Intrinsic and Extrinsic Evaluation of Word Embedding Models

Gökçe Yeşiltaş[1], Tunga Güngör[1]
[1]Computer Engineering, Boğaziçi University, Istanbul, Turkey
{gokce.yesiltas,gungort}@boun.edu.tr

***Abstract*—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. We tried to improve the quality of word representations by changing the orientation of context windows. By changing the context window orientation, we aimed to train models with better accuracy results without increasing the training time. The impact of different window sizes and vector dimensions on the quality of word representations was analyzed both intrinsically and extrinsically.**

***Index Terms*—word embedding; word vector; word2vec; intrinsic evaluation; extrinsic evaluation.**

I. 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 [1], [2], [3], [4], [5].

Rumelhart *et al.* worked on one of the earliest use of word representations [6]. 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 [7]. 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 word orientation. In NLP tasks, changing context orientation is a commonly used approach. For instance, in Part-of-Speech (POS) tagging task, to find the POS tag of the current word, models may look only past context, only future context, or both past (left) and future (right) context [8] [9].

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.

II. BACKGROUND

A. Related Work in English

1) Word2Vec: In [7], Mikolov *et al.* worked on a Neural Language Model. They found out that word representations can capture syntactic regularities such as singular/plural forms of common nouns and semantic regularities such as gender or country-capital relations. The regularities are observed as constant vector offsets between pairs of words sharing a particular relationship. In [10], 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. The Skip-gram tries to predict the context words based on the current word. The Skip-gram is an efficient method because it does not require dense matrix multiplications. In [11], Mikolov *et al.* worked on Skip-gram model to improve training time and quality of vector representations. They introduced Negative Sampling (NEG) which is an approach where each training sample is used to update only a small percentage of the model's weights. Another improvement was the subsampling of frequent words since the common words like "the" are not informative. Mikolov *et al.* stated that the NEG improved representations for frequent words and the subsampling improved rare words' representations, and both approaches reduced the training time.

2) GloVe: In [12], Pennington *et al.* worked on a model that combines the advantages of two primary model families: 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

978-1-7281-9136-2/20/$31.00

large corpus like Skip-gram. They called the proposed model as GloVe, for Global Vectors.

3) fastText: Previously mentioned models 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 previously mentioned models have good representations for frequent words, such as "distinct", whereas worse representation for rare ones, such as "distinctiveness". In [13], 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. The results of the study showed that morphological information significantly improves the accuracy of syntactic tasks, whereas it does not improve the accuracy of semantic tasks.

B. Related Work in Turkish

In [14], 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. 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. Şen *et al.* prepared word analogy tasks for both semantic and syntactic regularities in Turkish. Their analogy tasks, similar with the tasks in [7], consist of questions like "what word is similar to *olay* (event) in the same sense *kelimeler* (words) is similar to *kelime* (word)?".

In [15], Güngör *et al.* aimed to explore the morphological information captured by the Turkish word representations. The 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. Results showed that even without preprocessing, word representations in Turkish can capture morphological information.

In [16], Üstün *et al.* claimed that using words as they are to learn vector representations results in inadequate representations for rare words because of the lack of statistics. They declared that using characterscould result in distant representations of semantically related words with different forms of the same morpheme.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.

III. CORPORA AND DATASETS

A. Corpora

1) Turkish Corpus: We trained Turkish word embedding models on *BounWebCorpus*. The corpus was collected using news and web pages by Sak *et al.* [17]. 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.

2) English Corpus: We trained English word embedding models on *Wikipedia dump data*. [18] The corpus was parsed by using Wikipedia Extractor that is provided by MediaLab of the University of Pisa. [19] 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.

B. Analogy Tasks

We used analogy tasks for the intrinsic evaluation of models that we trained. Analogy task sets consist of statements like that the relation between a and b is similar to the relation between c and d. We evaluated our word embedding models on: total accuracy; semantic accuracy and syntactic accuracy; and accuracy on individual analogy task categories.

1) Turkish Analogy Task: We evaluated the Turkish word embedding models on the analogy task set that was created in [14]. The analogy task set contains 10 different categories. Six of them contain 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. The analogy task set includes 15902 semantic and 10686 syntactic questions.

2) English Analogy Task: We evaluated the English models on analogy tasks that were created in [7]. 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. The analogy task set consists of 8869 semantic and 10675 syntactic questions.

C. Word Similarity Tasks

We used word similarity tasks to evaluate the quality of word embedding models extrinsically. We evaluated how the word embedding models perform in an NLP task. We used *WordSimTr* dataset that is prepared by Üstün *et al.* [16] for Turkish word embedding models. For English models, we used five different word similarity datasets: *WordSim353* [20], *RW* [21], *RG* [22], *MC* [23], and *SCWS* [24]. The similarity scores in datasets were calculated by taking the average of the scores of human annotators. Summary about word similarity datasets is in Table I.

TABLE I
SUMMARY ABOUT WORD SIMILARITY DATASETS.

Dataset	Number of word pairs	Score range
WordSimTR [16]	138	1-10
WordSim353 [20]	353	0-10
MC [23]	28	0-4
RG [22]	65	0-4
RW [21]	2034	0-10
SCWS [24]	2003	0-10

IV. METHODOLOGY

In this study, we focused on Skip-gram architecture that was proposed by Mikolov *et al.* in [11]. In the Skip-gram architecture, given a sequence of words $w_1, w_2, ...w_T$, the aim is to maximize the average log probability

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

where c is the training context size. [11] 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 [11], Mikolov *et al.* stated that larger context window could result in higher accuracy, whereas the training time will increase.

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 the context window contains words placed in only one side of the target word to improve accuracy without enlarging the window size. 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.

We trained Turkish word embedding models on *BounWebCorpus* and English word embedding models on *Wikipedia Dump Data*. We used the Gensim Python library [25] 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 changed the source code and used it to train word embedding models on modified architectures.

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.
- The minimum frequency was set to five.
- The number of iterations (epochs) over the corpus was set to five.

V. EXPERIMENTAL RESULTS

We denote the original 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 and *"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. For the same vector dimension and the same window size, we denote the best results in each category with an asterisk (*). The best results among models with the same vector dimension are denoted with two asterisks (**).

A. Intrinsic Evaluation of Word Representations

Analogy tasks contain statements like *"a is to b as c is to d"*. 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

$$y = vector(b) - vector(a) + vector(c) \quad (2)$$

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

$$D_w = \frac{x_w y}{\|x_w\|\|y\|} \quad (3)$$

where y is the vector computed in Equation 2 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

$$w^* = argmax_w(D_w) \quad (4)$$

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.

1) Evaluation of Turkish Word Representations: We report accuracy on semantic and syntactic questions in Table II with total accuracy. We observed that models trained by the original Skip-gram 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*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 II
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
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

When we look at the accuracy results on each word analogy task category, we observed that 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 outperform in all configuration settings, such as *capital* and *present tense*.

When we look at the 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. The 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.

For *present tense* analogy questions, we observed that modified models outperform the original Skip-gram models. The results show that the bigger window sizes do not improve accuracy for this type of analogy questions.

In contrast with previously shown accuracy results in *district-city* and *present tense* analogy questions, increasing the window size gives better accuracy results on *capital-country* analogy questions. The reason behind that the modified Skip-gram models perform better than the original Skip-gram models on these types of analogy questions may be that the more distant words are being used for training when we are looking to 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 better accuracy.

2) Evaluation of English Word Representations: We report accuracy on semantic and syntactic questions in Table III. Total accuracy is also represented in the table. 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. 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.

We observed the effect of the window size and vector dimension on accuracy results. Results show that both smaller window size and smaller vector dimensions give better accuracy results on *kinship* analogy questions. On *city-state* and *nationality adjective* analogy questions, results show that both bigger window size and bigger vector dimension give better accuracy results on *city-state* and *nationality adjective* analogy questions. We observed that smaller window size gives better accuracy results on *comparative* and *opposite* analogy questions, whereas a 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 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 better accuracy.

TABLE III
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*
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
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

B. 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 to evaluate the quality of word embedding models that we trained. We used Spearman's rank correlation [26] 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.

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 IV. We observed that the modified models have better results in most cases. When the window size is set to 4 or 6, we observed the best results among the models with the same vector dimension.

2) Evaluation of English Word Representations: We used 5 different word similarity test sets to evaluate how the models perform on word similarity tasks.

We observed that the original Skip-gram models give better results on *RW* and *SCWS* test sets. Additionally, as the vector dimension increases, results are getting better.

When we look at the Spearman's rank correlation results on *WordSim353*, we observed that an increase in window size and vector dimension results in a better correlation score. Where the 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. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 300 are shown in Figure 1.

There were no big differences between correlation scores of different word embedding models on *RG*. There is not a better model architecture or window size among the configuration settings that we used for training models. The only observation is that the bigger vector dimension gives better correlation results on this test set.

We observed that an increase in window size and vector dimension does not result in a better correlation score on *MC*. The modified word embedding models achieve the best results for all vector dimensions that we trained our models with.

VI. CONCLUSION

In this study, word embedding models on Turkish and English were trained with different configurations. We focused

TABLE IV
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*

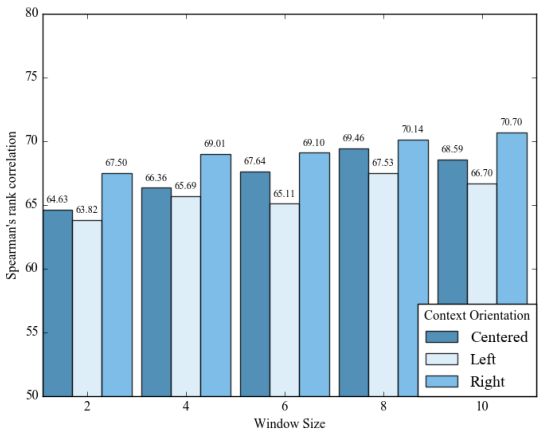

Fig. 1. Spearman's rank correlation ($\rho \times 100$) on WordSim353 for models where vector dimension was set to 300 are shown in the graphic.

on the word2vec methodology and tried to improve quality by changing the orientation of context windows. By changing the context window orientation, we aimed to train models with better accuracy results without increasing the training time.

The 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 they work 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, they 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 they work 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.

REFERENCES

[1] Y. Bengio, 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.

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

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

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

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

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[7] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 746–751.

[8] F. Zamora-Martinez, M. J. C. Bleda, S. E. Boquera, S. Tortajada-Velert, and P. Aibar, "A connectionist approach to part-of-speech tagging." in *IJCCI*, 2009, pp. 421–426.

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

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

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

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

[13] P. Bojanowski, 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.

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

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

[16] A. Üstün, M. Kurfalı, and B. Can, "Characters or morphemes: How to represent words?" in *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018, pp. 144–153.

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

[18] *Wikimedia Downloads*, accessed in June 2019. [Online]. Available: https://dumps.wikimedia.org/

[19] *WikiExtractor*, accessed in June 2019. [Online]. Available: https://github.com/attardi/wikiextractor/

[20] L. Finkelstein, 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.

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

[22] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, vol. 8, no. 10, pp. 627–633, 1965.

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

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

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

[26] C. Spearman, "The proof and measurement of association between two things," *American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.