimension. 46

Figure 4.12. Accuracy (%) results of the original Skip-gram models on "opposite" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size but a negative relationship with vector dimension. 47

Figure 4.13. Spearman's rank correlation ($\rho \times 100$) on WordSimTR for models where vector dimension was set to 100 shown in the graphic. . . . 50

Figure 4.14. Spearman's rank correlation ($\rho \times 100$) on WordSimTR for models where vector dimension was set to 200 shown in the graphic. . . . 50

Figure 4.15. Spearman's rank correlation ($\rho \times 100$) on WordSimTR for models where vector dimension was set to 300 shown in the graphic. . . . 51

Figure 4.16. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 100 shown in the graphic. . . . 53

Figure 4.17. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 200 shown in the graphic. . . . 53

Figure 4.18. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 300 shown in the graphic. . . . 54

Figure 4.19. Spearman's rank correlation ($\rho \times 100$) on MC for models where vector dimension was set to 100 shown in the graphic. 56

Figure 4.20. Spearman's rank correlation ($\rho \times 100$) on MC for models where vector dimension was set to 200 shown in the graphic. 56

Figure 4.21. Spearman's rank correlation ($\rho \times 100$) on MC for models where vector dimension was set to 300 shown in the graphic. 57

Figure 4.22. Spearman's rank correlation ($\rho \times 100$) on RG for models where vector dimension was set to 100 shown in the graphic. 57

Figure 4.23. Spearman's rank correlation ($\rho \times 100$) on RG for models where vector dimension was set to 200 shown in the graphic. 59

Figure 4.24. Spearman's rank correlation ($\rho \times 100$) on RG for models where vector dimension was set to 300 shown in the graphic. 59

Figure 4.25. Spearman's rank correlation ($\rho \times 100$) on RW for models where vector dimension was set to 100 shown in the graphic. 61

Figure 4.26. Spearman's rank correlation ($\rho \times 100$) on RW for models where vector dimension was set to 200 shown in the graphic. 61

Figure 4.27. Spearman's rank correlation ($\rho \times 100$) on RW for models where vector dimension was set to 300 shown in the graphic. 62

Figure 4.28. Spearman's rank correlation ($\rho \times 100$) on SCWS for models where vector dimension was set to 100 shown in the graphic. 62

Figure 4.29. Spearman's rank correlation ($\rho \times 100$) on SCWS for models where vector dimension was set to 200 shown in the graphic. 64

Figure 4.30. Spearman's rank correlation ($\rho \times 100$) on SCWS for models where vector dimension was set to 300 shown in the graphic. 64

LIST OF TABLES

Table 3.1. Examples for each category in the Turkish analogy task. 10

Table 3.2. The number of questions in each category of the Turkish analogy task. 11

Table 3.3. Examples for each category in the English analogy task. 12

Table 3.4. The number of questions in each category of the English analogy task. 12

Table 3.5. Summary about word similarity datasets. 13

Table 3.6. Examples for the word similarity task *WordSimTr*. 13

Table 3.7. Examples for the word similarity task *WordSim353*. 14

Table 3.8. Examples for the word similarity task *MC*. 15

Table 3.9. Examples for the word similarity task *RG*. 15

Table 3.10. Examples for the word similarity task *RW*. 16

Table 3.11. Examples for the word similarity task *SCWS*. 16

Table 4.1. The results on the Turkish word analogy task, given as accuracy (%). 23

Table 4.2. Accuracy (%) of trained models on each category of the Turkish analogy task set. 26

Table 4.3. The results on the English word analogy task, given as accuracy (%). 34

Table 4.4. Accuracy (%) of trained models on *semantic question categories* of the English analogy task set. 36

Table 4.5. Accuracy (%) of trained models on *syntactic question categories* of the English analogy task set. 39

Table 4.6. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *WordSimTR*. 49

Table 4.7. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *WordSim353*. 52

Table 4.8. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *MC*. 55

Table 4.9. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *RG*. 58

Table 4.10. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *RW*. 60

Table 4.11. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *SCWS*. 63

LIST OF ACRONYMS/ABBREVIATIONS

CBOW	Continuous Bag-Of-Words
NLM	Neural Language Models
NLP	Natural Language Processing
POS	Part-of-Speech
RNN	Recursive Neural Network

1. INTRODUCTION

In Natural Language Processing (NLP) tasks, representing a word is an important issue. Word representations are used as inputs for NLP tasks such as classification of documents, machine translation, named entity recognition, and sentiment analysis. Representing each word as a one-hot encoded vector results in a sparse high-dimensional vector space where its dimension equals the size of the vocabulary. Word embedding is a mathematical embedding from high dimensional sparse space into a dense continuous vector space with a lower dimension. There are two main benefits of the distributed word representations: lower dimension results in a less computational cost; grouping similar words achieves a better performance in NLP tasks [2], [3], [4], [5], [6].

Rumelhart et al. worked on one of the earliest use of word representations in 1986 [7]. With technological developments and researches, distributed word representations have become more popular. Mikolov et al. proposed a method named word2vec and showed that word embedding could capture meaningful syntactic and semantic similarities in 2013 [8]. Their research showed that word vectors obtained by using the word2vec could have linear relationships. For instance, vector("queen") is the closest vector for the result of vector("king") - vector("man") + vector("woman"). Many methods and implementations have been proposed for English since then. However, there are only a few studies on word representations in Turkish.

In this study, we aim to understand and analyze how word embedding models work on Turkish, which is an agglutinative and morphologically rich language. We aim to evaluate the quality of word embedding models with intrinsic and extrinsic tasks in both Turkish and English.

We focused on the word2vec word embedding model. We tried to modify the proposed model to improve the quality of word representations. We intended to change context words orientation. In NLP tasks, changing context orientation is a commonly used approach. For instance, in Part-of-Speech (POS) tagging task, to find POS tag

of the current word, models may look only past context, only future context, or both past (left) and future (right) context [9] [10].

In the classical word2vec methodology, context words are chosen from both sides of the target word. We changed context orientation to just the left side or just the right side words of the target word. The impact on the quality of word representations was analyzed both intrinsically and extrinsically. We used word analogy tasks for intrinsic evaluation and word similarity tasks for extrinsic evaluation.

This thesis is organized as follows: Word embedding models and background research are represented in Chapter 2. Corpora that are used to train models and datasets that are used for evaluation of models are described in Chapter 3. In Chapter 4, methodology, experimental setups and evaluation results are shared. In Chapter 5, we conclude our research and discuss possible future research directions.

2. BACKGROUND

2.1. Related Work in English

Representation of words in a continuous vector space has a long history dating back to 1986 [7]. After Bengio et al. introduced a simple neural network language model that learns word vector representations in 2003 [11], representing words in continuous vector space has become more popular.

2.1.1. Word2Vec

In [8], Mikolov et al. worked on a Neural Language Model (NLM). They found out that word representations can capture syntactic regularities such as singular/plural forms of common nouns and semantic regularities such as gender relation or country-capital relations. One example for syntactic regularity that can be captured is that the word *big* is similar to *bigger* in the same sense that *small* is similar to *smaller*. Moreover, the regularities are observed as constant vector offsets between pairs of words sharing a particular relationship. To find a word that is similar to *small* in the same sense as *bigger* is similar to *big*, one can compute vector X = vector("bigger") - vector("big") + vector("small"). The word closest to X measured by cosine distance can be used as the answer to the question. One example for semantic regularities that can be captured is that vector("king") - vector("man") + vector("woman") results in a vector that is closest to the vector("queen").

In [1], regarding minimizing the computational cost of learning distributed word embedding, Mikolov et al. proposed two new architectures: Continuous Bag-Of-Words (CBOW) and Continuous Skip-gram. The CBOW model tries to predict the current word based on the context words (previous and following words in the given window size). On the other hand, the Skip-gram tries to predict words within a range before and after the current word. In other words, Skip-gram tries to predict the context words based on the current word. Mikolov et al. enlarged and used syntactic and semantic

analogy test sets provided in [8] to measure the quality of the word representations. This work is essential for not only capturing regularities among words but also being an efficient method compared to previous word embedding methods. The model can learn word vectors from 1.6 billion words data set in less than a day. The Skip-gram model does not require dense matrix multiplications in contrast to most of the neural network architectures.

In [12], Mikolov et al. worked on Skip-gram model and enhanced the original Skip-gram model to improve training time and quality of vector representations. As we mentioned previously, in Skip-gram architecture, given a sequence of words $w_1, w_2, ...w_T$, the aim is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{2.1}$$

where c is the training context size. The Skip-gram formulation defined $p(w_{t+j}|w_t)$ using the softmax function [12]:

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W}\exp({v'_w}^{\top} v_{w_I})} \tag{2.2}$$

where v_w and v'_w are input and output vector representations of w, and W is the vocabulary size. Since the computational cost of the formula is proportional to vocabulary size, the formula is impractical. Mikolov et al. introduced a computationally efficient approximation and defined Negative Sampling (NEG) by the objective [12]

$$\log\sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k}\mathbb{E}_{w_i\sim P_n(w)}[\log\sigma(-{v'_{w_O}}^{\top} v_{w_I})] \tag{2.3}$$

which was used to replace $\log p(w_O|w_I)$ term in the Skip-gram objective where $P_n(w)$ is the noise distribution. Negative sampling is an approach where each training sample is used to update only a small percentage of the model's weights instead of all weights. Negative sampling also resulted in better training time and better representations for frequent words.

Another improvement introduced for the Skip-gram model in [12] was subsampling of frequent words. The common words like "the" are not informative about the words that appear in the same context because they appear in the context of lots of words. To address the problem, they introduced a subsampling approach: each word w_i in the training set is discarded with the probability computed by the formula [12]

$$P(w_i) = 1 - \sqrt{\frac{1}{f(w_i)}} \tag{2.4}$$

where $f(w_i)$ is the frequency of word w_i and t is a chosen threshold. Mikolov et al. stated that the given subsampling formula resulted in significant speedup and improved accuracy for rare words' representations.

2.1.2. GloVe

In [13], Pennington et al. worked on a model that combines the advantages of two primary model families to learn word representations: global matrix factorization methods such as latent semantic analysis (LSA) and local context window methods such as Skip-gram. The proposed model efficiently leverages statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix like LSA or individual context windows in a large corpus like Skip-gram. They called the proposed model as GloVe, for Global Vectors, because the global corpus statistics are captured directly by the model. Word analogy tasks provided in [1] had been used to evaluate the quality of the word representations.

2.1.3. fastText

Previously mentioned models, in Subsection 2.1.1 and Subsection 2.1.2, try to learn a distinct word vector for each word. They do not take internal structures of words into account. Especially in morphologically rich languages, a word may have different forms that rarely appear in a training set. As a result, the word embedding models such as word2vec and GloVe have good representations for frequent words, such

as “distinct”, whereas worse representation for rare ones, such as “distinctiveness”. In [14], Bojanowski et al. proposed a model, called fastText, to overcome these limitations. The model was based on the Skip-gram model, where each word has been represented as a bag-of-character n-gram. The model learns representations for character n-grams. Word representations are calculated by the sum of the vector representations of its n-grams. Thus, vector representation can be calculated even for out-of-vocabulary words. They evaluated the quality of the method by testing them on analogy and similarity tasks. Results showed that morphological information significantly improves the accuracy of syntactic tasks, whereas it does not improve the accuracy of semantic tasks.

2.1.4. Morphological RNNs

In [15], Luong et al. proposed a model to use the morphological relationship among words. Their purpose was to introduce a model that may represent rare and compound words better by using morphemes. They treated each morpheme as a basic unit in a Recursive Neural Network (RNN). Representations for morphologically complex words were constructed from their morphemes. This model can build vector representations for out-of-vocabulary words by using known morphemes. They took contextual information into account by training an NLM and integrating RNN structures for compound words. The RNN was used to model the morphological structures of words, i.e., the syntactic information, to learn morphemic compositionality. The NLM was used to utilize word contexts to provide further semantics to the learned representations.

2.2. Related Work in Turkish

In [16], Şen et al. worked on Turkish word representations. They applied the Skip-gram model in Turkish and created test sets to evaluate the quality of word representations. In this work, negative sampling and subsampling the frequent words were followed as in [12]. Before training the model, words were preprocessed. Because Turkish is a morphologically rich language, stemming was performed for infrequently

used words to increase the quality of word representations. To evaluate the quality of the word representations, Şen et al. prepared word analogy tasks for both semantic and syntactic regularities in Turkish. Their analogy tasks, similar with the tasks in [8], consists of questions like "what word is similar to *olay* (event) in the same sense *kelimeler* (words) is similar to *kelime* (word)?". The syntactic test set includes four question groups testing singular/plural forms and negative forms of common nouns; base, past and third person present tense forms of verbs. The semantic test set includes six groups testing kinship, capital-country, district-city, country-currency relations, antonyms, and synonyms. Additionally, Şen et al. developed another test based on finding the word that does not belong to the group of six words. For instance, there are six countries, one of them is on a different continent, the aim is to find that country. Results showed that word representations could be useful in Turkish.

In [17], Güngör et al. aimed to explore the morphological information captured by the Turkish word representations. Skip-gram model was used to learn word representations. An analogical reasoning task was performed to evaluate the quality of information obtained between Turkish words in morphological relation with each other. They prepared question sets testing root and inflected or derivative forms of words. They analyzed the quality of word representations separately for noun and verb roots and each inflectional and derivational affix. Results showed that even without preprocessing, word representations in Turkish, such a morphologically rich language, can capture morphological information.

In [18], Üstün et al., in a recent work, claimed that using words as they are to learn vector representations result in inadequate representations for rare words because of lack of statistics. They declared that using characters or character n-grams could result in distant representations of semantically related words with different forms of the same morpheme (allomorphs). To overcome these problems and to learn better word representations, morphemes were used in this paper. They argued that using morphemes instead of characters results in more accurate word vectors, especially in morphologically complex languages, like Turkish. The proposed model learned word representations through its morphemes.

3. CORPORA AND DATASETS

3.1. Corpora

3.1.1. Turkish Corpus

We trained Turkish word embedding models on *BounWebCorpus*. The corpus was collected using news and web pages by Sak et al. [19]. The corpus contains more than four hundred million words. They have shared the preprocessed version of the corpus; numbers were written in words (for instance, "3" was turned into "üç" (three)), punctuation marks were removed, the corpus was split into sentences. We split the corpus into seven parts that have approximately the same number of sentences because there were not enough resources to train the whole corpus at one time.

3.1.2. English Corpus

We trained English word embedding models on *Wikipedia dump data*. [20] The corpus parsed by using Wikipedia Extractor that is provided by MediaLab of the University of Pisa. [21] The corpus contains more than two billion words. Due to a lack of resources to train the whole corpus at one time, we split the corpus into 28 parts that have approximately the same number of sentences.

3.2. Analogy Tasks

We used analogy tasks for intrinsic evaluation of models that we trained. We used a Turkish analogy task that is shared by Şen et al. [16]. For English models, we used the analogy task that is shared by Mikolov et al. [12]. Analogy task sets consist of statements like *"a is to b as c is to d"*. In other words, the relation between a and b is similar to the relation between c and d. Details about analogy task sets are given in the following two subsections.

3.2.1. Turkish Analogy Task

We evaluated the Turkish word embedding models on analogy task set that were created in [16]. The analogy task set contains 10 different categories. Six of them contains word pairs that have semantic relations, which are kinship, capital-country, synonyms, district-city, currency, and antonyms. Four of them consist of word pairs that have syntactic relations, which are plural, past tense, present tense, and negative present tense. Examples for each category are shown in Table 3.1. The analogy task set includes 15902 semantic and 10686 syntactic questions. The number of questions in each category is listed in Table 3.2.

We evaluated our word embedding models on:

- total accuracy
- accuracy on individual analogy task categories
- semantic accuracy and syntactic accuracy

3.2.2. English Analogy Task

We evaluated the English models on analogy tasks that were created in [8]. The analogy task set contains fourteen different categories. Five types of semantic and nine types of syntactic questions are part of the word relationship test set. Relations that are questioned in semantic tasks are kinship, common capital-country relations, all capital-country relations, state-city, and country-currency. Syntactic tasks contain the following relations; opposite, comparative, superlative, plural nouns, plural verbs, present tense, past tense, adjective-adverb, and nationality adjective. One example from each category is shown in Table 3.3. The analogy task set consists of 8869 semantic and 10675 syntactic questions. The number of questions in each category is listed in Table 3.4.

Table 3.1. Examples for each category in the Turkish analogy task.

Type of Relationship	**Word Pair 1**		**Word Pair 2**	
kinship	anne (mother)	baba (father)	kız (daughter)	oğul (son)
capital-country	abuja (Abuja)	nijerya (Nigeria)	amman (Amman)	ürdün (Jordan)
synonyms	abece (ABC)	alfabe (alphabet)	abide (monument)	anıt (memorial)
district-city	seyhan	adana	akyurt	ankara
country-currency	abd (USA)	dolar (dollar)	arjantin (Argentina)	peso (peso)
antonym	acemi (novice)	usta (master)	ak (white)	kara (black)
plural nouns	araştırma (research)	araştırmalar (researches)	arkadaş (fried)	arkadaşlar (friends)
past tense	almak (to receive)	aldı (received)	bırakmak (to leave)	bıraktı (left)
present tense	almak (to receive)	alır (receives)	getirmek (to bring)	getirir (brings)
negative present tense	açar (opens)	açmaz (doesn't open)	ağlar (cries)	ağlamaz (doesn't cry)

Table 3.2. The number of questions in each category of the Turkish analogy task.

Category	**Number of Questions**	**Category**	**Number of Questions**
kinship	132	capital	2970
synonyms	3422	district-city	6466
currency	156	antonyms	2756
plural nouns	4830	past tense	3540
present tense	1560	neg. present tense	756

We evaluated the quality of word embedding models that we trained on:

- total accuracy
- accuracy on individual analogy task categories
- semantic accuracy and syntactic accuracy

3.3. Word Similarity Tasks

We used word similarity tasks to evaluate the quality of word embedding models extrinsically. We would like to evaluate how the word embedding models perform in an NLP task. We used *WordSimTr* dataset that is prepared by Üstün et al. [18] for Turkish word embedding models. For English models, we used five different word similarity datasets: *WordSim353* [22], *RW* [15], *RG* [23], *MC* [24], and *SCWS* [25]. Summary about word similarity datasets is in Table 3.5. One can find details about datasets in the following subsections.

3.3.1. Turkish Word Similarity Task

The word similarity data set for Turkish, *WordSimTr*, was prepared by Üstün et al. [18]. The dataset contains 138-word pairs, and their similarity scores changing from 1 to 10 where 1 represents weak similarity and 10 represents strong similarity.

Table 3.3. Examples for each category in the English analogy task.

Type of Relationship	**Word Pair 1**		**Word Pair 2**	
common capital-country	Paris	France	Rome	Italy
all capital-country	Copenhagen	Denmark	Ankara	Turkey
country-currency	Japan	yen	Argentina	peso
city-state	Austin	Texas	Sacramento	California
kinship	father	mother	son	daughter
opposite	acceptable	unacceptable	certain	uncertain
comparative	fast	faster	old	older
superlative	bright	brightest	weak	weakest
plural nouns	child	children	cat	cats
plural verbs	think	thinks	search	searches
present tense	code	coding	dance	dancing
past tense	dancing	danced	decreasing	decreased
adjective-adverb	amazing	amazingly	calm	calmly
nationality adjective	Albania	Albanian	Australia	Australian

Table 3.4. The number of questions in each category of the English analogy task.

Category	**Number of Questions**	**Category**	**Number of Questions**
common capitals	506	all capitals	4524
currency	866	city-state	2467
kinship	506	adj.-adv.	992
opposite	812	comparative	1332
superlative	1122	present tense	1056
nationality	1599	past tense	1560
plural nouns	1332	plural verbs	870

Table 3.5. Summary about word similarity datasets.

Dataset	**Number of word pairs**	**Score range**
WordSimTR	138	1-10
WordSim353	353	0-10
MC	28	0-4
RG	65	0-4
RW	2034	0-10
SCWS	2003	0-10

Similarity scores were calculated by taking the average of similarity scores given by 15 human annotators. 81-word pairs in the dataset are semantically similar words with at least two suffixes such as *kitaplarım*(my books)-*romanlarım*(my novels). Remaining 57-word pairs are semantically unrelated whereas they have orthographic similarity due to their suffixes such as *yazılardan*(from writings)-*kazılardan*(from excavations). Examples of the word similarity task are represented in Table 3.6.

Table 3.6. Examples for the word similarity task *WordSimTr*.

Word 1	**Word 2**	**Similarity Score**
kitaplarım (my books)	romanlarım (my novels)	7.611
saatlerce (for hours)	dakikalarca (for minutes)	6.583
koltuklarında (at their seats)	okullarında (at their schools)	1.543
kalabalıklar (crowds)	balıklar (fishes)	1.520

3.3.2. English Word Similarity Tasks: WordSim353

One of the similarity datasets that we used is *WordSim353* [22]. The dataset contains 353-word pairs with similarity scores that were given by human annotators. Thirteen human annotators rated 153-word pairs, and 16 human annotators evaluated 200-word pairs. Similarity scores are changing from 0 to 10, where 0 represents the weak similarity between the two words, and 10 represents strong similarity. The final similarity scores were calculated by taking the average of individual scores. Examples of the word similarity task are represented in Table 3.7.

Table 3.7. Examples for the word similarity task *WordSim353*.

Word 1	**Word 2**	**Similarity Score**
computer	keyboard	7.62
gem	jewel	8.96
production	hike	1.75
sugar	approach	0.88

3.3.3. English Word Similarity Tasks: MC

Miller and Charles 28 (MC) dataset was created by Resnik et al. [24] by taking a subset of the original Miller and Charles 30 (MC-30) dataset [26]. The dataset contains 28-word pairs. The final similarity scores in the dataset are the average of similarity scores given by 38 human annotators. Similarity scores are in a range from 0 to 4, where 0 represents weak similarity, and 4 represents strong similarity. Examples of the word similarity task are represented in Table 3.8.

3.3.4. English Word Similarity Tasks: RG

Rubenstein and Goodenough (RG) dataset was prepared for 65-word pairs in [23]. Fifty-one human annotators contributed to creating the data set. The average score was used for the final similarity values. Similarity scores for each word pair change

Table 3.8. Examples for the word similarity task *MC*.

Word 1	**Word 2**	**Similarity Score**
journey	voyage	3.84
food	fruit	3.08
noon	string	0.08
coast	hill	0.87

from 0 to 4, where the higher rating means the higher similarity of meaning. Examples of the word similarity task are represented in Table 3.9.

Table 3.9. Examples for the word similarity task *RG*.

Word 1	**Word 2**	**Similarity Score**
grin	smile	3.46
cemetery	graveyard	3.88
automobile	wizard	0.11
coast	hill	1.26

3.3.5. English Word Similarity Tasks: RW

Stanford Rare Word (*RW*) Similarity Dataset was introduced by Luong et al. in [15]. Since the dataset is that the most used word similarity datasets contain frequent words, Luong et al. prepared the dataset focusing on rare words. The dataset consists of 2034 word pairs with similarity scores changing from 0 to 10. Similarity scores were calculated by averaging individual scores given by 10 human annotators. Examples of the word similarity task are represented in Table 3.10.

3.3.6. English Word Similarity Tasks: SCWS

Huang et al. introduced Stanford's Contextual Word Similarities (SCWS) dataset [25]. The dataset consists of 2003 word pairs and their sentential contexts. Similarity

Table 3.10. Examples for the word similarity task *RW*.

Word 1	**Word 2**	**Similarity Score**
campfires	fire	9.33
urbanize	change	5.67
producing	together	1.11
conformations	balance	2.33

scores are changing from 0 to 10, where 0 means poor similarity relation. The final similarity scores were calculated by taking the average of individual scores given by 10 human annotators. We used the dataset by ignoring the provided context. We used only word pairs and similarity scores. Examples of the word similarity task are represented in Table 3.11.

Table 3.11. Examples for the word similarity task *SCWS*.

Word 1	**Word 2**	**Similarity Score**
Wednesday	weekday	6.3
advised	inform	7.1
collect	take	5.0
develop	mental	0.85

4. METHODOLOGY AND EXPERIMENTAL RESULTS

In this chapter, we explain methodologies that we used to train word embedding models in this study. We give explanations about experimental setups in Section 4.1. Later, we intrinsically and extrinsically evaluate word embedding models that we trained. We share experimental results on analogy and word similarity tasks in Section 4.2 and Section 4.3.

4.1. Methodology and Experimental Setup

In this study, we focused on Skip-gram architecture that was proposed by Mikolov et al. in [12]. As we mentioned previously in Chapter 2.1.1, in Skip-gram architecture, given a sequence of words $w_1, w_2, ...w_T$, the aim is to maximize the average log probability

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

where c is the training context size. [12] In other words, each target word is used as input and words within a certain range before and after the target word are predicted. Context range is also called as the (context) window size. In [12], Mikolov et al. stated that larger context window could result in higher accuracy, whereas the training time will increase. We tried to improve accuracy by changing the orientation of the context window rather than enlarging the window size.

In the original Skip-gram model, the context window includes both right and left side of the target word. We trained word embedding models where context window contains words in only one side of the target word. The original architecture of the Skip-gram is shown in Figure 4.1. The modified architectures that we used in this study are shown in Figure 4.2 and Figure 4.3.

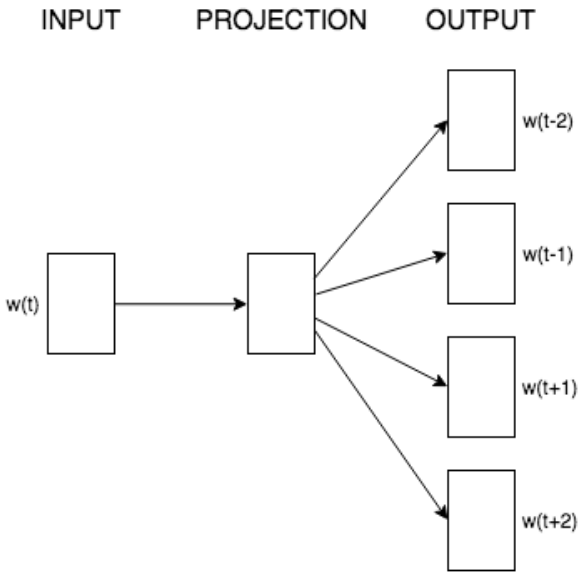


Figure 4.1. The original architecture of the Skip-gram where window size is 2. [1]

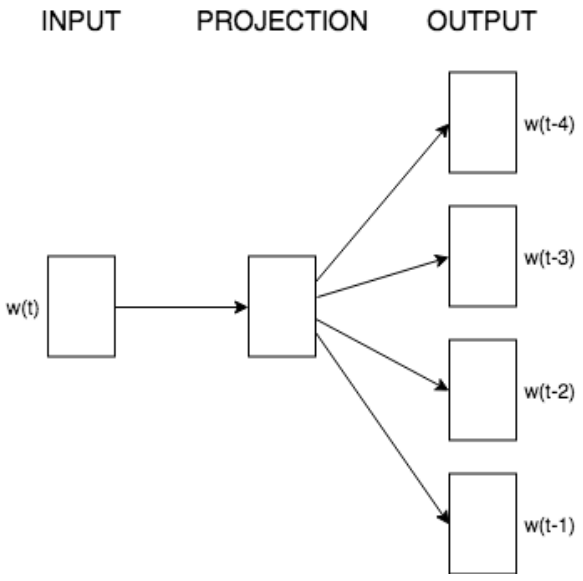


Figure 4.2. The modified architecture of the Skip-gram where the context window is chosen from only left side of the target word and window size is 4.

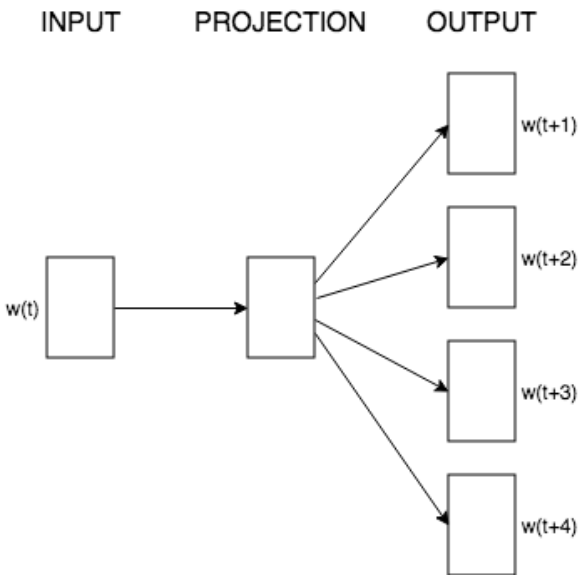


Figure 4.3. The modified architecture of the Skip-gram where the context window is chosen from only right side of the target word and window size is 4.

The models that we trained have three different context orientations. The models with *centered context orientation* were trained using the original Skip-gram architecture. For the models with *left context orientation*, context words were the words on the left side of the target word within the window size. For the models with *right context orientation*, context words were selected from the right side of the target word within the window size. Windowing operation for three different models is shown in Figure 4.4.

We trained Turkish word embedding models on *BounWebCorpus* and English word embedding models on *Wikipedia Dump Data*. Corpora that we used are described previously in Section 3.1.

We used Gensim Python library [27] to train word embedding models. We used the library as it is to train word embedding models on the original architecture of Skip-gram. We also changed the source code and used it to train word embedding models on modified architectures.

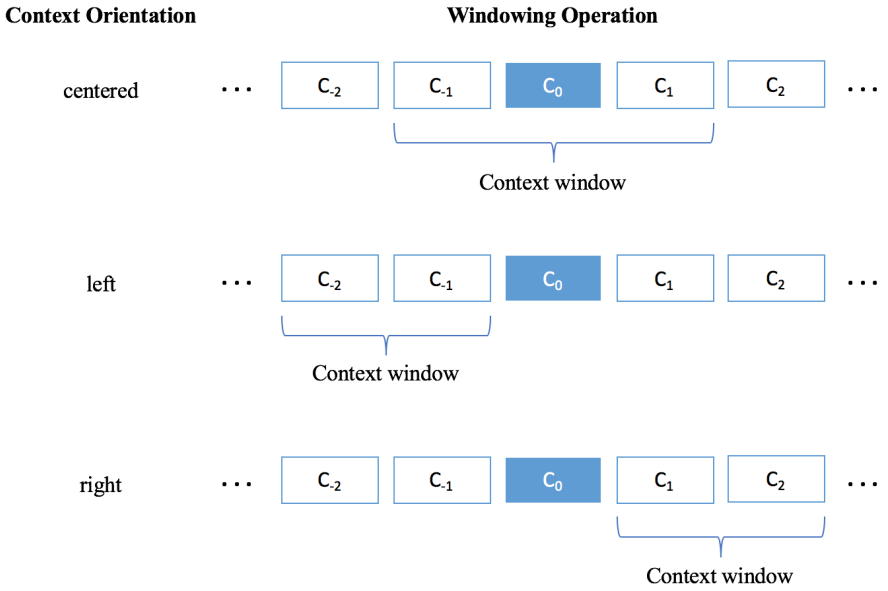


Figure 4.4. Windowing operation with different context orientations.

Configurations of models are as follows:

- The vector dimensions were set to 100, 200, and 300.
- The window size was set to 1, 2, 3, 4, and 5 for the original Skip-gram architecture whereas it was set to 2, 4, 6, 8, and 10 for modified versions of Skip-gram since the original architecture takes context words from both sides of the target word.
- The number of negative samples for negative sampling was set to five (default value set in the used library).
- The minimum frequency was set to five (default value set in the used library).
- The number of iterations (epochs) over the corpus was set to five (default value set in the used library).

We obtained word vectors for 663.832 unique words in Turkish, and 1.831.274 unique words in English.

In the following two sections, we show experimental results on analogy and word similarity tasks for Turkish and English word embedding models. We denote the orig-

inal Skip-gram architecture as *"centered context orientation"* in the following sections. We use *"left context orientation"* to represent the architecture that we used only words from the left side of the target word to train the models. *"right context orientation"* is used to represent the architecture that we used only words from the right side of the target word. We used the following name format for our trained models for convenience: *<context-orientation>_d-<vector-dimension>_w-<window-size>*, e.g. *centered_d-100_w-2.* In all tables, models are sorted by vector dimension, window size, and context orientation.

4.2. Intrinsic Evaluation of Word Representations

Analogy tasks contain statements like *"a is to b as c is to d"* as mention before in Section 3.2. We transformed these statements as a question and answer pairs. Our aim is finding a word that is similar to c in the same sense as b is similar to a and the correct word is d according to the previous statement. To answer these questions, we computed vectors by the formula

$$y = vector(b) - vector(a) + vector(c) \tag{4.1}$$

Then we searched the vector space for the closest word vector to y. To measure the distance between vectors, we used cosine distance by the formula

$$D_w = \frac{x_w y}{\|x_w\|\|y\|} \tag{4.2}$$

where y is the vector computed in Equation 4.1 and x_w is the vector of word w. The closest word vector was selected as an answer. We selected words as an answer using the following formula

$$w^* = argmax_w(D_w) \tag{4.3}$$

where D_w is the cosine distance of word w to vector y and w^* is the closest word to y. Answers were assumed as correct only if the answer given by language model was the same with the correct word in the question.

4.2.1. Evaluation of Turkish Word Representations

The Turkish word analogy task, described in detail in Section 3.2.1, was used to evaluate the quality of word embedding models that we trained using different configurations.

We report accuracy on semantic and syntactic questions in Table 4.1. Total accuracy is also represented in the table. The first column denotes the word embedding model. For the same vector dimension and the same window size, we denote the best results in each category as bold. The best results among models with the same vector dimension are denoted as underlined.

We observed that models trained by the original Skip-gram (denoted as *centered*) performs better in larger window sizes. In particular, when window sizes are greater than or equal to 6, the original Skip-gram models give the best results. On the other hand, models with smaller window sizes, which are less than 6, the modified Skip-gram models give better results.

The models with *right context orientation*, which we trained by choosing context word from the right side of the target word, perform better, especially on syntactic test set where window size equals to 2 and 4. The best results within the same vector dimension belong to models with the *right context orientation*. They perform better than all the other models with the same vector dimension. *right_d-100_w-4* has 27.20% accuracy where *right_d-100_w-8* has 27.03% accuracy.

Table 4.1 The results on the Turkish word analogy task, given as accuracy (%).

Model	**Semantic**	**Syntactic**	**Total**
centered_d-100_w-2	18.55	24.50	20.95
left_d-100_w-2	**19.10**	24.47	21.26
right_d-100_w-2	18.86	**25.26**	**21.44**
centered_d-100_w-4	**20.11**	25.96	**22.46**
left_d-100_w-4	19.38	26.17	22.11
right_d-100_w-4	18.95	**27.20**	22.27
centered_d-100_w-6	**19.45**	**25.85**	**22.02**
left_d-100_w-6	17.72	25.72	20.94
right_d-100_w-6	17.16	25.49	20.51
centered_d-100_w-8	**18.88**	**27.03**	**22.16**
left_d-100_w-8	16.40	25.72	20.15
right_d-100_w-8	16.25	24.82	19.70
centered_d-100_w-10	**17.92**	**26.56**	**21.39**
left_d-100_w-10	15.19	23.04	18.35
right_d-100_w-10	15.13	24.42	18.87
centered_d-200_w-2	21.72	26.97	23.83
left_d-200_w-2	25.47	26.24	25.78
right_d-200_w-2	**27.01**	**28.21**	**27.49**
centered_d-200_w-4	25.28	27.26	26.08
left_d-200_w-4	**27.12**	28.75	**27.78**
right_d-200_w-4	25.60	**29.61**	27.21
centered_d-200_w-6	**26.17**	**28.66**	**27.17**
left_d-200_w-6	25.33	28.33	26.54
right_d-200_w-6	25.40	27.51	26.25
centered_d-200_w-8	**24.30**	**28.56**	**26.02**
left_d-200_w-8	23.99	26.17	24.87

Table 4.1. The results on the Turkish word analogy task, given as accuracy (%). (cont.)

Model	**Semantic**	**Syntactic**	**Total**
right_d-200_w-8	23.86	26.50	24.92
centered_d-200_w-10	**25.74**	**28.06**	**26.68**
left_d-200_w-10	23.55	25.11	24.18
right_d-200_w-10	21.43	24.92	22.84
centered_d-300_w-2	19.30	23.30	20.91
left_d-300_w-2	22.72	24.79	23.56
right_d-300_w-2	**23.41**	**25.74**	**24.35**
centered_d-300_w-4	23.23	25.02	23.95
left_d-300_w-4	**26.62**	27.41	**26.94**
right_d-300_w-4	25.48	**27.49**	26.29
centered_d-300_w-6	26.08	**26.29**	26.17
left_d-300_w-6	**26.94**	25.66	**26.42**
right_d-300_w-6	24.66	25.40	24.96
centered_d-300_w-8	**26.24**	**26.23**	**26.24**
left_d-300_w-8	25.23	24.51	24.94
right_d-300_w-8	25.40	25.67	25.51
centered_d-300_w-10	**24.60**	**27.32**	**25.70**
left_d-300_w-10	23.20	23.25	23.22
right_d-300_w-10	24.39	23.81	24.16

In Table 4.2, we report accuracy results on every word analogy task categories. In two categories, *currency* and *kinship*, results are so close, too low, and indistinguishable because these categories have too few analogy questions. In some categories, the

original Skip-gram models outperform all the other models for all configuration settings, such as *plural nouns* and *synonyms*. In some categories, the modified Skip-gram models outperforms in all configuration settings, such as *capital* and *present tense*.

Table 4.2 Accuracy (%) of trained models on each category of the Turkish analogy task set.

Model	**Accuracy (%)**									
	capital	currency	district	kinship	neg. present tense	antonym	past tense	plural	present tense	synonyms
centered_d-100_w-2	8.96	1.28	24.85	6.82	14.55	14.33	15.00	35.07	17.81	19.90
left_d-100_w-2	11.75	2.56	26.51	8.33	16.14	14.99	17.54	32.88	17.88	16.31
right_d-100_w-2	12.29	2.56	25.88	10.61	17.99	13.28	18.98	33.19	18.15	17.15
centered_d-100_w-4	14.88	1.92	27.99	6.82	20.50	13.46	19.66	34.16	17.07	16.80
left_d-100_w-4	15.66	2.56	29.39	8.33	25.00	11.54	22.20	30.41	22.40	11.66
right_d-100_w-4	14.04	2.56	29.22	6.06	21.16	11.94	24.18	31.84	22.33	11.16
centered_d-100_w-6	13.91	1.28	28.65	6.82	20.37	12.37	20.51	32.44	19.91	14.29
left_d-100_w-6	14.01	1.28	27.46	6.82	23.81	11.54	23.33	27.45	26.72	9.12
right_d-100_w-6	13.74	2.56	26.08	6.82	24.21	11.72	22.09	28.55	24.29	9.09
centered_d-100_w-8	12.53	3.85	29.01	7.58	24.21	11.90	22.43	32.28	22.33	12.45
left_d-100_w-8	13.70	0.64	25.72	6.06	27.12	10.45	22.12	28.05	25.98	7.45
right_d-100_w-8	12.96	2.56	25.28	5.30	21.56	11.18	21.05	27.89	25.51	7.60
centered_d-100_w-10	14.18	1.92	26.83	5.30	23.54	12.01	23.36	30.02	24.43	10.70
left_d-100_w-10	15.42	3.21	22.46	6.06	21.03	10.41	20.54	24.14	26.45	6.34
right_d-100_w-10	13.97	1.28	23.28	4.55	23.15	9.54	21.95	25.94	26.05	6.63

Table 4.2 Accuracy (%) of trained models on each category of the Turkish analogy task set. (cont.)

Model	**Accuracy (%)**									
	capital	currency	district	kinship	neg. present tense	antonym	past tense	plural	present tense	synonyms
centered_d-200_w-2	13.16	2.56	27.92	7.58	13.10	15.93	16.69	40.17	15.59	23.79
left_d-200_w-2	17.64	2.56	37.12	8.33	24.07	16.73	18.50	34.68	18.35	19.52
right_d-200_w-2	17.21	2.56	41.48	7.58	26.06	15.93	20.85	35.73	22.40	19.61
centered_d-200_w-4	17.61	1.92	37.63	9.85	24.21	14.88	20.59	34.31	21.79	19.20
left_d-200_w-4	19.87	2.56	42.84	8.33	29.50	15.93	26.13	31.61	25.30	15.25
right_d-200_w-4	17.37	2.56	40.75	8.33	27.65	16.22	27.74	31.97	27.40	14.06
centered_d-200_w-6	18.62	1.92	40.72	7.58	24.47	15.09	24.21	34.06	23.82	16.60
left_d-200_w-6	19.60	3.85	40.39	6.82	32.67	15.82	26.13	29.71	26.86	11.89
right_d-200_w-6	19.33	2.56	40.73	8.33	28.31	15.49	26.47	28.34	26.86	12.04
centered_d-200_w-8	17.31	3.21	37.11	8.33	29.63	15.82	25.90	31.33	25.37	15.17
left_d-200_w-8	19.09	2.56	37.77	7.58	27.91	16.65	24.77	26.65	27.06	10.32
right_d-200_w-8	20.51	1.92	37.69	7.58	27.38	14.48	26.75	26.09	26.79	10.43
centered_d-200_w-10	18.15	2.56	41.75	9.09	28.70	15.49	25.93	30.10	26.18	12.74
left_d-200_w-10	20.13	2.56	37.47	7.58	29.37	14.55	25.51	23.52	27.19	9.64
right_d-200_w-10	17.95	1.92	33.35	9.09	26.59	14.55	24.12	24.78	26.45	9.35

Table 4.2 Accuracy (%) of trained models on each category of the Turkish analogy task set. (cont.)

Model	**Accuracy (%)**									
	capital	currency	district	kinship	neg. present tense	antonym	past tense	plural	present tense	synonyms
centered_d-300_w-2	12.86	3.85	22.92	6.06	9.92	14.84	13.08	36.71	10.86	23.03
left_d-300_w-2	16.87	2.56	30.65	8.33	18.12	14.44	19.55	32.17	16.67	21.33
right_d-300_w-2	16.06	1.28	33.13	8.33	20.50	14.44	18.22	34.53	17.68	20.66
centered_d-300_w-4	15.69	1.28	32.59	8.33	24.21	15.28	19.21	32.09	16.26	20.46
left_d-300_w-4	18.28	4.49	44.11	8.33	31.48	14.30	27.88	28.41	20.99	13.21
right_d-300_w-4	16.84	3.85	40.91	8.33	29.23	15.60	25.42	30.23	22.60	14.11
centered_d-300_w-6	16.13	3.85	40.81	9.09	24.74	16.40	22.09	31.64	19.70	17.01
left_d-300_w-6	19.09	5.13	45.45	10.61	28.04	14.73	25.90	25.28	25.10	11.05
right_d-300_w-6	18.32	3.21	40.66	12.12	31.35	14.51	24.18	26.17	22.74	10.29
centered_d-300_w-8	16.97	2.56	43.33	8.33	32.54	13.64	22.01	30.48	19.30	14.67
left_d-300_w-8	20.77	2.56	40.96	10.61	30.16	15.09	25.11	23.21	24.43	9.82
right_d-300_w-8	19.53	3.21	41.97	9.09	31.88	15.64	26.92	24.76	22.47	9.41
centered_d-300_w-10	17.51	3.85	39.55	9.09	28.31	14.95	27.46	28.43	22.87	12.51
left_d-300_w-10	18.79	2.56	37.00	9.85	32.01	15.17	24.10	21.14	23.62	9.50
right_d-300_w-10	21.58	1.92	38.23	11.36	29.37	15.97	25.03	21.68	25.03	9.59

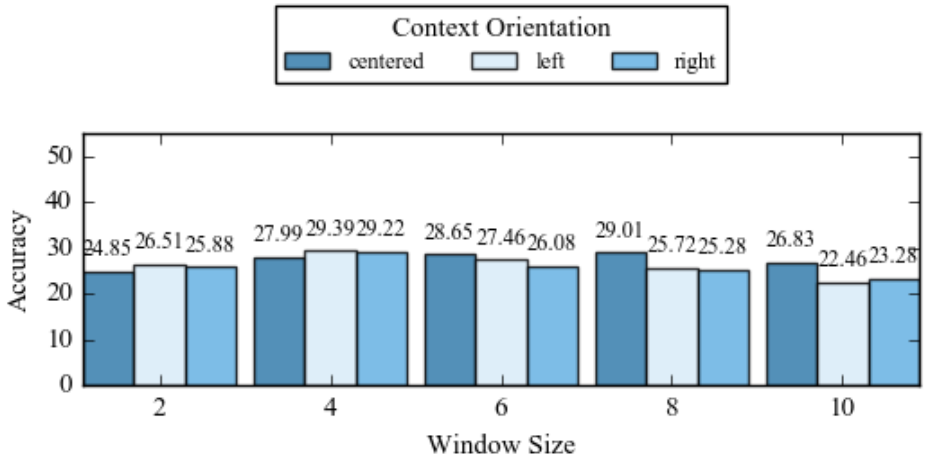
In Figure 4.5, Figure 4.6, and Figure 4.7, we report accuracy results on different analogy task categories to analyze them in detail.

In Figure 4.5, we report accuracy of trained models on *district-city* analogy questions. We observed that models with *left context orientation* perform better than the original Skip-gram models with larger window sizes. Larger vector dimension results in better accuracy. However, it is not the same for window size. For models with vector size 200, taking four words only from the left side for training gives a better result than taking more words from both sides for this particular analogy task category.

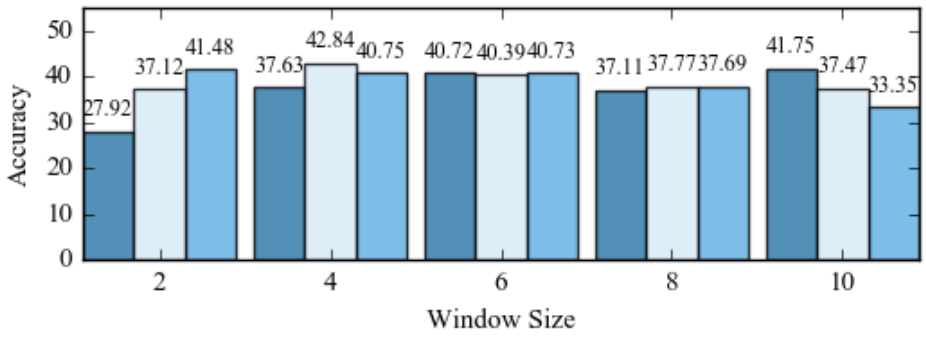
In Figure 4.6, we report accuracy of trained models on *present tense* analogy questions. We observed that modified models outperform the original Skip-gram models. Setting context orientation asymmetrically only one side of the target word results in better accuracy than setting a symmetrical context window for *present tense* analogy task. The results show that the bigger window sizes does not improve accuracy for this type of analogy questions.

In Figure 4.7, accuracy results on *capital-country* analogy questions are represented. In contrast with previously shown accuracy results in Figure 4.5 and Table 4.6, increasing the window size gives better accuracy results. The reason behind that the modified Skip-gram models perform better on this type of analogy questions than the original Skip-gram models may be that the more distant words are being used for training when we are looking only one side of the target word.

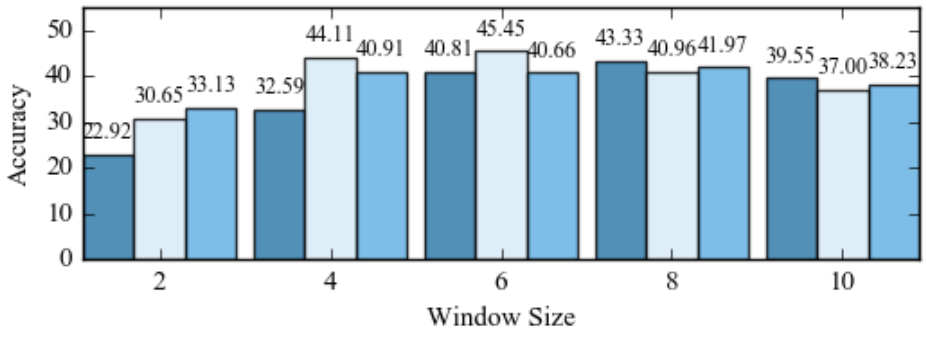
To sum up, we observe that the effects of window size and vector dimension are changing from task to task in Turkish. For some analogy task categories such as *capital, past tense,* and *negative present tense* analogy task questions, bigger window size has a positive impact on accuracy. On the other hand, for *plural nouns* and *synonyms* analogy questions, smaller window size results in a better accuracy.



a. Dimension size is set to 100.

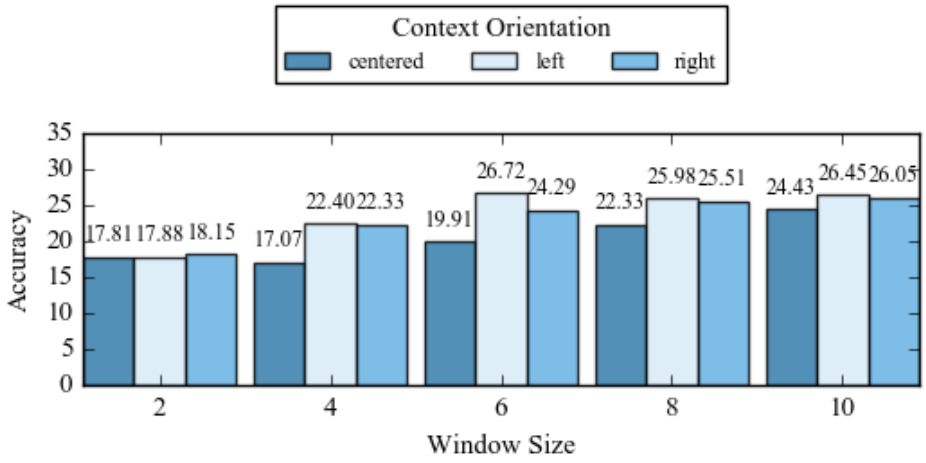


b. Dimension size is set to 200.

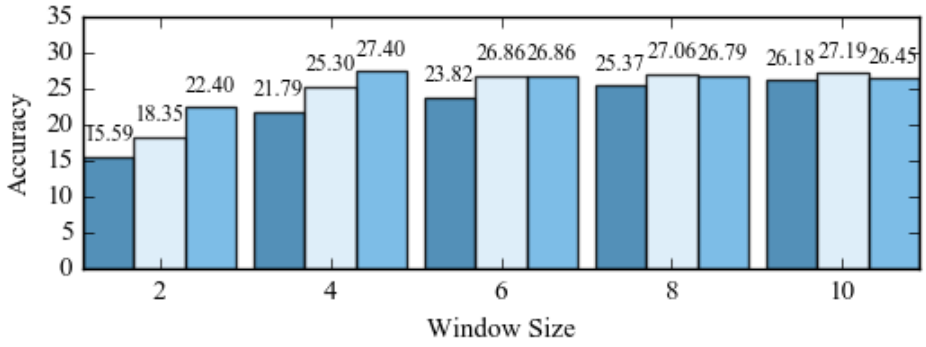


c. Dimension size is set to 300.

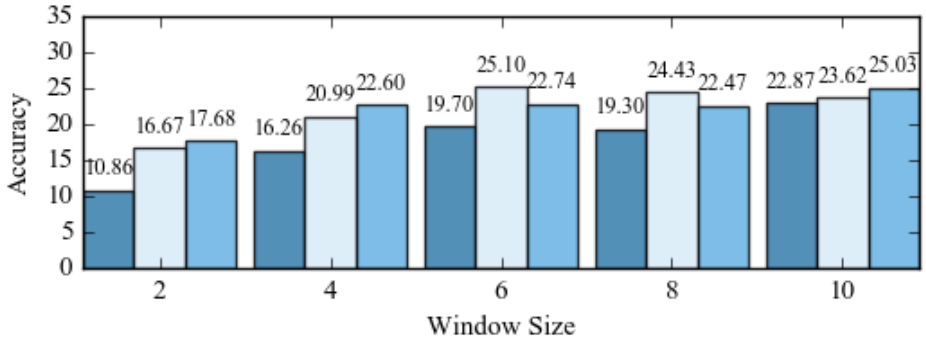
Figure 4.5. Accuracy (%) results on "district-city" analogy task questions.



a. Dimension size is set to 100.

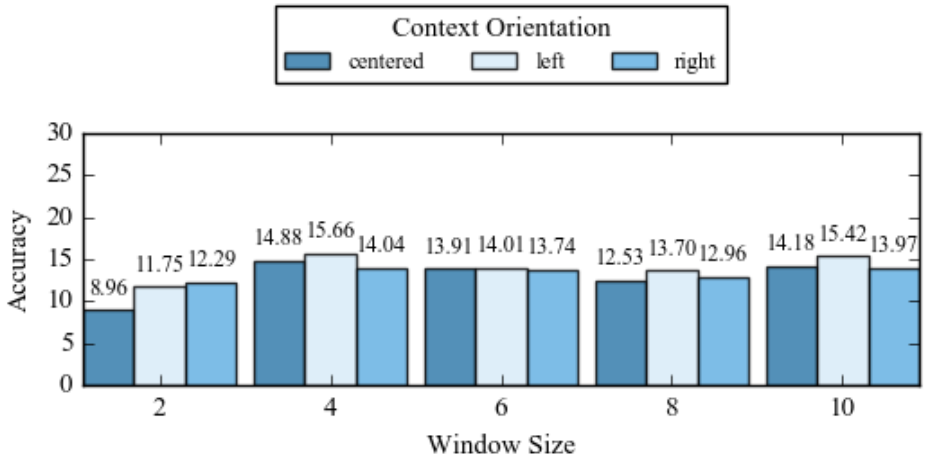


b. Dimension size is set to 200.

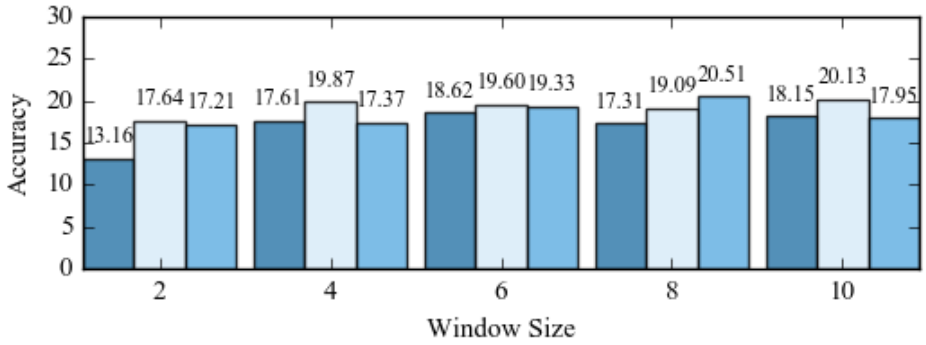


c. Dimension size is set to 300.

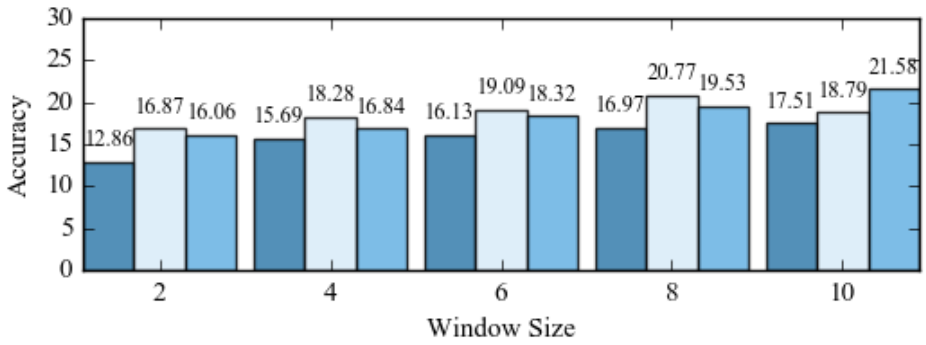
Figure 4.6. Accuracy (%) results on "present tense" analogy task questions.



a. Dimension size is set to 100.



b. Dimension size is set to 200.



c. Dimension size is set to 300.

Figure 4.7. Accuracy (%) results on "capital-country" analogy task questions.

4.2.2. Evaluation of English Word Representations

The English word analogy task, described in detail in Section 3.2.2, was used to evaluate the quality of our word embedding models.

We report accuracy on semantic and syntactic questions in Table 4.3. Total accuracy is also represented in the table. The first column denotes the word embedding model. For the same vector dimension and the same window size, we denote the best results in each category as bold. The best results among models with the same vector dimension are denoted as underlined. We observe that the original Skip-gram models perform better than the other in most of the cases. However, the models with *right context orientation* and *300-vector dimension* have better accuracy results on semantic questions.

Table 4.3 The results on the English word analogy task, given as accuracy (%).

Model	**Semantic**	**Syntactic**	**Total**
centered_d-100_w-2	18.77	**47.69**	**33.10**
left_d-100_w-2	**19.69**	37.74	28.63
right_d-100_w-2	19.01	37.18	28.01
centered_d-100_w-4	**26.11**	**47.40**	**36.66**
left_d-100_w-4	20.20	35.73	27.90
right_d-100_w-4	20.57	35.22	27.82
centered_d-100_w-6	**28.26**	**48.57**	**38.32**
left_d-100_w-6	20.13	32.79	26.40
right_d-100_w-6	20.08	32.62	26.29
centered_d-100_w-8	**30.16**	**49.06**	**39.52**
left_d-100_w-8	18.99	30.84	24.86
right_d-100_w-8	18.84	30.67	24.70
centered_d-100_w-10	**30.77**	**47.09**	**38.86**

Table 4.3. The results on the English word analogy task, given as accuracy (%). (cont.)

Model	**Semantic**	**Syntactic**	**Total**
left_d-100_w-10	18.99	27.76	23.34
right_d-100_w-10	18.90	28.26	23.54
centered_d-200_w-2	24.34	**55.51**	**39.78**
left_d-200_w-2	23.92	45.50	34.61
right_d-200_w-2	**24.45**	47.59	35.91
centered_d-200_w-4	**33.44**	**57.34**	**45.28**
left_d-200_w-4	25.36	42.09	33.65
right_d-200_w-4	26.70	42.79	34.67
centered_d-200_w-6	**35.37**	**56.78**	**45.98**
left_d-200_w-6	24.70	38.80	31.69
right_d-200_w-6	24.37	37.51	30.88
centered_d-200_w-8	**36.32**	**55.17**	**45.66**
left_d-200_w-8	22.68	34.73	28.65
right_d-200_w-8	23.17	36.03	29.54
centered_d-200_w-10	**36.95**	**53.88**	**45.34**
left_d-200_w-10	22.52	33.07	27.75
right_d-200_w-10	22.53	33.18	27.81
centered_d-300_w-2	25.09	58.15	41.47
left_d-300_w-2	24.79	48.48	36.53
right_d-300_w-2	**32.64**	**59.38**	**45.89**
centered_d-300_w-4	32.82	**58.69**	45.64
left_d-300_w-4	25.08	42.78	33.85
right_d-300_w-4	**37.78**	55.77	**46.69**
centered_d-300_w-6	35.47	**56.68**	45.98
left_d-300_w-6	25.09	39.20	32.08

Table 4.3. The results on the English word analogy task, given as accuracy (%). (cont.)

Model	**Semantic**	**Syntactic**	**Total**
right_d-300_w-6	**40.62**	51.73	**46.13**
centered_d-300_w-8	38.02	**56.09**	**46.97**
left_d-300_w-8	22.36	35.98	29.11
right_d-300_w-8	**38.73**	49.52	44.07
centered_d-300_w-10	38.60	**54.44**	**46.44**
left_d-300_w-10	21.60	32.22	26.86
right_d-300_w-10	**38.96**	46.28	42.59

In Table 4.4 and Table 4.5, accuracy results on each category in the analogy task are reported. In most of the categories, the original Skip-gram models outperform. Only when the window size is set to 2, models with *right context orientation* have better accuracy results on semantic analogy question categories.

Table 4.4. Accuracy (%) of trained models on *semantic question categories* of the English analogy task set.

Model	**Accuracy (%)**				
	common capitals	all capitals	currency	city-state	kinship
centered_d-100_w-2	45.26	19.94	8.20	9.44	63.04
left_d-100_w-2	48.62	22.72	6.70	11.35	48.02
right_d-100_w-2	48.42	21.44	8.66	11.35	44.47
centered_d-100_w-4	60.67	29.44	12.70	14.11	70.36

Table 4.4. Accuracy (%) of trained models on *semantic question categories* of the English analogy task set. (cont.)

Model	**Accuracy (%)**				
	common capitals	all capitals	currency	city-state	kinship
left_d-100_w-4	54.35	23.34	11.55	11.96	44.86
right_d-100_w-4	54.55	24.23	12.24	11.31	42.29
centered_d-100_w-6	71.34	33.73	13.39	13.34	66.40
left_d-100_w-6	52.77	24.16	10.28	11.43	39.53
right_d-100_w-6	54.94	22.50	10.05	13.86	41.50
centered_d-100_w-8	74.90	37.62	12.47	15.24	61.07
left_d-100_w-8	54.15	21.95	10.97	11.96	34.39
right_d-100_w-8	55.34	21.44	10.62	12.57	33.79
centered_d-100_w-10	70.55	39.26	12.47	16.54	56.72
left_d-100_w-10	55.14	21.79	9.24	12.00	38.14
right_d-100_w-10	51.78	20.82	12.24	13.58	34.98
centered_d-200_w-2	53.95	25.95	8.89	17.39	65.61
left_d-200_w-2	55.73	27.01	8.31	18.81	50.40
right_d-200_w-2	55.73	27.12	9.93	19.05	53.95
centered_d-200_w-4	74.11	39.15	14.32	23.31	67.00
left_d-200_w-4	52.96	28.56	10.97	21.93	50.59
right_d-200_w-4	57.31	30.53	13.74	20.96	52.96
centered_d-200_w-6	81.23	42.53	16.86	24.56	59.88
left_d-200_w-6	55.73	28.05	13.28	20.31	48.02
right_d-200_w-6	58.70	27.83	11.43	19.25	46.25
centered_d-200_w-8	80.43	45.07	14.55	24.93	59.88
left_d-200_w-8	58.50	24.71	11.55	18.81	45.06
right_d-200_w-8	59.68	25.86	10.62	19.25	43.87
centered_d-200_w-10	80.63	46.49	16.28	25.37	54.74
left_d-200_w-10	59.09	25.09	13.05	18.12	40.32

Table 4.4. Accuracy (%) of trained models on *semantic question categories* of the English analogy task set. (cont.)

Model	**Accuracy (%)**				
	common capitals	all capitals	currency	city-state	kinship
right_d-200_w-10	56.32	24.34	12.70	20.02	38.54
centered_d-300_w-2	53.16	26.55	10.05	20.47	63.64
left_d-300_w-2	60.28	26.90	10.97	20.63	50.20
right_d-300_w-2	71.54	36.38	16.74	27.28	59.88
centered_d-300_w-4	76.88	36.76	15.24	25.78	64.62
left_d-300_w-4	56.72	28.82	8.43	22.01	48.02
right_d-300_w-4	85.38	45.18	14.09	32.10	51.19
centered_d-300_w-6	80.63	40.56	13.97	30.81	57.11
left_d-300_w-6	53.95	28.16	12.59	22.46	47.63
right_d-300_w-6	82.81	49.49	16.17	34.29	53.95
centered_d-300_w-8	83.20	45.67	18.01	30.81	52.96
left_d-300_w-8	54.94	25.04	10.62	18.32	45.85
right_d-300_w-8	82.21	46.79	14.20	34.25	45.65
centered_d-300_w-10	85.97	46.62	16.51	32.10	49.21
left_d-300_w-10	54.15	22.83	12.93	19.42	41.90
right_d-300_w-10	79.84	47.48	15.70	34.01	43.87

Table 4.5 Accuracy (%) of trained models on *syntactic question categories* of the English analogy task set.

Model	**Accuracy (%)**								
	adj.-adv.	opposite	comparative	superlative	present tense	nationality	past tense	plural nouns	plural verbs
centered_d-100_w-2	9.78	12.56	73.05	48.13	44.60	42.84	45.71	43.99	62.99
left_d-100_w-2	8.77	8.25	52.40	28.34	36.55	50.47	36.67	36.04	37.47
right_d-100_w-2	8.06	5.30	52.03	27.27	35.13	51.28	38.01	36.11	33.68
centered_d-100_w-4	12.30	9.98	63.51	41.80	43.09	61.85	45.06	40.77	57.93
left_d-100_w-4	3.93	5.54	37.61	19.61	27.94	67.42	38.08	34.01	31.49
right_d-100_w-4	5.75	3.33	37.84	19.52	30.11	65.85	37.24	31.68	32.87
centered_d-100_w-6	12.00	9.73	57.43	38.68	43.56	69.11	49.42	47.15	52.99
left_d-100_w-6	5.44	2.83	29.58	13.55	25.00	72.11	35.13	31.08	26.09
right_d-100_w-6	4.54	2.22	27.78	12.75	27.37	71.11	35.26	32.13	25.75
centered_d-100_w-8	10.08	6.90	54.80	34.14	45.27	76.55	52.44	45.12	52.87
left_d-100_w-8	4.23	1.23	22.15	9.98	25.57	75.42	31.60	29.35	24.02
right_d-100_w-8	3.53	1.35	22.90	10.34	27.56	72.92	31.92	28.68	23.10
centered_d-100_w-10	9.88	7.76	49.55	30.57	43.84	76.92	49.94	43.32	51.15
left_d-100_w-10	3.93	0.62	17.94	8.29	21.12	71.98	28.85	25.90	20.92
right_d-100_w-10	4.23	0.62	17.04	8.38	20.27	76.11	29.68	25.00	21.03

Table 4.5 Accuracy (%) of trained models on *syntactic question categories* of the English analogy task set. (cont)

Model	**Accuracy (%)**								
	adj.-adv.	opposite	comparative	superlative	present tense	nationality	past tense	plural nouns	plural verbs
centered_d-200_w-2	11.59	17.36	83.33	53.21	46.69	52.10	52.18	56.46	72.99
left_d-200_w-2	6.45	9.98	69.22	36.63	39.30	60.85	41.54	43.02	44.02
right_d-200_w-2	7.36	10.96	69.37	37.17	42.33	65.35	41.60	46.55	47.93
centered_d-200_w-4	11.49	15.64	75.45	54.28	54.73	71.54	48.53	54.20	70.11
left_d-200_w-4	4.94	7.27	51.58	25.31	39.02	75.61	36.09	36.04	43.91
right_d-200_w-4	5.85	7.27	49.47	26.83	39.20	75.73	40.00	39.71	39.77
centered_d-200_w-6	9.88	15.27	67.94	47.68	54.45	77.42	52.44	55.41	64.94
left_d-200_w-6	2.62	4.68	36.56	23.44	34.28	77.67	37.37	37.69	32.18
right_d-200_w-6	3.93	4.43	38.44	19.79	30.21	75.42	35.00	37.31	33.79
centered_d-200_w-8	9.27	13.79	66.59	44.03	53.03	80.93	53.08	48.72	59.54
left_d-200_w-8	3.02	3.08	32.28	17.47	25.47	77.49	33.27	33.41	27.59
right_d-200_w-8	2.42	1.97	31.53	16.76	28.03	81.61	36.54	34.53	26.90
centered_d-200_w-10	8.97	13.30	65.62	39.48	48.30	83.99	54.10	46.77	54.25
left_d-200_w-10	2.32	0.37	25.15	13.01	26.52	79.24	34.62	32.66	22.53
right_d-200_w-10	3.73	0.62	22.15	15.69	25.09	80.36	34.36	32.58	24.94

Table 4.5 Accuracy (%) of trained models on *syntactic question categories* of the English analogy task set. (cont)

Model	**Accuracy (%)**								
	adj.-adv.	opposite	comparative	superlative	present tense	nationality	past tense	plural nouns	plural verbs
centered_d-300_w-2	9.07	21.67	83.11	55.61	54.92	58.60	51.28	59.31	70.92
left_d-300_w-2	6.55	11.08	70.57	41.35	43.18	65.23	42.95	47.22	46.21
right_d-300_w-2	9.07	16.63	75.38	52.94	50.66	75.73	55.51	59.38	70.57
centered_d-300_w-4	8.97	19.70	77.55	49.64	55.30	75.30	51.03	56.08	69.20
left_d-300_w-4	2.32	7.51	55.11	27.18	34.94	72.98	39.62	39.19	42.07
right_d-300_w-4	7.66	14.66	68.02	43.58	52.94	82.49	53.78	47.60	61.49
centered_d-300_w-6	8.57	17.36	72.75	49.02	51.33	80.18	52.24	48.20	62.87
left_d-300_w-6	2.32	5.30	39.94	24.69	31.91	77.42	37.69	35.06	36.09
right_d-300_w-6	8.97	10.47	62.46	35.74	45.93	84.30	50.26	43.92	56.21
centered_d-300_w-8	7.86	13.55	70.27	43.23	48.58	84.18	55.13	49.55	59.89
left_d-300_w-8	1.81	1.11	34.23	18.63	30.21	76.30	36.67	35.36	26.21
right_d-300_w-8	8.77	8.37	55.93	29.41	46.88	86.05	47.95	46.47	47.59
centered_d-300_w-10	7.86	13.42	65.99	40.55	51.80	84.74	48.91	48.57	59.31
left_d-300_w-10	2.02	0.49	19.44	12.30	26.61	78.92	32.82	34.01	24.25
right_d-300_w-10	9.38	7.51	52.10	23.53	40.81	86.43	43.72	41.29	47.93

We observed the effect of the window size and vector dimension on accuracy results. In Figure 4.8, results show that both smaller window size and smaller vector dimension give better accuracy results on *kinship* analogy questions.

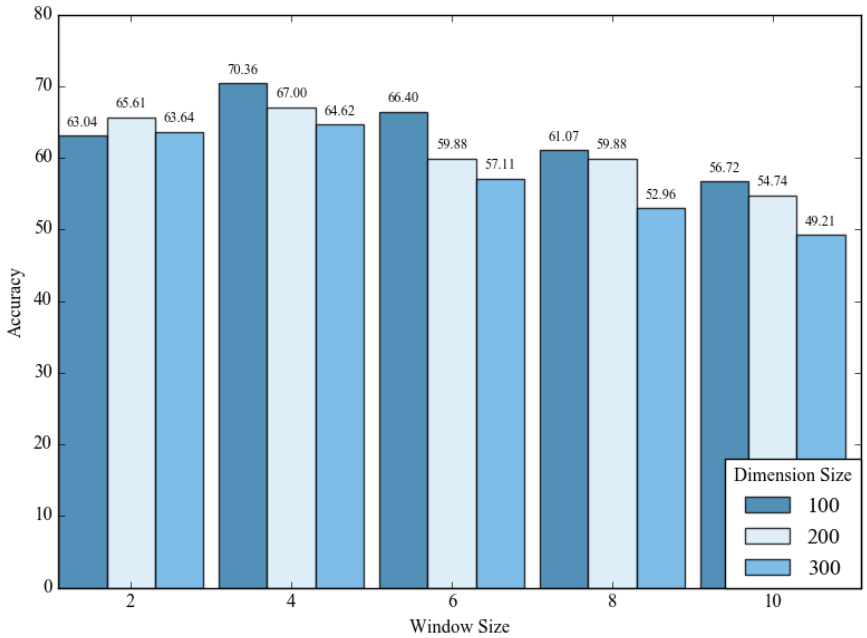


Figure 4.8. Accuracy (%) results of the original Skip-gram models on "kinship" analogy task questions. Accuracy on this task is in a negative proportional relationship with window size and vector dimension.

In Figure 4.9 and Figure 4.10, results show that both bigger window size and bigger vector dimension give better accuracy results on *city-state* and *nationality adjective* analogy questions.

In Figure 4.11 and Figure 4.12, results show that smaller window size gives better accuracy results on *comparative* and *opposite* analogy questions, whereas bigger vector dimension gives better accuracy results.

All in all, we observe that the effects of window size and vector dimension are changing from task to task in English, just like in Turkish. For some analogy task categories such as *capital-country, city-state,* and *nationality adjective* analogy task

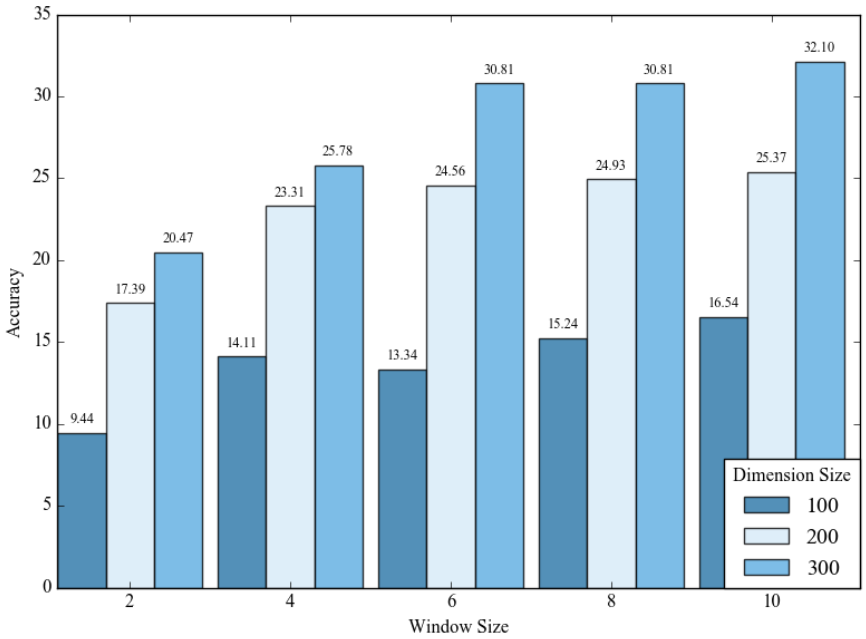


Figure 4.9. Accuracy (%) results of the original Skip-gram models on "city-state" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size and vector dimension.

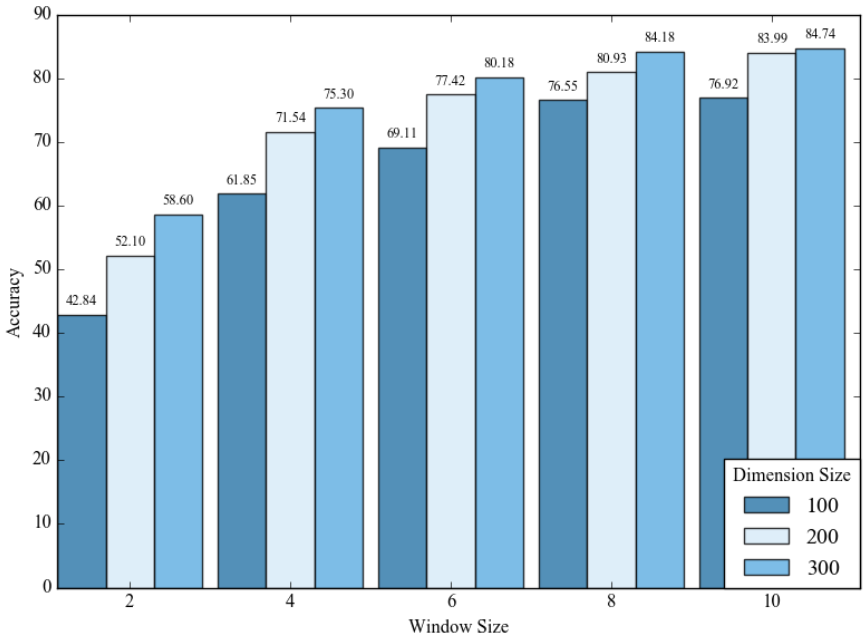


Figure 4.10. Accuracy (%) results of the original Skip-gram models on "nationality adjective" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size and vector dimension.

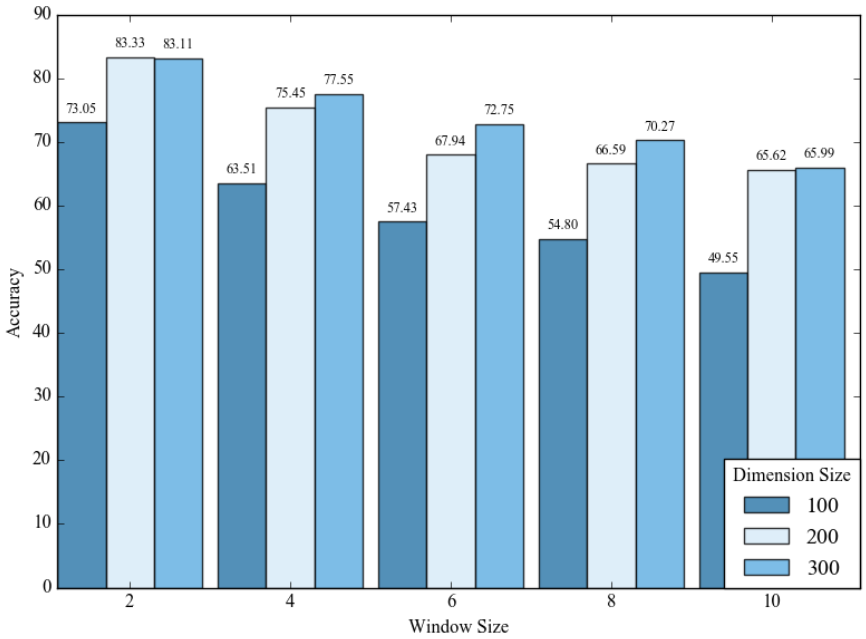


Figure 4.11. Accuracy (%) results of the original Skip-gram models on "comparative" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size but a negative relationship with vector dimension.

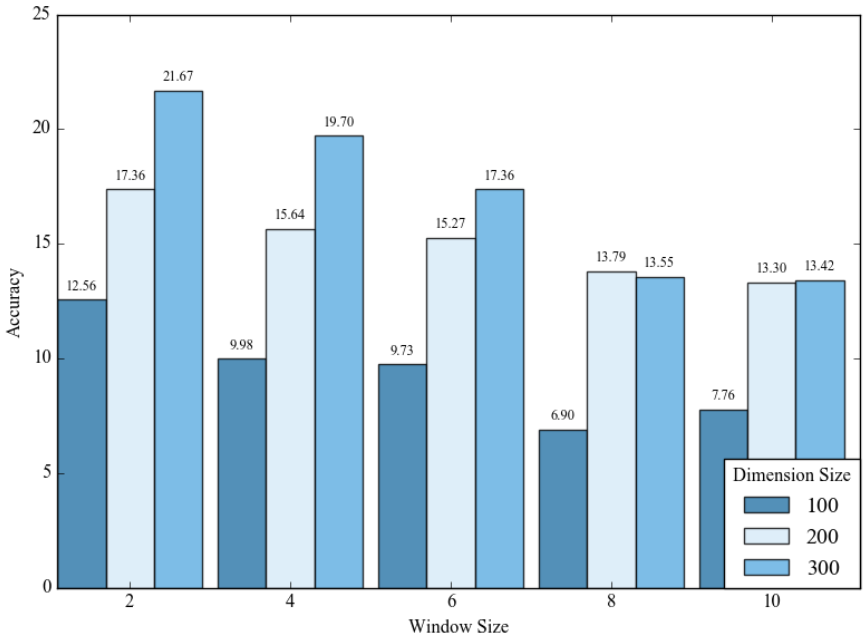


Figure 4.12. Accuracy (%) results of the original Skip-gram models on "opposite" analogy task questions. Accuracy on this task is in a positive proportional relationship with window size but a negative relationship with vector dimension.

questions, bigger window size has a positive impact on accuracy. On the other hand, for *comparative, superlative, opposite, plural verbs,* and *kinship* analogy questions, smaller window size results in a better accuracy.

4.3. Extrinsic Evaluation of Word Representations

We would like to see how word embedding models trained with different configurations perform in an NLP task. We used word similarity tasks described in Section 3.3 to evaluate the quality of word embedding models that we trained. We used Spearman's rank correlation [28] to evaluate how well the relationship between the similarity scores given by word embedding models and human annotators. Similarity scores are obtained by calculating the cosine similarity between the learned word vectors. We then calculated Spearman's rank correlation coefficient between human judgments and computed similarity scores.

4.3.1. Evaluation of Turkish Word Representations

We used *WordSimTR* word similarity dataset to evaluate the quality of Turkish word embedding models. We report Spearman's rank correlation ($\rho \times 100$) on the word similarity test set in Table 4.6. For the same vector dimension and the same window size, we denote the best results in each category as bold. The best results among models with the same vector dimension are denoted as underlined. We observed that the modified models have better results in most cases.

In Figure 4.13, Figure 4.14, and Figure 4.15, the Spearman's rank correlation results on *WordSimTR* are represented for models with 100, 200, and 300 vector dimensions respectively. The best results are reached when the window size is set to 4 or 6 among the models with the same vector dimension.

Table 4.6. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *WordSimTR*.

	Context Orientation		
Dimension-Window Size	**Centered**	**Left**	**Right**
d-100_w-2	73.69	**75.30**	71.51
d-100_w-4	75.04	**78.26**	72.36
d-100_w-6	**76.07**	74.49	71.30
d-100_w-8	**76.97**	74.50	**74.88**
d-100_w-10	74.00	75.72	**76.11**
d-200_w-2	69.86	**70.95**	71.48
d-200_w-4	63.60	**77.71**	68.98
d-200_w-6	72.85	**77.11**	71.73
d-200_w-8	68.52	**74.53**	66.05
d-200_w-10	70.87	**73.38**	70.63
d-300_w-2	75.43	73.73	**75.46**
d-300_w-4	74.99	**77.82**	76.24
d-300_w-6	76.37	77.42	**78.09**
d-300_w-8	76.08	75.90	**77.00**
d-300_w-10	72.45	72.55	**73.88**

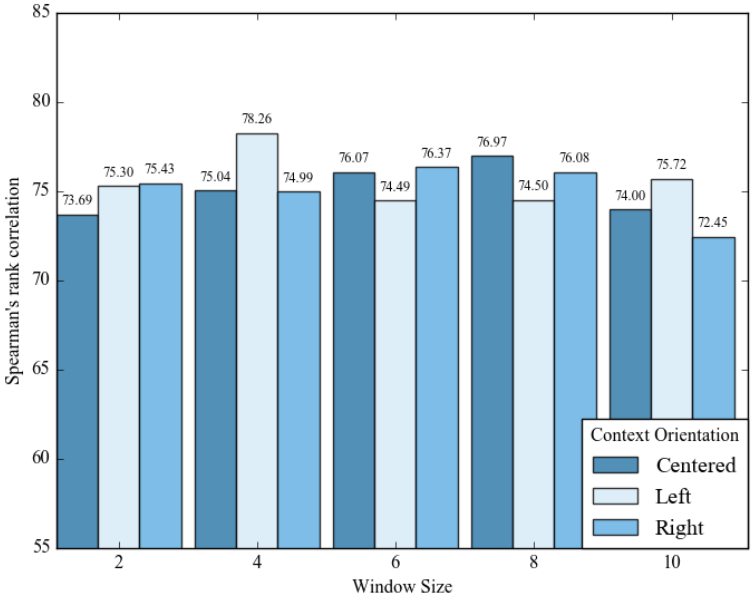


Figure 4.13. Spearman's rank correlation ($\rho \times 100$) on WordSimTR for models where vector dimension was set to 100 shown in the graphic.

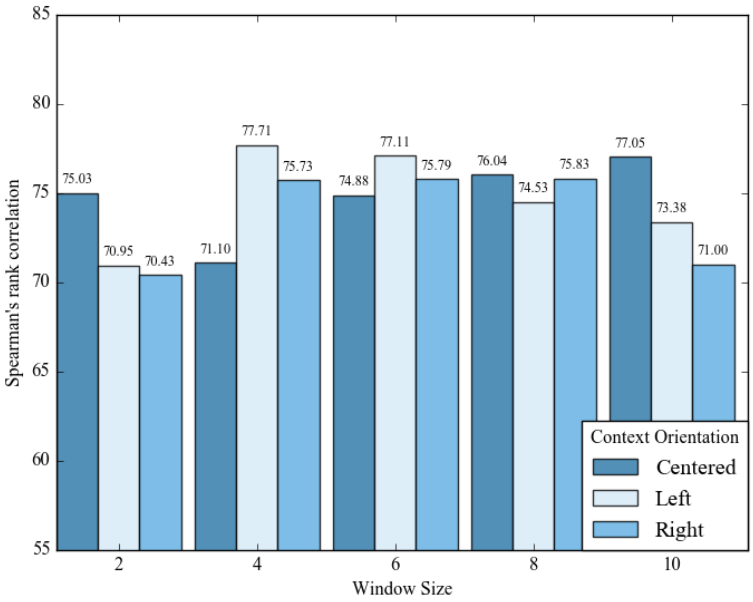


Figure 4.14. Spearman's rank correlation ($\rho \times 100$) on WordSimTR for models where vector dimension was set to 200 shown in the graphic.

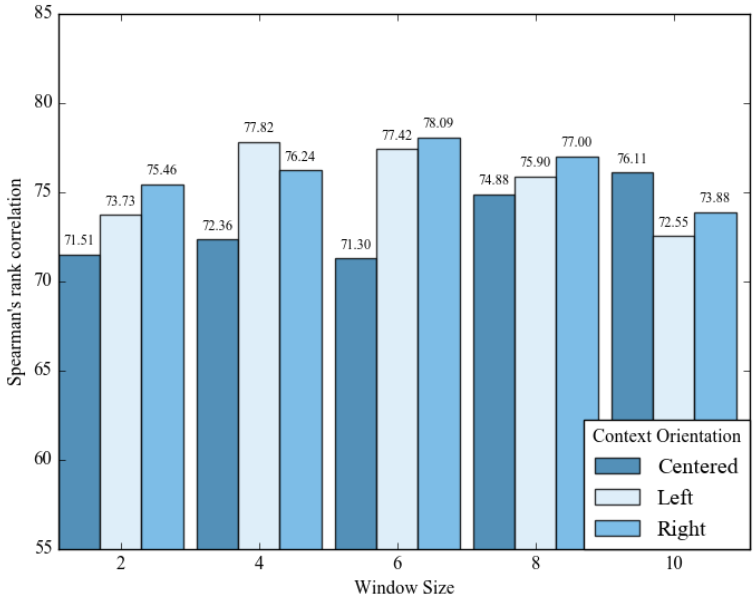


Figure 4.15. Spearman's rank correlation ($\rho \times 100$) on WordSimTR for models where vector dimension was set to 300 shown in the graphic.

4.3.2. Evaluation of English Word Representations

We used 5 different word similarity test sets to evaluate how good the models perform on word similarity task. In Table 4.7, Table 4.8, Table 4.9, Table 4.10, and Table 4.11, Spearman's rank correlation between human judgments and computed similarity scores are represented for the following word similarity tasks respectively; *WordSim353*, *MC*, *RG*, *RW*, and *SCWS*. For the same vector dimension and the same window size, we denote the best results in each category as bold. The best results among models with the same vector dimension are denoted as underlined.

In Table 4.7, Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *WordSim353* are reported. In Figure 4.16, Figure 4.17, and Figure 4.18, the Spearman's rank correlation results on *WordSim353* are represented in charts for models with 100, 200, and 300 vector dimensions respectively. We observed that an increase in window size and vector dimension results in a better correlation score. Where vector

dimension is set to 100 and 200, the original Skip-gram models perform better than the modified models. Where vector dimension is set to 300, models with *right context orientation* give better results.

Table 4.7. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *WordSim353*.

	Context Orientation		
Dimension-Window Size	**Centered**	**Left**	**Right**
d-100_w-2	**62.37**	62.04	61.74
d-100_w-4	**64.05**	63.60	62.70
d-100_w-6	**66.65**	63.16	63.10
d-100_w-8	**66.76**	62.62	62.23
d-100_w-10	**67.80**	63.67	62.52
d-200_w-2	**64.2**8	62.24	62.31
d-200_w-4	**66.55**	65.13	65.22
d-200_w-6	**67.69**	65.88	65.07
d-200_w-8	**67.90**	65.73	64.66
d-200_w-10	**68.00**	65.56	66.54
d-300_w-2	64.63	63.82	**67.50**
d-300_w-4	66.36	65.69	**69.01**
d-300_w-6	67.64	65.11	**69.10**
d-300_w-8	69.46	67.53	**70.14**
d-300_w-10	68.59	66.70	**70.70**

In Table 4.8, Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *MC* are reported. In Figure 4.19, Figure 4.20, and Figure 4.21, the Spearman's rank correlation results on *MC* are represented in charts for models with 100, 200, and 300 vector dimensions respectively. We observed that an increase in window size and vector dimension does not result in a better correlation score. The modified word embedding models achieve the best results for all vector dimensions that we trained

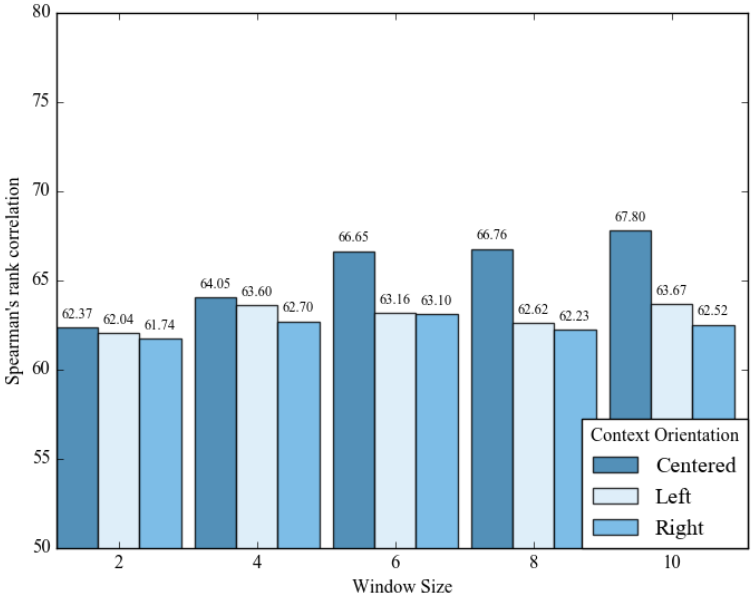


Figure 4.16. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 100 shown in the graphic.

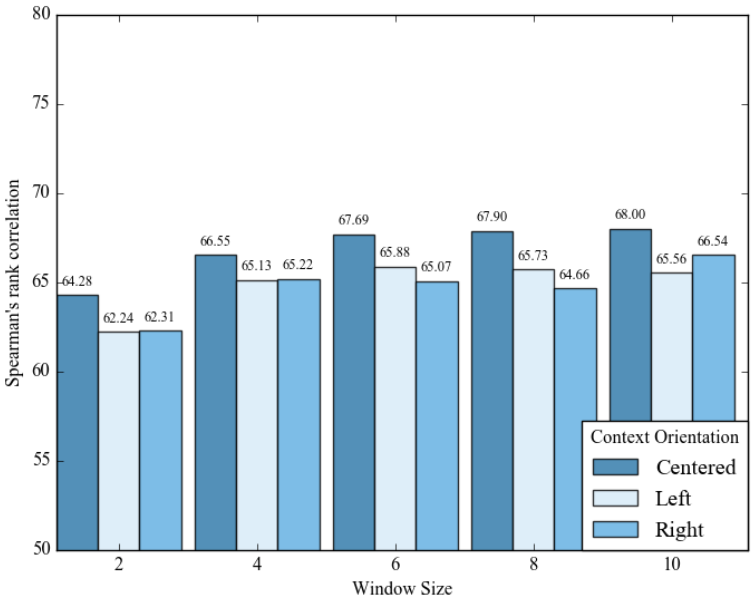


Figure 4.17. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 200 shown in the graphic.

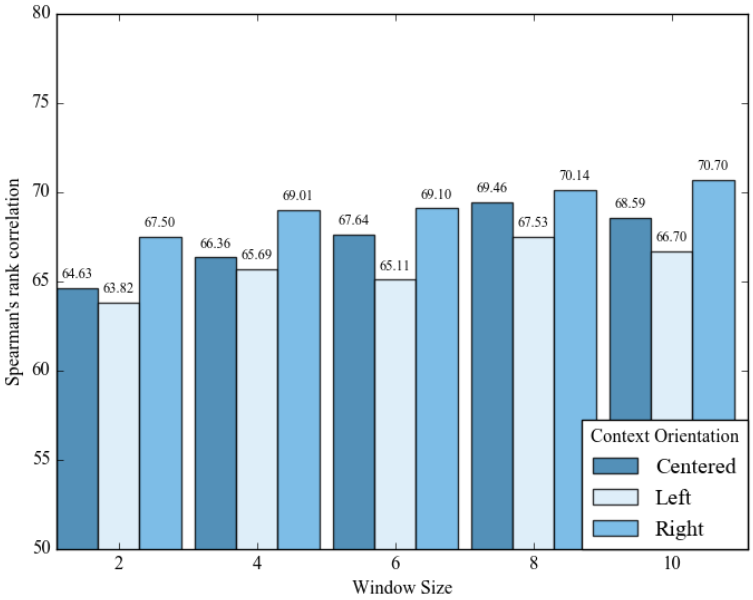


Figure 4.18. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 300 shown in the graphic.

our models with.

In Table 4.9, Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *RG* are reported. In Figure 4.22, Figure 4.23, and Figure 4.24, the Spearman's rank correlation results on *RG* are represented in charts for models with 100, 200, and 300 vector dimensions respectively. We observed that there are no big differences between correlation scores of different word embedding models. There is not a better model architecture or window size among the configuration settings that we used for training models. Only observation is that the bigger vector dimension gives better correlation results.

In Table 4.10, Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *RW* are reported. In Figure 4.25, Figure 4.26, and Figure 4.27, the Spearman's rank correlation results on *RW* are represented in charts for models with 100, 200, and 300 vector dimensions respectively. We observed that the original Skip-gram models

Table 4.8. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *MC*.

	Context Orientation		
Dimension-Window Size	**Centered**	**Left**	**Right**
d-100_w-2	73.16	**78.85**	74.17
d-100_w-4	73.35	**79.29**	76.14
d-100_w-6	75.15	75.24	**75.95**
d-100_w-8	**75.29**	74.63	72.85
d-100_w-10	**76.36**	73.92	74.55
d-200_w-2	**77.48**	76.58	75.62
d-200_w-4	76.19	80.30	**81.78**
d-200_w-6	75.24	76.63	**83.12**
d-200_w-8	77.18	75.98	**81.40**
d-200_w-10	**78.28**	77.75	77.92
d-300_w-2	**80.74**	78.17	78.38
d-300_w-4	77.37	77.81	**79.64**
d-300_w-6	78.99	77.67	**80.03**
d-300_w-8	78.58	**85.50**	80.88
d-300_w-10	76.63	**81.45**	79.59

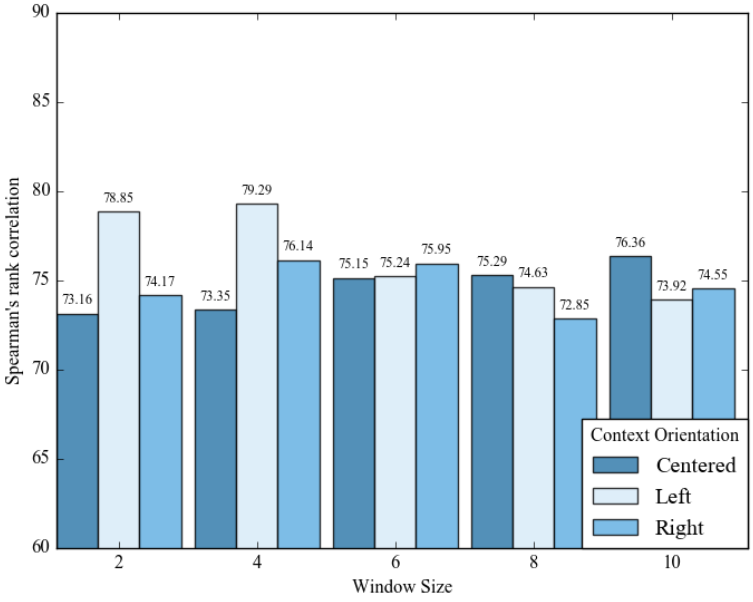


Figure 4.19. Spearman's rank correlation ($\rho \times 100$) on MC for models where vector dimension was set to 100 shown in the graphic.

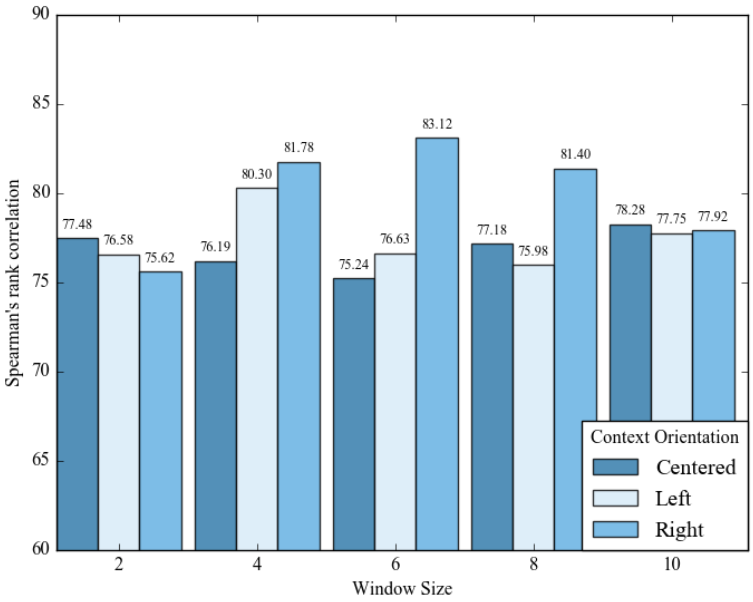


Figure 4.20. Spearman's rank correlation ($\rho \times 100$) on MC for models where vector dimension was set to 200 shown in the graphic.

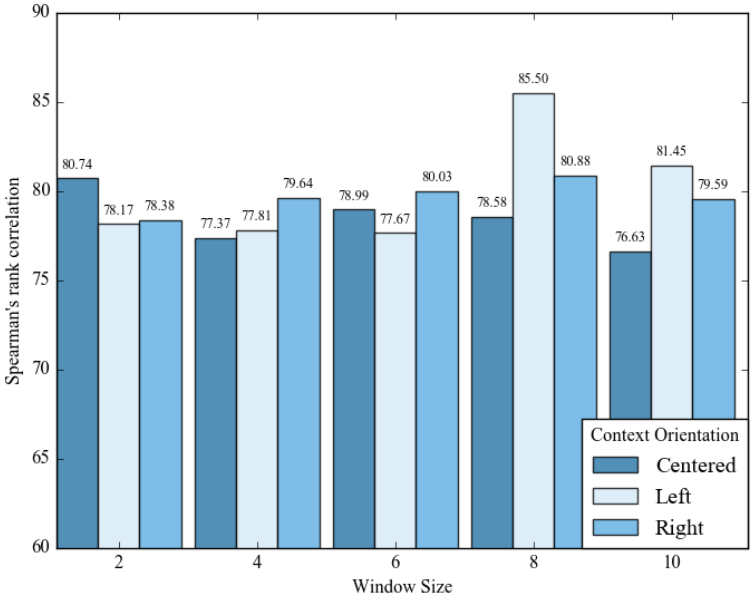


Figure 4.21. Spearman's rank correlation ($\rho \times 100$) on MC for models where vector dimension was set to 300 shown in the graphic.

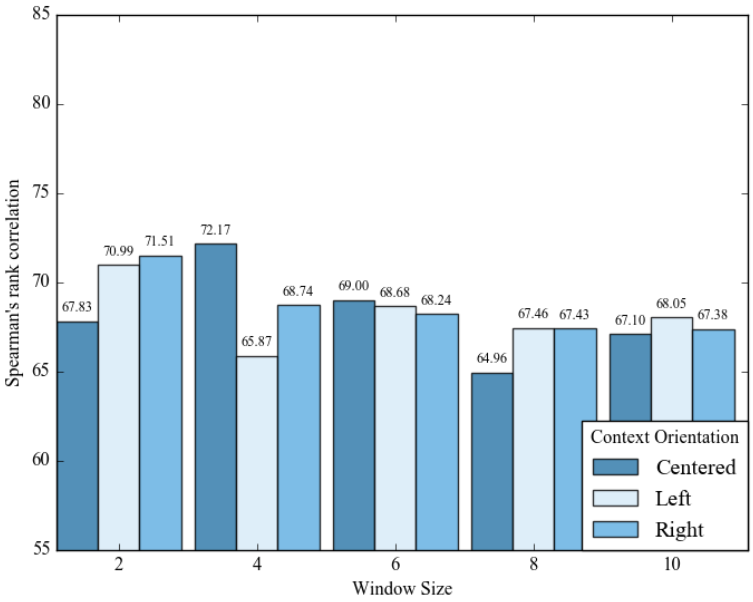


Figure 4.22. Spearman's rank correlation ($\rho \times 100$) on RG for models where vector dimension was set to 100 shown in the graphic.

Table 4.9. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *RG*.

	Context Orientation		
Dimension-Window Size	**Centered**	**Left**	**Right**
d-100_w-2	67.83	70.99	**71.51**
d-100_w-4	**72.17**	65.87	68.74
d-100_w-6	**69.00**	68.68	68.24
d-100_w-8	64.96	**67.46**	67.43
d-100_w-10	67.10	**68.05**	67.38
d-200_w-2	**71.52**	69.44	70.91
d-200_w-4	72.57	**73.34**	71.52
d-200_w-6	71.28	**72.36**	71.82
d-200_w-8	71.55	71.33	**73.62**
d-200_w-10	71.07	**73.12**	69.48
d-300_w-2	**73.77**	73.28	71.96
d-300_w-4	**72.18**	68.62	71.68
d-300_w-6	70.75	69.19	**71.10**
d-300_w-8	71.04	**74.15**	69.59
d-300_w-10	71.44	**74.34**	70.13

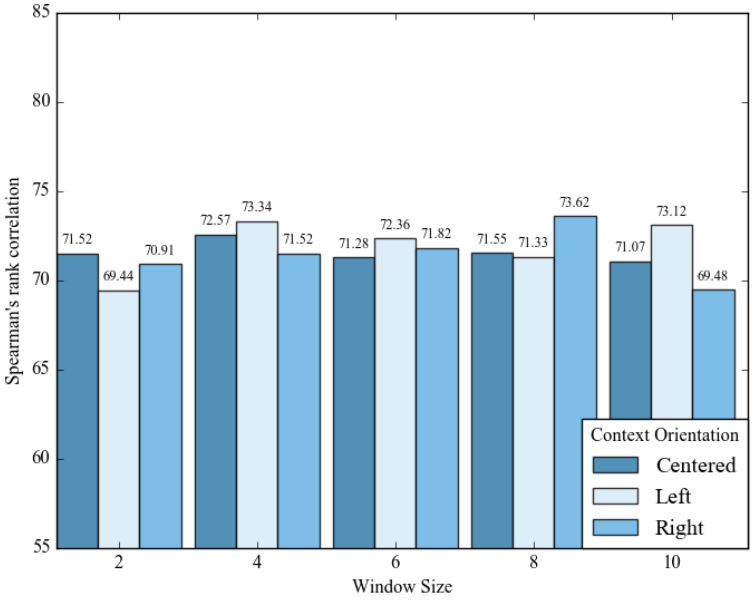


Figure 4.23. Spearman's rank correlation ($\rho \times 100$) on RG for models where vector dimension was set to 200 shown in the graphic.

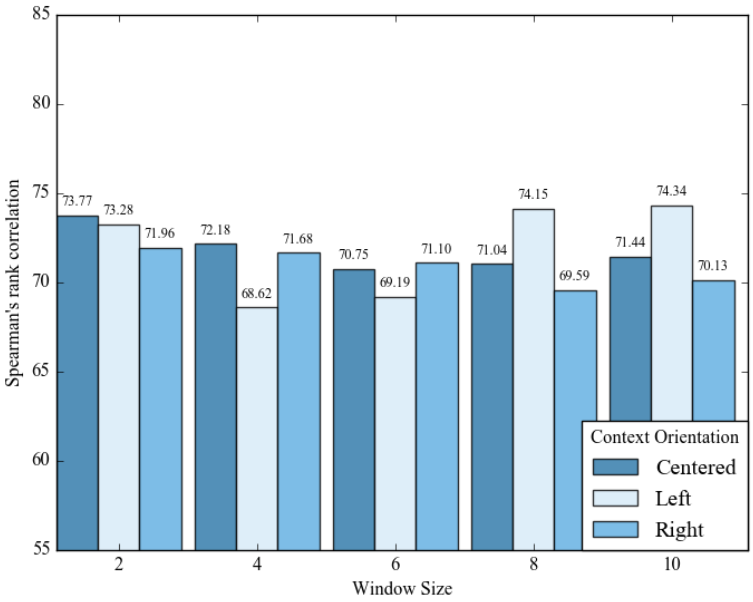


Figure 4.24. Spearman's rank correlation ($\rho \times 100$) on RG for models where vector dimension was set to 300 shown in the graphic.

give better results on this test set. Additionally, as the vector dimension increases, results are getting better.

Table 4.10. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *RW*.

	Context Orientation		
Dimension-Window Size	**Centered**	**Left**	**Right**
d-100_w-2	39.85	40.57	**41.51**
d-100_w-4	40.60	40.45	**40.97**
d-100_w-6	**41.06**	40.56	39.56
d-100_w-8	**41.84**	39.74	39.56
d-100_w-10	**41.66**	39.38	39.24
d-200_w-2	42.33	**43.49**	43.04
d-200_w-4	**42.91**	42.76	42.75
d-200_w-6	**43.52**	41.89	41.65
d-200_w-8	**43.36**	42.19	41.64
d-200_w-10	**43.66**	40.56	40.27
d-300_w-2	42.99	42.90	**43.41**
d-300_w-4	**44.09**	43.30	43.76
d-300_w-6	**43.71**	42.52	43.03
d-300_w-8	43.42	41.45	**43.76**
d-300_w-10	**43.45**	40.36	42.67

In Table 4.11, Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *SCWS* are reported. In Figure 4.28, Figure 4.29, and Figure 4.30, the Spearman's rank correlation results on *SCWS* are represented in charts for models with 100, 200, and 300 vector dimensions respectively. We observed that the results are changing slightly between different models. The original Skip-gram models outperform the other models in most cases.

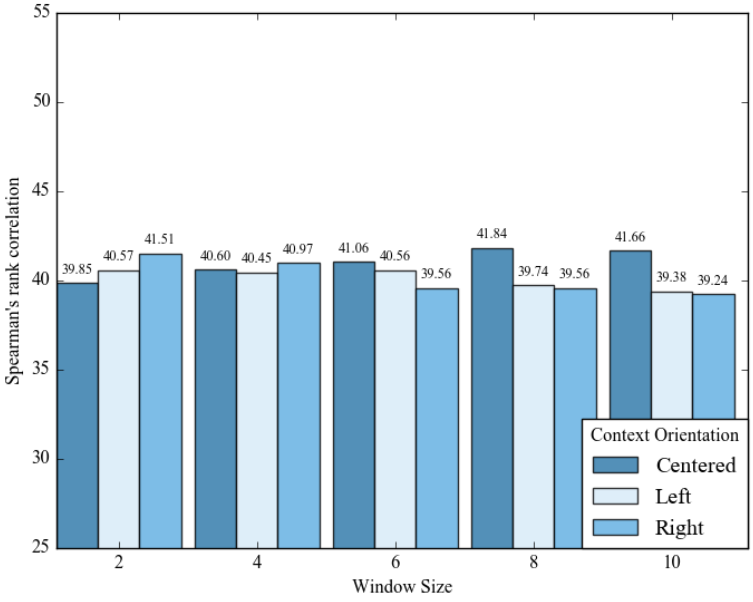


Figure 4.25. Spearman's rank correlation ($\rho \times 100$) on RW for models where vector dimension was set to 100 shown in the graphic.

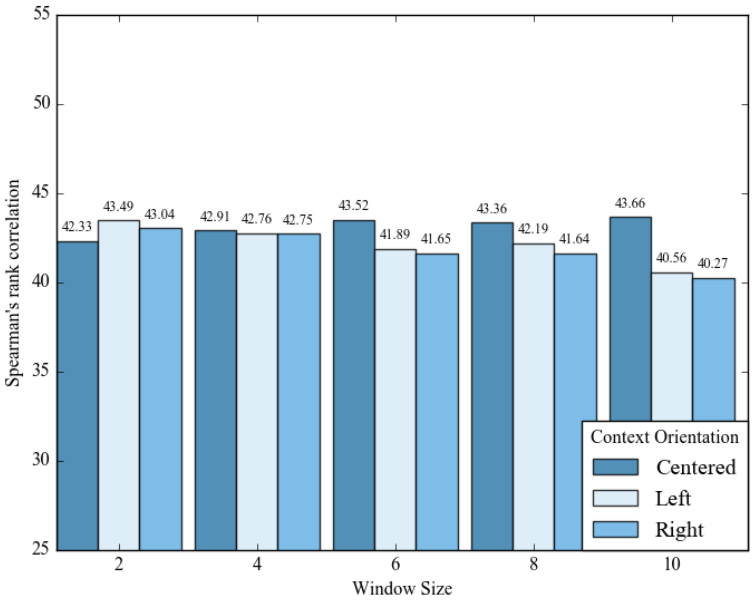


Figure 4.26. Spearman's rank correlation ($\rho \times 100$) on RW for models where vector dimension was set to 200 shown in the graphic.

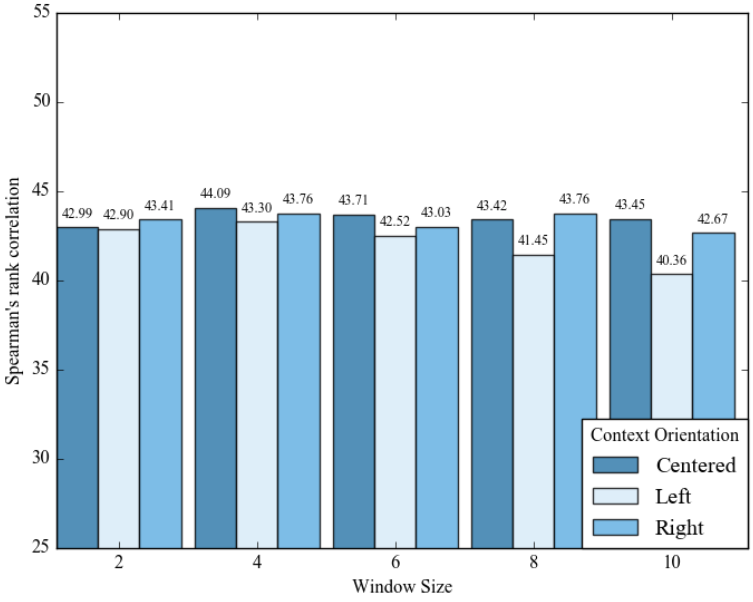


Figure 4.27. Spearman's rank correlation ($\rho \times 100$) on RW for models where vector dimension was set to 300 shown in the graphic.

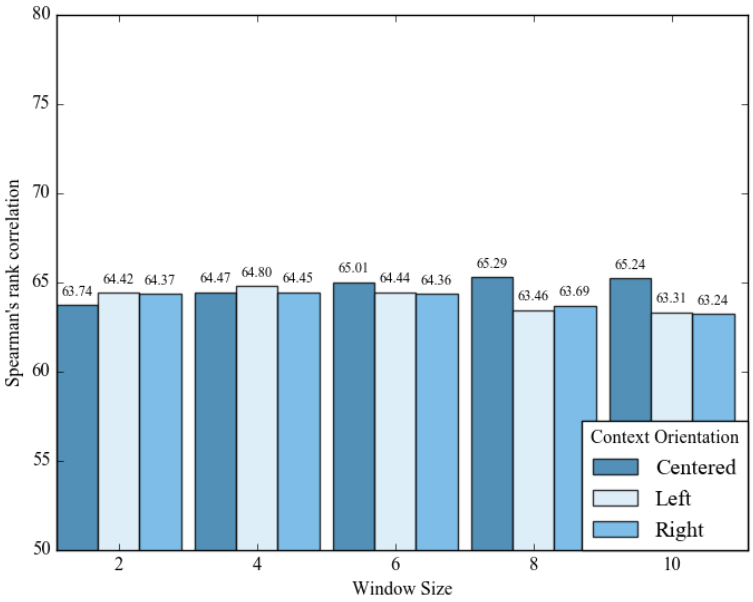


Figure 4.28. Spearman's rank correlation ($\rho \times 100$) on SCWS for models where vector dimension was set to 100 shown in the graphic.

Table 4.11. Spearman's rank correlation ($\rho \times 100$) on the word similarity test set *SCWS*.

	Context Orientation		
Dimension-Window Size	**Centered**	**Left**	**Right**
d-100_w-2	63.74	**64.42**	64.37
d-100_w-4	64.47	**64.80**	64.45
d-100_w-6	**65.01**	64.44	64.36
d-100_w-8	**65.29**	63.46	63.69
d-100_w-10	**65.24**	63.31	63.24
d-200_w-2	**65.71**	65.58	65.55
d-200_w-4	**66.21**	65.65	65.69
d-200_w-6	**66.34**	65.59	65.41
d-200_w-8	**66.01**	65.06	65.31
d-200_w-10	**65.61**	64.36	64.49
d-300_w-2	66.36	65.86	**66.68**
d-300_w-4	**66.60**	65.66	66.19
d-300_w-6	**66.50**	65.53	65.82
d-300_w-8	**66.23**	64.79	65.34
d-300_w-10	**65.99**	64.45	65.17

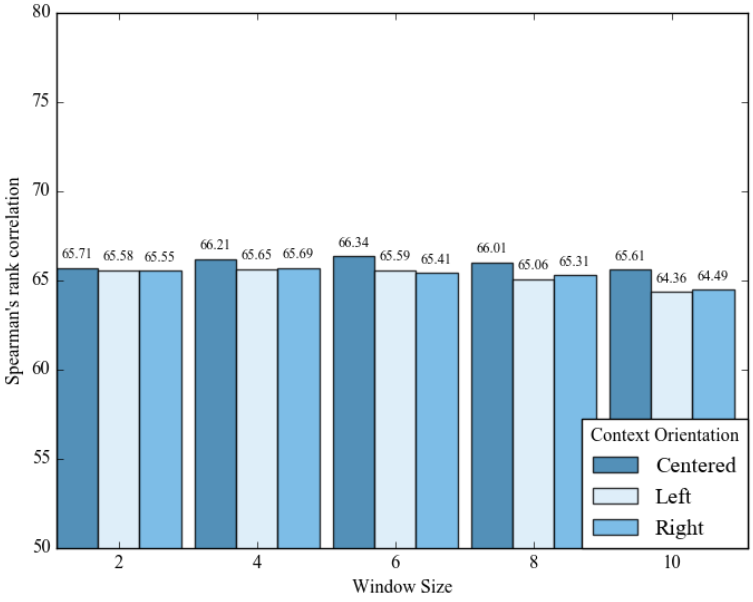


Figure 4.29. Spearman's rank correlation ($\rho \times 100$) on SCWS for models where vector dimension was set to 200 shown in the graphic.

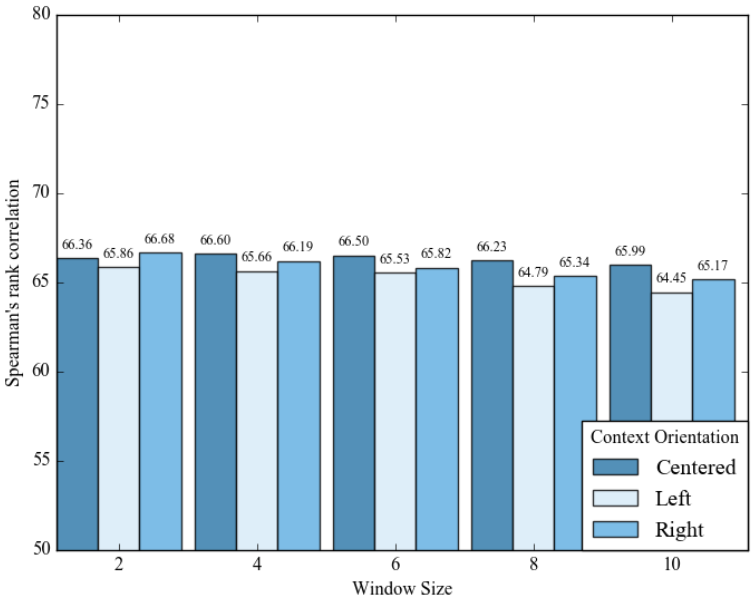


Figure 4.30. Spearman's rank correlation ($\rho \times 100$) on SCWS for models where vector dimension was set to 300 shown in the graphic.

5. CONCLUSIONS

5.1. Conclusion

In this thesis study, word embedding models on Turkish and English were trained with different configurations. We focused on the word2vec methodology and tried to improve quality by changing the orientation of context windows during training. By changing the context window orientation, we aimed to train models with better accuracy results without increasing the training time.

We proposed two new models based on Skip-gram modeling. We changed the window orientation from looking at both sides of the target word to looking only one side of it. Since word order in the sentence might carry useful information, looking words come before a word or words come after might be more helpful than looking both sides.

In addition to the change in context orientation, we trained models with different window sizes and vector dimensions. We observed how experimental results are changing with the change of window size and vector dimension.

We conducted experiments on analogy and word similarity tasks in order to evaluate the quality of word embedding models. Our observations, restricted to configurations used, are as follows:

- Our findings for Turkish word embedding models;
  (i) The models with *right context orientation* perform better on the total of syntactic tasks. The best results within the same vector dimension belong to models with the *right context orientation*.
  (ii) The modified models outperform the original Skip-gram models on *capital*, *district-city*, *present tense*, *negative present tense*, and *past tense* analogy questions.

(iii) For word embedding models, the following analogy tasks are in a positive relationship with window size: *past tense, negative present tense, present tense*, and *district-city*. Whereas, the following analogy tasks are in a negative relationship: *plural nouns* and *synonyms*.

(iv) The modified models gave better results on the word similarity task in most cases. The modified models reached the best results among the models with the same vector dimension.

- Our findings for English word embedding models;

(i) In most of the analogy task categories, the original Skip-gram models outperform.

(ii) Only when the window size is set to 2, models with right context orientation have better accuracy results on semantic analogy question categories.

(iii) For word embedding models, the following analogy tasks are in a positive relationship with window size: *capital-country, city-state*, and *nationality adjective*. Whereas, the following analogy tasks are in a negative relationship: *comparative, superlative, opposite, plural verbs*, and *kinship*.

(iv) For word embedding models, the following analogy tasks are in a positive relationship with vector dimension: *opposite, comparative, capital-country, city-state*, and *nationality adjective*. Whereas, the following analogy tasks are in a negative relationship: *adjective-adverb* and *kinship*.

All in all accuracy results on each analogy task category may be useful to researchers that would like to use word embedding models to solve domain-specific NLP problems. One should use a model with small window size and small vector dimension if he/she works on a Turkish NLP task where kinship relations are more important for the task. If one works on a task where syntactical analogy relations, such as plural forms of nouns and synonyms, in Turkish are more important to be captured, he/she should use a model with small window size but larger vector dimension.

For English NLP tasks, according to our observations, one should use a model with small window size and big vector dimension if he/she works on a task where

opposite and comparative noun relations are more important to capture. On the other hand, if city-state and nationality adjective analogy relations are more important for the NLP task, one should use bigger window size and bigger vector dimension.

5.2. Future Work

The following steps can enlarge the scope of this study:

- adding word embedding models with larger vector dimensions and window sizes.
- adding word embedding models trained on different corpora with different sizes.
- adding word embedding models with different negative sampling and minimum frequency configurations.
- adding different NLP tasks for extrinsic evaluation, such as named entity recognition and sentiment analysis.

REFERENCES

1. Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space", *ICLR Workshop*, 2013.

2. Bengio, Y., R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model", *Journal of machine learning research*, Vol. 3, No. Feb, pp. 1137–1155, 2003.

3. Turian, J., L. Ratinov and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning", *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 384–394, Association for Computational Linguistics, 2010.

4. Collobert, R. and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning", *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, ACM, 2008.

5. Socher, R., C. C. Lin, C. Manning and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks", *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 129–136, 2011.

6. Turney, P. D., "Distributional semantics beyond words: Supervised learning of analogy and paraphrase", *Transactions of the Association for Computational Linguistics*, Vol. 1, pp. 353–366, 2013.

7. Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", *nature*, Vol. 323, No. 6088, p. 533, 1986.

8. Mikolov, T., W.-t. Yih and G. Zweig, "Linguistic regularities in continuous space word representations", *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, pp. 746–751, 2013.

9. Zamora-Martinez, F., M. J. C. Bleda, S. E. Boquera, S. Tortajada-Velert and P. Aibar, "A Connectionist Approach to Part-Of-Speech Tagging.", *IJCCI*, pp. 421–426, 2009.

10. Perez-Ortiz, J. A. and M. L. Forcada, "Part-of-speech tagging with recurrent neural networks", *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, Vol. 3, pp. 1588–1592, IEEE, 2001.

11. Bengio, Y., R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model", *Journal of machine learning research*, Vol. 3, No. Feb, pp. 1137–1155, 2003.

12. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", *Advances in neural information processing systems*, pp. 3111–3119, 2013.

13. Pennington, J., R. Socher and C. Manning, "Glove: Global vectors for word representation", *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

14. Bojanowski, P., E. Grave, A. Joulin and T. Mikolov, "Enriching word vectors with subword information", *Transactions of the Association of Computational Linguistics (TACL)*, pp. 135–146, 2017.

15. Luong, T., R. Socher and C. Manning, "Better word representations with recursive neural networks for morphology", *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pp. 104–113, 2013.

16. Şen, M. U. and H. Erdogan, "Learning word representations for Turkish", *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 1742–1745, IEEE, 2014.

17. Güngör, O. and E. Yıldız, "Linguistic features in Turkish word representations", *Signal Processing and Communications Applications Conference (SIU), 2017 25th*, pp. 1–4, IEEE, 2017.

18. Üstün, A., M. Kurfalı and B. Can, "Characters or Morphemes: How to Represent Words?", *Proceedings of The Third Workshop on Representation Learning for NLP*, pp. 144–153, 2018.

19. Sak, H., T. Güngör and M. Saraçlar, "Turkish language resources: Morphological parser, morphological disambiguator and web corpus", *Advances in natural language processing*, pp. 417–427, Springer, 2008.

20. "Wikimedia Downloads", `https://dumps.wikimedia.org/`, accessed: 2019-06-05.

21. "WikiExtractor", `https://github.com/attardi/wikiextractor/`, accessed: 2019-06-05.

22. Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppin, "Placing search in context: The concept revisited", *ACM Transactions on information systems*, Vol. 20, No. 1, pp. 116–131, 2002.

23. Rubenstein, H. and J. B. Goodenough, "Contextual correlates of synonymy", *Communications of the ACM*, Vol. 8, No. 10, pp. 627–633, 1965.

24. Resnik, P., "Using information content to evaluate semantic similarity in a taxonomy", *arXiv preprint cmp-lg/9511007*, 1995.

25. Huang, E. H., R. Socher, C. D. Manning and A. Y. Ng, "Improving word representations via global context and multiple word prototypes", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 873–882, Association for Computational Linguistics, 2012.

26. Miller, G. A. and W. G. Charles, "Contextual correlates of semantic similarity",

Language and cognitive processes, Vol. 6, No. 1, pp. 1–28, 1991.

27. Řehůřek, R. and P. Sojka, "Software Framework for Topic Modelling with Large Corpora", *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, ELRA, Valletta, Malta, May 2010, `http://is.muni.cz/publication/884893/en`.

28. Spearman, C., "The proof and measurement of association between two things", *American Journal of Psychology*, Vol. 15, No. 1, pp. 72–101, 1904.