Uludağ University Journal of The Faculty of Engineering, Vol. 23, No. 1, 2018 RESEARCH

DOI: 10.17482/uumfd.318615

CO-OCCURRENCE WEIGHT SELECTION FOR WORD EMBEDDINGS TO ENHANCE TEST PERFORMANCE

Aykut KOÇ *

Veysel YÜCESOY *

Received: 05.06.2017 ; revised: 26.12.2017 ; accepted: 07.02.2018

Abstract: This study revisits the problem of maximizing the performance of mathematical word representations for a given task. It is aimed to improve performance in analogy and similarity tasks by suggesting innovative weights instead of the counting weights used conventionally in counting-based methods of generating word representations (adding the statistics of word co-occurrences to the account). The language of study was selected as Turkish. The root structures of Turkish words were managed during the compilation of corpus such that each word having a suffix was considered as a new word. The performance of the proposed co-occurrence weights are analyzed with respect to the varying parameter and the results are presented within the paper.

Keywords: Word embeddings, Natural language processing, Statistical linguistics

Kelime Temsilleri için Test Performansını Geliştirmeye Yönelik Eşdizimlilik Ağırlıklarının Seçimi

Öz: Bu çalışma, matematiksel kelime temsillerinin belirli bir görev için performanslarının en iyilenmesi problemini yeniden ele almaktadır. Sayma tabanlı (kelimelerin eşdizimlilik istatistiklerini hesaba katan) kelime temsili oluşturma yöntemlerinde klasik olarak kullanılan sayma ağırlıkları yerine yenilikçi ağırlıklar önererek analoji ve benzerlik bulma görevlerinde performans artışı sağlamak hedeflenmektedir. Çalışma dili olarak Türkçe seçilmiş, derlem oluşturulurken Türkçe'ye has ek-kök yapıları ek alan her kelime yeni bir kelime gibi kabul edilecek şekilde yorumlanmıştır. Önerilen eşdizimlilik ağırlıklarının performansı değişen parametreye göre analiz edilerek sonuçlar çalışma içerisinde paylaşılmıştır.

Anahtar Kelimeler: Kelime temsilleri, Doğal dil işleme, İstatistiksel dilbilimi

1. INTRODUCTION

Natural language processing (NLP) is a subfield of computer science in which computers are used for interpreting and processing the natural languages. As a special application, NLP methods can extract relevant information from a piece of text. Following the developments in internet access and smart devices, especially in the last decade, social media (e.g. Facebook, Twitter, Instagram) has become greatly popular. In these social media platforms, people generally express their opinions or feelings by writing short pieces of texts. Triggered by the desire to classify these texts automatically and by the abundance of applications, natural language processing has increased its popularity as a research area.

Besides these developments, improvements in deep learning and GPU technologies yield very satisfactory results in image processing applications. Especially in image classification, convolutional neural network based approaches outperformed classical hand crafted features, see (Krizhevsky, et. al. 2012). By the inspiration of these achievements, some research on text

* ASELSAN Research Center, Ankara 06370, Turkey
Correspondence Author: Aykut Koç (aykutkoc@aselsan.com.tr)

classification by deep neural networks emerged (Bengio, et. al. 2003), (Bahdanau, et. al. 2014). Especially after the introduction of recurrent neural networks, deep learning seem to be a promising tool for text classification. All these methods require digitized and mathematical representations of words by keeping their meanings and relations among their meanings within these mathematical representations.

Unlike image and sound processing, text processing does not have a data of predetermined size. For example, any image has a predetermined size of the recording sensor whereas any sound is sampled by the sampling frequency of the recording microphone. Images and sounds are suitable to be analyzed by mathematical tools (e.g. linear transforms, rotations, scaling, resampling, etc.) as a result of having predetermined structure. However, since none of the basic elements of a text (i.e. word, sentence, paragraph, etc.) can have a predetermined size or structure, it is not straightforward to use mathematical analysis on text.

The idea of word representations was proposed to solve this inconvenience. A word representation, or a Vector Space Model which dates back to Salton, et. al. (1975), can be defined to be a vector of real variables in a high dimensional space. The simplest example of such a space model is to form a binary vector for each word which has the same size as the total vocabulary and is zero everywhere except at the index of the relevant word as shown in Figure 1.

$$\begin{array}{lllllll} \text{Word 1}: & [\,1 & 0 & \ldots & 0 &] \in & \mathbb{R}^{1\times n} \\ \text{Word 2}: & [\,0 & 1 & \ldots & 0 &] \in & \mathbb{R}^{1\times n} \\ \quad\vdots & & & & & & \\ \text{Word n}: & [\,0 & 0 & \ldots & 1 &] \in & \mathbb{R}^{1\times n} \end{array}$$

Figure 1 Simplest binary word representation

It is important to note that although the aforementioned word representation is suitable for mathematical analysis, it is not a good idea to use such representations as a starting point since the mapping includes no implicit relations between words.

In the literature, there have been some strong efforts to find dense word representations, named a word embedding. Some initial efforts tried to analyze the co-occurrence matrix mathematically. LSA and randomized embedding from Ravichandran, et. al. (2005) are the well-known examples of such efforts. Co-occurrence matrix is the one which holds the weighed number of co-occurrences of each word pair within a predetermined window throughout the corpus. These initial methods were successful in similarity tasks, however, they found to be unsatisfying in analogy tasks.

A similarity task aims to find the similar subgroup within a group of words. In other words, it tries to detect the odd one out. When each word is simply described by a vector of real variables, the similarity task is just to find the furthest vector in a group of vectors to the average vector of the group. With this point of view, a similarity question is a multiple choice question because the answer is in the group of words given in the question. An example of a similarity task question is given in Figure 2. Within the group of *apple*, *orange*, *banana* and *table*, first three words form a group of fruits and *table* is the odd one out and hence the answer of the question. On the other hand, analogy task aims to find a word which has a relation with the word of question and the relation is implicitly defined by a pair of other words. For example, analogy question comes as "which word has the relation with *Spain* as *London* and *England* have in-between". For a human, the answer (which is *Madrid*) is easy provided that the meanings of the words are clear, however, from the point of view of word embedding, it is an open ended question and any word in the vocabulary is a candidate answer. Another important difference between similarity and analogy questions is that the answer of the analogy question is related to alignment of word pairs (i.e. difference vectors of words), whereas the former one only depends on the relative distance of word pairs. These relations between word pairs are depicted in Figure 2.

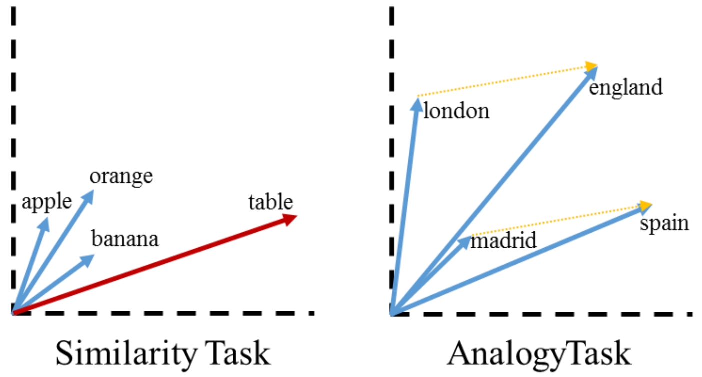

Figure 2 Representational difference between similarity task and analogy task

There have been different efforts in the literature that aimed to optimize the word embeddings for different tasks (Mnih and Hinton, 2007), (Socher, et. al. 2013), (Luong, et. al. 2013), (Le and Mikolov, 2014), (Faruqui, et. al. 2014). The most recent and powerful methods to generate word embedding in order to succeed in analogy tasks are word2vec from Mikolov, et. al. (2013a), Mikolov, et. al. (2013c), GLoVe from Pennington, et. al. (2014) and fastText from Bojanowski, et. al. (2016). The last one can be seemed as an extension of word2vec in order to account for the lexical similarity of the words. The most powerful feature of these methods is to operate on unlabeled corpus, which means they can be trained by unsupervised learning. That is why it is easy to find and download large corpora to operate on by these methods.

Despite being different approaches, both word2vec and GLoVe assume that the nearest words in a corpus are related to each other more than distant ones. In other words, the semantic relation weakens as the distance between two words increases. The linguistic roots of this approach goes back to the theory of renowned linguist J. R. Firth reflected by his famous quote: "*you shall know a word by the company it keeps*" in (Firth, 1957). Almost all modern word representation techniques are just some kind of mathematical implementations of this linguistic approach by the use of co-occurrences of words within the same context. That is also why all the co-occurrence based embedding methods weight the co-occurrences by a weighting function (mostly decreasing). Current weighting functions uses a simple discount method: the more distant two words are, the more their co-occurrence count is discounted. However, this approach does not always grasp the semantic relation of two words if this relation does not pronounce itself by the words co-occurrence statistics.

In the literature, there have been a large number of studies making use of the word embedding idea for image captioning Karpathy and Fei-Fei, (2015), neuroscientific research Huth, et. al. (2016), cross lingual relations Şenel, et. al. (2017a), automatic translation Mikolov, et. al. (2013b) and text classification Tang, et. al. (2014).

There exists an additional field of study that is called Language Models, in which the aim is to estimate the relative likelihood of words or phrases following other words or phrases. Some of the well-known recent examples of these models are Bengio, et. al. 2003, and Mikolov, et. al. 2010. Similar to generation of word embeddings, these models utilize the co-occurrence statistics of words in order to estimate the conditional probabilities of words given a prior word or phrase. Although the underlying statistics is similar, such models are out of the scope of this study since we just try to model the semantic and syntactic relations between words by dense embeddings. We only make use of the spatial occurrences of words, we do not generate a tool to estimate them in generating artificial sentences.

In this paper, we propose a new weighting function in order to enhance the analogy task success of the resulting vectors. The aim of the proposed weights is to emphasize the semantic relation between words appearing in the same context but not appearing near enough. A typical example about this situation is the "carpenter" and "hammer" example. Although these two words are semantically closely related, they are usually accompanied by some other words in-between. Some example usages are "… carpenter uses a black hammer…" or "… carpenter nails the wood by a hammer …". In order to get rid of this effect, a parametrically defined weighting function is proposed and numerical optimization will be conducted to optimize the parameter. We have a similar work for English and it was published in Yücesoy and Koç, (2017) where the proposed weights were shown to outperform the classical weight in English analogy task.

The paper is organized as follows: Section 2 introduces the proposed weights and compares it with the state of the art weights. Section 3 explains how the optimization of the parameter and simulations are conducted and represents the success rates of the proposed weights with respect to the baseline success rate. Section 4 concludes the paper with some remarks.

2. State of the Art and Proposed Weights

GloVe is a count based word embedding generation method. It is a count based method because it uses weighted co-occurrence data to optimize word vectors for a given goal. See Pennington, et. al. (2014) for a detailed explanation of the algorithm. While counting co-occurrences, a window size is required to accept the words being co-occurred if a pair is contained within the specified window. In addition, according to a weighting function of the form shown in Figure 3, the co-occurrence data is weighted by the assumption that nearest word pairs share more semantic relation than far apart pairs.

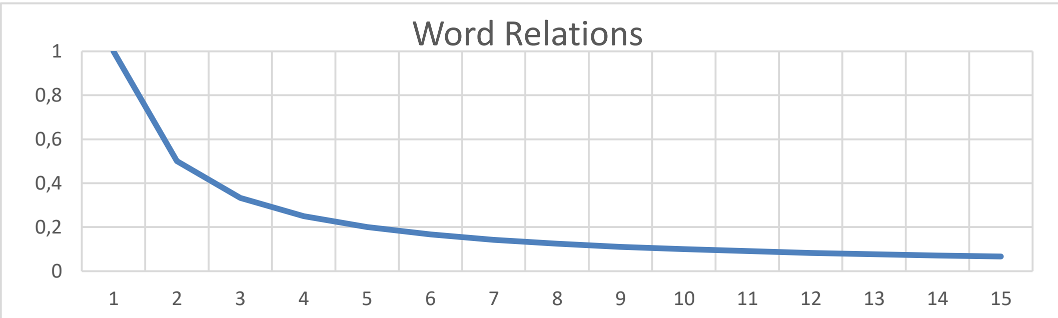

Figure 3 Example co-occurrence weight from original GloVe algorithm (Pennington, et. al. 2014). For this specific example window size is taken to be 15.

There exist no theoretical insight that the weighting function in Figure 3 will fit the requirements of any task, however, it achieves a satisfying success rate for English analogy test set. To the best of our knowledge, still the question of better weighting functions for different tasks or for entirely different languages is open. The aim of this paper is to suggest novel weighting functions for Turkish language in order to increase the success rate of word embedding both in similarity and analogy tasks.

The mathematical representation of the original weighting function from Pennington, et. al. (2014) is as follows:

$$c_{ij} = \frac{1}{x_{ij}}$$

where $\boldsymbol{x_{ij}}$ is the relative distance between words $\boldsymbol{i}$ and $\boldsymbol{j}$. An element of the co-occurrence matrix M is calculated by the formula

$$\boldsymbol{M_{ij}} = \sum_{\substack{all\ co-occurrences \\ of\ word\ i\ and\ j\ within \\ predefined\ window\ size}} \boldsymbol{c_{ij}}$$

It is clear from the expression that the weighting function is of fundamental importance for co-occurrence statistics. Knowing this, we tried to find a simple one parameter family of functions in order to conduct a search over the parameter to see if it is possible to beat the original weighting function in terms of analogy and similarity test results.

In order to reason about the shape of the new weighting function, let us consider the example given in introduction. Although carpenter and hammer are semantically related words, they generally appear far from each other in text. In addition to this, it is a well-known fact that the Turkish sentences have the most semantically rich words at the end. To overcome the aforementioned disadvantage and make use of the semantic property of the Turkish sentences, we need to favor some distant words more than the original weighting function. From this point of view, a simple one parameter family of function can be defined as follows:

$$\boldsymbol{\hat{c}_{ij}} = \boldsymbol{a} + \frac{\boldsymbol{b}}{\boldsymbol{x_{ij}}}$$

where $a, b > 0$ and $a + b = 1$ are also constrained. With these extra constraints, $\hat{c}_{ij}$ becomes a one parameter weight. The tail of the weighting function goes up as a increases as shown in Figure 4. Final expression for the proposed weighting function is given as

$$\boldsymbol{\hat{c}_{ij}} = \boldsymbol{a} + \frac{\boldsymbol{1-a}}{\boldsymbol{x_{ij}}} \qquad \text{for} \qquad \boldsymbol{0} < a < 1$$

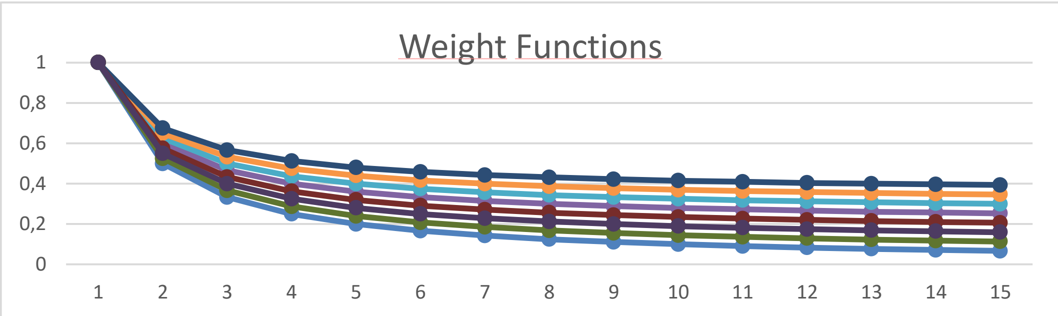

Figure 4 First alternative weighting functions for word embedding. Due to scaling effect on the tail of the function this approach is named as scaled weights. The functions shown in the figure are parametrized by one variable, so it is convenient to make a search over that parameter.

3. Simulations and Tests

Turkish Wikipedia[1] pages were utilized as the main corpus for this study. A python code[2] was used to clear and store the data as text from downloaded Wikidump. This code produces a bunch of files in xml format. All those files were post-processed and combined together to form a one line corpus which is suitable for GloVe processing. Post processing steps include lowercasing, removal of non-alphabetic characters, removal of headings, removal of apostrophes and separation of words which include apostrophes into two words. After all these steps, final corpus reached approximately 52M tokens.

[1] https://www.wikipedia.org
[2] http://github.com/attardi/wikiextractor

In Şenel, et. al. (2017b) we have defined analogy and similarity test sets for Turkish. These sets are used to measure the performance of the word vectors. Five of the analogy test sets were used throughout this paper: Common capitals, world capitals, nationality-language, country-language and country nationality. Apart from these, a similarity test set which is composed of 189k questions was also used in the simulations. All the parameters which are used to generate the GloVe vectors are given in Table 1. Most of the parameters are used as the suggested values from Pennington, et. al. (2014) and vocabulary threshold was chosen to have a vocabulary size of 125k approximately.

Table 1 Parameters used for learning word vectors with GloVe (Pennington, et. al. 2014).

Parameter Name	VECTOR_SIZE	MAX_ITER	WINDOW_SIZE	X_MAX	VOCAB_MIN_COUNT
Value	200	20	15	10	20

Table 2 gives all the results for all test sets. First column of the table gives the baseline results for original GloVe algorithm. For each row in the table, green cells show the maximum available performance within the row. Blue cells are the second best scores and yellow ones are the third top performer of the rows. As it is clear from the table, proposed weights have the best performance for all analogy test sets when the proposed parameter is between 0.05 and 0.20. For the similarity test set, the best score is attained when the proposed parameter is 0.80.

Table 2 Simulation results for GloVe with the modified weights proposed in this paper. The proposed weights are parametrized by a parameter and a column of the table gives the performance of the final vectors for a value of the parameter. The first column is the baseline performance of GloVe over the Turkish test sets defined in Şenel, et. al. (2017b). The performance of a vector space is calculated as the ratio of the correctly answered questions to the total number of questions in a given set.

	Original	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45
Common Capital	83.14	83.14	83.71	**83.90**	83.14	80.49	79.92	81.06	80.78	81.06
World Capital	62.48	64.43	66.34	67.33	**67.87**	67.41	65.96	66.96	67.66	67.78
Nationality-Language	73.68	**77.37**	75.26	75.79	76.32	76.84	75.79	74.21	68.42	75.79
Country-Language	91.43	91.43	92.86	90.00	**95.24**	93.33	89.52	88.10	89.05	87.62
Country-Nationality	66.17	**68.92**	68.29	67.76	68.39	66.28	67.55	66.91	65.86	64.59
Similarity Test	85.58	86.66	86.88	86.87	87.15	87.64	87.66	88.48	88.56	88.55
	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
Common Capital	80.87	79.92	78.41	79.55	78.03	78.03	78.98	79.17	78.22	78.79
World Capital	67.04	66.96	66.54	65.18	64.80	64.18	63.60	63.35	63.35	63.44
Nationality-Language	71.58	70.53	70.53	68.95	68.95	67.89	64.74	66.32	61.58	65.79
Country-Language	86.67	84.76	86.67	86.67	86.19	85.71	84.29	87.14	84.76	86.19
Country-Nationality	63.53	63.85	63.42	64.06	62.79	62.58	62.37	60.25	60.99	60.57
Similarity Test	88.78	89.05	88.77	88.54	88.80	88.91	**89.26**	89.25	89.25	88.85

Figures 5 and 6 show the overall performance graph of the proposed weights in analogy test set and similarity test set, respectively. In order to calculate the overall analogy success of the proposed weights, success rates of five analogy test sets are combined according to their number of questions. Number of questions for each test set is given in Table 3. As it is obvious from the figures, proposed weights outperform the classical weights approximately by 5.66% in analogy task (when a = 0.2) and by 4.3% in similarity task (when a = 0.8) at their maximum performance. It is also interesting to note that as the parameter "a" increases (i.e. as the curve in Figure 4 becomes flatter), the analogy performance goes below the benchmark performance whereas the similarity performance increases. This behavior is due to the different natures of analogy and similarity tasks. As the parameter “a” increases, all the words within the predetermined window share a relatively similar weight for the embedding of the center word. Such equal sharing is suited for similarity test because the test just aims to determine the word that do not co-occur with the rest of the group. However, analogy task aims to find a specific word having a predetermined relation with the query word. For this task, relatively equal weight sharing degrades the performance. With these interpretations, we can understand the behaviors of the curves in Figures 5 and 6 as the parameter “a” increases.

Table 3 Total number of questions for each test set

Name of Set	Common Capital	World Capital	Nationality Language	Country Language	Country Nationality	Similarity Test
# of questions	528	3081	231	231	1081	189000

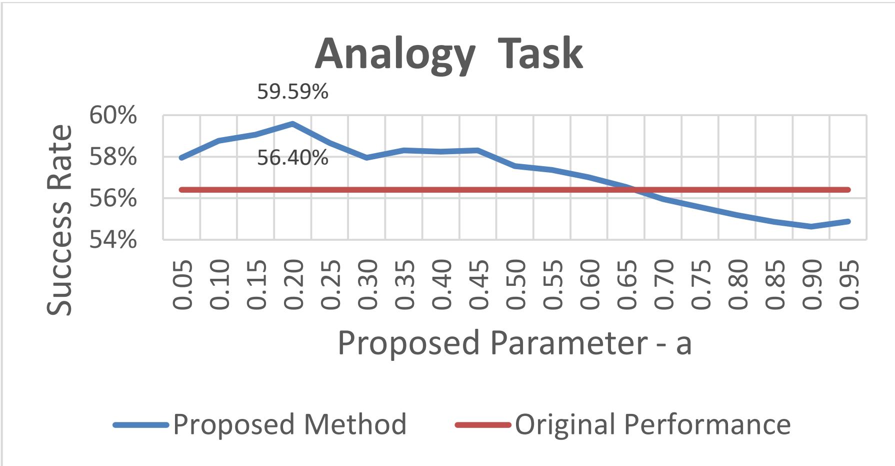

Figure 5 Results obtained by using the proposed weights in analogy test set. Horizontal axis represents the parameter of the proposed weights. As it is clear from the figure, it is possible to obtain approximately 5.66% increase in the success rate by utilizing new weights (for a = 0.2). It is also good to observe that the performance of the proposed weights outperforms the original performance for a ϵ [0.05 – 0.65].

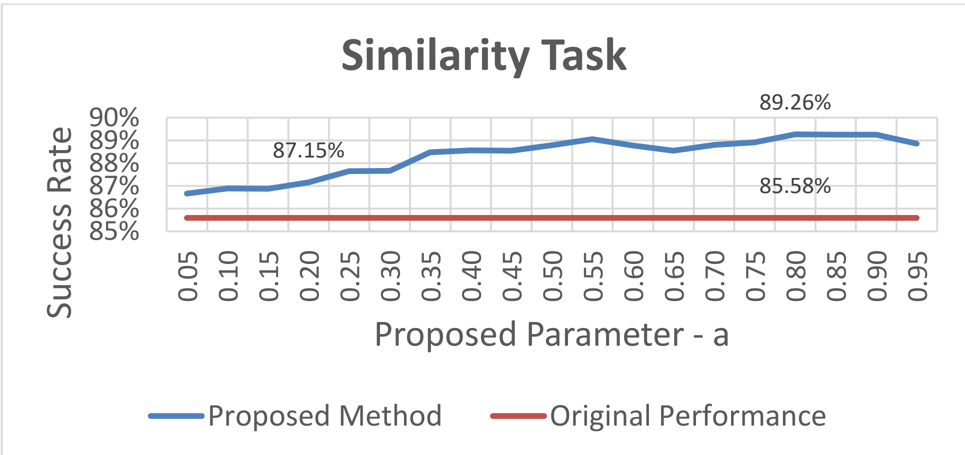

Figure 6 Results obtained by using the proposed weights in similarity test set. Horizontal axis represents the optimization parameter. As it is seen, it is possible to obtain approximately 4.3% increase in the

success rate by utilizing new weights (for a = 0.8). It is also good to observe that the performance of the proposed weights outperforms the original performance for all a ϵ [0.05 – 0.95].

4. Conclusion and Discussion

The problem of performance enhancement of word embedding for a given task is revisited. Analogy and similarity task performances are considered for this study. Count based word embedding generation methods are considered and an improvement for the calculation of co-occurrence statistics is proposed. Classically, the co-occurrence of a pair of word in a predetermined window is weighted by a decaying first order power series with respect to the relative position of the words. A new one-parameter weighting function is proposed and its effect on the analogy and similarity test sets is analyzed.

For different optimum points of the parameter, it is possible to achieve 5.66% increase (a = 0.2) in analogy test success and 4.3% increase (a = 0.8) in similarity test success. If we stick to the optimum point at which the analogy performance is at its best (i.e. a = 0.2), performance of the similarity task increase by approximately 1.83%. Hence, for the parameter a = 0.2, the performance of the word embedding on both test sets increases significantly.

It is also important to note that for similarity test set, the success of the proposed weight is always higher than the baseline performance (i.e. for all a = [0.05 – 0.95]) and for analogy test set it is higher for a = [0.05 – 0.65]. This shows us that the proposed weight is capable of capturing the semantic relations between Turkish words better than the original one for a large range of its parameter. This result shows that the words that do not co-occur in a small neighborhood in Turkish corpus should not be penalized harshly due to the semantic properties of Turkish. Since the proposed weight favors the distant relations more than the original weight, it is more suitable for Turkish semantic structure. Indeed, we have a similar study for English in Yücesoy and Koç, (2017) and that study revealed that the performance of the proposed weight increases the performance of the word embedding for analogy test by approximately 2%. When these two results are considered together, it might be possible to conclude that the proposed weight is more suitable to Turkish semantic relations than the English relations.

The design of the optimal weight for a specific task is still an open problem for a given language. Since the semantic structure of the languages differ from each other, weighting should be designed language specific for any desired task. This study is a showcase to illustrate that there could be optimized weight for each specific task which performs better than the original weight proposed in Pennington, et. al. (2014) for count based word embedding generation algorithms. We examined a simple one-parameter family of weight for Turkish similarity and analogy tasks and outperformed the original weight in both tasks. Some more complicated family of weights (i.e. polynomial weights, rational weights, etc.) can be considered to include the semantic properties of the language more into the word embedding.

REFERENCES

1. Bahdanau, D., Cho, K. and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv: 1409.0473.

2. Bengio, Y., Ducharme, R., Vincent, P., and Jauvin C. (2003). A neural probabilistic language model. Journal of machine learning research, 1137 – 1155. doi: 10.1162/153244303322533223

3. Bojanowski, P., Grave, E., Joulin, A., ve Mikolov, T. (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

4. Faruqui, M., Dodge, J. , Jauhar, S. K., Dyer, C., Hovy, E. ve Smith, N. A. (2014) Retrofitting word vectors to semantic lexicons, arXiv preprint arXiv:1411.4166. doi: 10.3115/v1/N15-1184

5. Firth, J. R., (1957). A synopsis of linguistic theory 1930-1955. In Studies in linguistic analysis, 1-32. Oxford:Blackwell.

6. Huth, A.G., de Heer, W.A., Griffiths, T.L., Theunissen, F.E. and Gallant, J.L. (2016) Natural speech reveals the semantic map that tile human cerebral cortex. Nature, vol. 532, no. 7600, 453 – 458. doi:10.1038/nature17637

7. Karpathy, A. and Fei-Fei, L. (2016). Deep visual-semantic alignments for generating image descriptions. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39 (4), 664-676. doi: 10.1109/TPAMI.2016.2598339

8. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural netwroks. In Advances in neural information processing systems, 1097-1105. doi: 10.1145/3065386

9. Le, Q. V. ve Mikolov, T. (2014) Distributed representations of sentences and documents, ICML, vol. 14, 1188–1196.

10. Luong, T., Socher, R. ve Manning, C.D. (2013) Better word representations with recursive neural networks for morphology, CoNLL, 104–113.

11. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J. and Khudanpur S. (2010) Recurrent neural network based language model, Interspeech, Vol 2, 3.

12. Mikolov, T., Chen, K., Corrado, G. ve Dean, J. (2013a) Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

13. Mikolov, T., Le, Q.V. and Sutskever, I. (2013b) Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.

14. Mikolov, T., Sutskever, I, Chen, K., Corrado, G.S. ve Dean J. (2013c) Distributed representations of words and phrases and their compositionality, Advances in neural information processing systems, 3111–3119.

15. Mnih, A., ve Hinton, G., (2007) Three new graphical models for statistical language modelling, Proceedings of the 24th International Conference on Machine Learning. ACM, 641–648. doi:10.1145/1273496.1273577

16. Pennington, J., Socher, R. ve Manning, C.D. (2014) Glove: Global vectors for word representation, Empirical Methods in Natural Language Processing (EMNLP), 1532–1543. doi: 10.3115/v1/D14-1162

17. Ravichandran, D., Pantel, P. ve Hovy, E. (2005) Randomized algorithms and nlp: Using locality sensitive hash function for high speed noun clustering, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 622–629. doi: 10.3115/1219840.1219917

18. Salton, G., Wong, A., and Yang, C.S. (1975) A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620. doi: 10.1145/361219.361220

19. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., ve Potts C., (2013) Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 1631-1642.

20. Şenel, L.K., Yücesoy, V., Koç, A. and Çukur, T. (2017) Measuring cross-lingual semantic similarity across European languages. In International Conference on Telecommunications and Signal Processing (TSP), 359-363. doi: 10.1109/TSP.2017.8076005

21. Şenel, L. K., Yücesoy, V., Koç, A., Çukur, T. (2017b). Semantic similarity between Turkish and european languages using word embeddings. 25th Signal Processing and Communications Applications Conference, 1-4. doi: 10.1109/SIU.2017.7960365

22. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T. and Qin, B. (2014) Learning sentiment specific word embedding for twitter sentiment classification. ACL, 1555 – 1565. doi: 10.3115/v1/P14-1146

23. Yücesoy, V., Koç A. (2017). Effect of co-occurrence weighting to English word embeddings. 25th Signal Processing and Communications Applications Conference, 1-4. doi: 10.1109/SIU.2017.7960385