Towards Turkish Word Embeddings: An Intrinsic Evaluation

Oğuz Ali Arslan
Department of Artificial Intelligence & Data Engineering
Istanbul Technical University
İstanbul, Türkiye
arslanog20@itu.edu.tr

Berfin Duman
Department of Electronics and Communication Engineering
Istanbul Technical University
İstanbul, Türkiye
dumanb19@itu.edu.tr

Hakan Erdem
Department of Artificial Intelligence & Data Engineering
Istanbul Technical University
İstanbul, Türkiye
erdemh20@itu.edu.tr

Can Günyel
Department of Computer Engineering
Istanbul Technical University
İstanbul, Türkiye
gunyel20@itu.edu.tr

Bike Sönmez
Department of Mathematics
Istanbul Technical University
İstanbul, Türkiye
sonmezb19@itu.edu.tr

Doğukan Arslan
Department of Artificial Intelligence & Data Engineering
Istanbul Technical University
İstanbul, Türkiye
arslan.dogukan@itu.edu.tr

2023 8th International Conference on Computer Science and Engineering (UBMK) | 979-8-3503-4081-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/UBMK59864.2023.10286768

***Abstract*—Effective representation of textual data is a prerequisite for most of the downstream tasks, which increases the importance of word embedding evaluation methods. The intrinsic approach assesses the similarity between word representations and human judgements. In this paper, we present a comprehensive intrinsic evaluation of Turkish word embedding models with different tasks using task-specific datasets such as SemEval-2017, MC-30, SimVerb-3500 for word similarity, MSR for word analogy and methods that have not been tested for Turkish before such as oncept categorization with BLESS and ESSLLI and outlier detection with 8-8-8 Dataset. While each of these datasets were originally in English, we translated them into Turkish and trained Wor2Vec, FastText and Glove language models with these datasets from scratch. The results suggest that while Word2Vec is generally more successful in word similarity and outlier detection tasks, fastText outperforms other models in word analogy and concept categorization.**

Keywords*—*word embedding evaluation; intrinsic evaluation; Turkish word embeddings

I. Introduction

In the field of Natural Language Processing (NLP), the successful representation of text data is a crucial requirement to unlock the potential of several subtasks. Numerous studies have been conducted in the literature to develop and evaluate word embeddings, which are mathematical representations of text data using numerical vectors that exploit grammatical and statistical features to capture semantic relationships and similarities between words.

In evaluating embedding algorithms, several evaluation methods have been developed. Intrinsic evaluation and extrinsic evaluation stand out as the two main categorization. Intrinsic evaluation has been studied for the languages like English with a broader scope, whereas low-resource languages such as Turkish have received little or no attention.

The main objective of this study is to compare the performance of three commonly used word embedding models, Word2Vec [1], GloVe [2], and fastText [3], with comprehensive intrinsic evaluation methods for Turkish, including tasks tested for the first time, to the best of our knowledge. In addition, we present an extensive test dataset consisting of Turkish data for various intrinsic evaluation tasks, translated from English. All code and data[1] used in this study is openly available for further research and reproducibility purposes.

The paper is organized as follows: In Section II, previous work on word embedding evaluation is briefly reviewed. Evaluated models, used datasets and the intrinsic evaluation methods are described in Section III. In Section IV, results of the experiments are presented. Finally, the paper concludes with a summary of the main findings and future suggestions in Section V.

II. Related Work

The fact that how well word vectors represent words directly affects the performance of downstream tasks has led to the development and importance of research on the evaluation of these vectors. A wide variety of evaluation methods have been proposed in this direction. In general, there are two types of methods: intrinsic and extrinsic. Extrinsic evaluation measures the success of word vectors by using them directly in a downstream task. This type of evaluation might be insufficient if one wants to use the word representation algorithm in a wide

[1]https://github.com/swarm-nlp/TRIntrinsicEval

variety of tasks. To compensate for this, intrinsic evaluation methods have been proposed.

In intrinsic methods, the idea is to compare word representations with human judgments. An example of an intrinsic evaluation is the word similarity task, which aims to find the similarity of a pair of words. For example, [4] defines the task in a multilingual and cross-lingual perspective and states that the most accurate results can be obtained by combining lexical knowledge with word embeddings. Intrinsic evaluation can also be applied through the word analogy task. [5] uses word analogy to explore the effectiveness of simple vector subtraction in capturing different lexical relations in word embeddings. Concept categorization can also be used for intrinsic evaluation, as in [6]. The authors treat the task as an unsupervised clustering task and evaluate clusters based on a metric called purity. Another approach to intrinsic evaluation is outlier detection. [7] defines the task and provides a framework for testing models' ability to produce well-defined clusters in a vector space.

Several papers have proposed and utilized intrinsic evaluation methods in the context of the Turkish language. In [8], an intrinsic evaluation dataset is introduced, specifically designed to assess various semantic relations beyond the traditional ones like synonyms, antonyms, hypernyms, meronyms, and morphological relations. [9] presents the AnlamVer dataset, which focuses on word similarity and relatedness evaluation in Turkish, aiming to balance words and word-pairs using multiple morphological and semantic attributes. Additionally, [10] conducts bilingual dictionary induction as an intrinsic task to evaluate the performance of Turkic languages, comparing different techniques such as MUSE, VecMap, and Meemi. Furthermore, [11] utilizes Turkish intrinsic evaluation to explore the joint training of geo-word embeddings in two languages, examining the proximity of resulting translation pairs in vector space. Finally, [12] discusses an intrinsic evaluation approach involving the assessment of machine-generated summaries based on their correspondence with human-generated summaries. These papers contribute to the development and application of intrinsic evaluation techniques for various aspects of the Turkish language.

III. Methodology

In this study, the three language models described below were trained on the Turkish Wikipedia dump data [13], which was created with Turkish wiki pages taken in June 2023 and evaluated on four intrinsic evaluation tasks. The dataset consists of a total word count of 70,278,107, encompassing 822,414 articles and 8,334,139 sentences, with an average word length of 6.

A. Models

1) Word2Vec: One of the most widely used algorithms for representing words as vectors for various downstream NLP tasks is called Word2Vec. It uses dense vectors to encapsulate words' meaning and structure in a high-dimensional space. Word2Vec can effectively capture both semantic and syntactic correlations by considering the context in which words occur.

Word2Vec includes two distinct architectures: Skip-Gram and Continuous Bag-of-Words (CBOW). The Skip-Gram model focuses on predicting context words given a target word, whereas the CBOW model predicts the target word based on the surrounding context words.

We used four different configurations for Word2Vec. Two of them are models using default parameters of the Gensim [14] framework, which parameters are 100 for vector size parameter, 5 for window parameter, and 5 for epoch parameter for the CBOW and Skip-Gram architectures. The other two are the same architectures trained with 300 for vector size parameter, 8 for window parameter to grasp and understand the context much broader, and 10 for epoch parameter to let the model grasp and learn relationships in the data. These models will be named Word2Vec$_{custom_cbow}$ and Word2Vec$_{custom_skipgram}$ from now on.

2) GloVe: Another algorithm used to obtain word representations is called GloVe (Global Vectors). While it successfully captures both semantic and syntactic relationships between words in a high-dimensional vector space, it is distinguished by its use of global statistical information, which includes consideration of word co-occurrence patterns on a global scale.

We have used two different configurations for GloVe. One of them is a model with default parameters of GloVe [15] library, which are 50 for vector size parameter, 15 for window size parameter, and 15 for epoch max iteration. The other one is trained with 75 for vector size parameter, 20 for window size parameter, and 10 for max iteration parameter. This model will be named $GloVe_{custom}$ from now on.

3) fastText: In addition to the previously discussed GloVe and Word2Vec algorithms, another notable approach to obtaining word representations is fastText. It differs from Word2Vec and GloVe by utilizing subword information to effectively capture both semantic and syntactic relationships within a high-dimensional vector space. Fasttext model also has CBOW and SkipGram architectures.

We have used three different configurations for fastText. Two of them are models using default parameters of fasttext library which created by Meta [16], which parameters are 1 for word n-grams, 5 for minCount and 5 for epoch parameter, for the CBOW and Skip-gram architectures. The other one is the trained version of the Skip-Gram architectures with 2 for word n-grams thus to capture the semantic dependency in the dataset, 10 for epoch parameter in order to learn better dataset and 1 for the minCount. This model will be named $Fasttext_{custom}$ from now on.

While adjusting hyperparameters of the models, we increased the number of epochs and maximum iterations to

minimize the loss function, considering the fact that our data is sufficiently big to potentially be overfitted with these additional epochs. Additionally, in GloVe and Word2Vec, we increased the window and vector sizes to expand the contextual grasp and obtain more detailed descriptions of words in the vector space. Finally in FastText we increased the wordNgram size to again expand the contextual understanding.

B. Dataset Translation

In order to test the trained models, the datasets compiled for each task below were translated into Turkish using Google Translate and checked by two native Turkish speakers. In the translation process some words were contextually incorrect due to their synonymity condition or they were completely mistranslated. This situation was corrected by native Turkish speakers and the words were translated in accordance with their context.

For instance, in the comparison of the words "window blind" and "curtain" in the SemEval dataset we used in the word similarity task, Google Translate translated the phrase "window blind" as "*kör pencere*" which literaly means "blind window" in Turkish. Again on the same dataset, as a comparison of two different journal names, the words "Guardian" and "Times" have been translated as if they are used in the first sense of the word. Both cases have been corrected to fit its' context.

Second, our models represent individual words in the Wikipedia dump dataset, so when translated into Turkish we must discard words that are represented by more than one word representation in Turksih. As an example from the BLESS dataset, the Turkish equivalent of the word "dishwasher" is "*bulaşık makinesi*", which is a word pair consisting of two words. Our native Turkish speakers checked this situation and first tried to convert the two-word representation to a one-word representation without breaking the meaning of the words; if these two word representations could not be represented by a semantically correct one-word representation, they discard samples containing these words from our datasets used for the experiments.

Finally, examples that can be compared with English in terms of grammar, but which are meaningless for Turkish after translation, were discarded. For example, in the Word analogy task, especially in the MSR dataset, analogies such as "good" - "better", "heavy" - "heavier" were excluded from the dataset because they did not create a comparable one-word situation when translated into Turkish.

C. Tasks

1) Word similarity: The main idea of this task is to measure the performance of the word representations obtained from the evaluated models by comparing their distance in vector space with the scores obtained from human judgments.For this purpose, the cosine similarity of the word representations obtained from the trained models and the harmonic mean of Pearson and Spearman correlations, which measure the linear and non-linear correlations between the variables, respectively, of human judgements in the datasets, were calculated as a method also used in [17]. Three commonly used datasets were translated and used for this task: SimVerb-3500, MC-30, and SemEval-2017.

SimVerb-3500 includes a total of 3,500 verb pairs which are evaluated for their semantic similarity, with ratings ranging from 0 to 10 where lower ratings indicate pairs that are related but not particularly similar. The dataset prepared for the SemEval-2017 Task 2 (Multilingual and Cross-Lingual Semantic Word Similarity) consists of 500 pairs that were assessed for their semantic similarity. The assessment was conducted using a scale ranging from 0 to 4. MC-30, comprises 30 pairs of words including equal number of word pairs with high, moderate, and low similarity. After translation and preprocessing steps 1825, 319, and 24 word pairs remained for each dataset respectively.

2) Word analogy: The task of finding a vector that captures the relationship between the words in an analogy is called word analogy. To illustrate, consider a set of words such as "king", "queen" and "man" with an unknown fourth word. The aim is to find the missing word that completes the analogy, in this case "woman", because the relationship is based on gender. For this task, the process involves subtracting the vector representation of "man" from "king" and adding the resulting vector to "queen". By finding the most similar vector to the resulting vector, we can identify the most appropriate word, which in this case would be "woman". The MSR dataset is translated and used for this task.

MSR (Microsoft Research Syntactic Analogies) dataset consists of 8,000 questions that are categorized into 16 classes. 4001 word pairs are obtained respectively after translation and preprocessing steps.

3) Concept categorization: The task involves partitioning a given set of words into subsets, where each subset consists of words belonging to distinct categories. BLESS and ESSLLI-2008 datasets are utilized for this task. The given words were clustered based on the number of categories in each dataset using K-means. To evaluate how pure the clustering results are we used a commonly used purity score in clustering algorithms which is also used in [18].

BLESS (Baroni and Lenci Evaluation of Semantic Spaces) dataset comprises 200 words from 27 semantic classes and ESSLI-2008 (European Summer School in Logic, Language and Information) dataset consists of 45 words, which are categorized into 9 semantic classes. After translation and preprocessing steps 181 and 32 pairs of words left for each dataset respectively.

4) Outlier word detection: As the name indicates, outlier word detection is finding a word that deviates semantically from the rest of a pre-established cluster. For this task, the 8-

TABLE I
COMPARISON OF LANGUAGE MODELS ON VARIOUS TASKS.

Benchmarks / Models	*Word Similarity*			*Word Analogy*			*Concept Categorization*		*Outlier Word Detection*
	SemEval-2017	**MC-30**	**SimVerb-3500**	**MSR @10**	**MSR @5**	**MSR @1**	**BLESS**	**ESSLLI**	**8-8-8 Dataset**
FastText$_{default\_skipgram}$	0.621	0.686	0.206	0.673	0.586	0.328	0.646	**0.735**	0.781
FastText$_{custom\_skipgram}$	0.622	**0.694**	0.212	0.623	0.537	0.311	**0.751**	0.588	0.781
FastText$_{cbow}$	0.587	0.649	0.160	**0.802**	**0.743**	**0.506**	0.597	0.647	0.718
Word2Vec$_{default\_cbow}$	0.593	0.575	0.153	0.665	0.585	0.375	0.552	0.552	0.719
Word2Vec$_{custom\_cbow}$	0.617	0.612	0.180	0.752	0.677	0.441	0.624	0.624	0.750
Word2Vec$_{default\_skipgram}$	0.623	0.596	0.194	0.578	0.508	0.320	0.607	0.608	**0.797**
Word2Vec$_{custom\_skipgram}$	**0.628**	0.686	**0.225**	0.655	0.591	0.361	0.657	0.657	0.796
GloVe$_{default}$	0.563	0.492	0.147	0.529	0.462	0.312	0.608	0.647	0.703
GloVe$_{custom}$	0.564	0.656	0.175	0.602	0.537	0.377	0.669	0.676	0.734

8-8 dataset is used, which contains eight different topics with eight words each and eight outliers.

IV. EXPERIMENTS AND RESULTS

1) Word similarity: We used the SemEval-2017, MC-30 and SimVerb-3500 datasets to test the trained models on the word similarity task. On this task, observed results from Table I show that the Skip-gram technique generally gives better results on the test datasets than CBOW and the best result is obtained with Skip-gram version of the Word2Vec model. When the datasets are examined, models are more successful on SemEval and MC-30 datasets in comparison with the SimVerb-3500. This poor result on SimVerb-3500 dataset occured due to the many inflected forms of verbs in the dataset.

2) Word analogy: For this task we used the MSR dataset. Table I shows that the model with the best performance in finding the target word in accordance with the word analogy task is the fastText trained with the CBOW model, and our other models have had results close to this accuracy rate. When the first 10 words in vector space, which are closest to the vector calculated by word analogy calculation are taken, it was observed that our accuracy of reaching the target word specified in the benchmark increased significantly. The accuracy dropped when experimenting with 5 and 1 words.

3) Concept categorization: When we look at this task specifically, we can observe from Table I that the Skip-gram models give better results. When we tested it with the BLESS dataset, the best result was obtained by the FastText$_{custom_skipgram}$ model. In the ESSLII 2018 dataset, the best result was again using the fastText model with default parameters.

4) Outlier Word Detection: Our last task was performed using the 8-8-8 Dataset. It is seen that the results from Table I are consistent with each other when we give each element one by one from the list of 8 words specified as outlier to the 8 words in the same category and ask them to find the outlier one each time. The best performing model is the Skip-gram version of Word2Vec, which is lower than human performance, estimated at 0.984 to 1 [7].

As a result, it can be said that the results are consistent and the models perform the tasks with a certain success. Although there are linguistic difficulties in translation due to the size of the common dataset we train our models, the preprocess operations performed and the datasets we used to test different tasks are not originally Turkish. It can also argued that the lack of occurence of a word in a meaningful context raised problems for tasks like concept categorization and outlier word detection.

V. CONCLUSION

In conclusion, this paper provides an intrinsic evaluation of Turkish word embeddings using three popular models: Word2Vec, GloVe, and fastText. The study evaluates the models on four different tasks: word similarity, word analogy, concept categorization, outlier word detection. The results of the experiments show that fastText performs well on most tasks, especially when trained with specific parameters. The study also highlights the importance of intrinsic evaluation methods in assessing the quality of word embeddings. Overall, this research contributes to the ongoing efforts to improve word embeddings in NLP and provides insights into their performance in the Turkish language. Future research can build upon these findings and explore other aspects of word embeddings in NLP.

REFERENCES

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

[2] 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)*. Association for Computational Linguistics, 2014. [Online]. Available: https://doi.org/10.3115/v1/d14-1162

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.

[4] J. Camacho-Collados, M. T. Pilehvar, N. Collier, and R. Navigli, "Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity," in *International Workshop on Semantic Evaluation*, 2017.

[5] E. Vylomova, L. Rimell, T. Cohn, and T. Baldwin, "Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning," *ArXiv*, vol. abs/1509.01692, 2015.

[6] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Annual Meeting of the Association for Computational Linguistics*, 2014.
[7] J. Camacho-Collados and R. Navigli, "Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations," in *RepEval@ACL*, 2016.
[8] H. V. Agun and O. Yilmazel, "Intrinsic evaluation of word embeddings for turkish," in *Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control*. ACM, Nov. 2020. [Online]. Available: https://doi.org/10.1145/3440084.3441184
[9] G. Ercan and O. T. Yildiz, "Anlamver: Semantic model evaluation dataset for turkish - word similarity and relatedness," in *International Conference on Computational Linguistics*, 2018.
[10] E. Kuriyozov, Y. Doval, and C. Gómez-Rodríguez, "Cross-lingual word embeddings for turkic languages," in *International Conference on Language Resources and Evaluation*, 2020.
[11] C. Callison-Burch and A. Cocos, "The language of place: Semantic value from geospatial context," in *Conference of the European Chapter of the Association for Computational Linguistics*, 2017.
[12] M. Kutlu, C. Cigir, and I. Çiçekli, "Generic text summarization for turkish," *2009 24th International Symposium on Computer and Information Sciences*, pp. 224–229, 2009.
[13] Wikimedia dumps. Accessed in June 2023. [Online]. Available: https://dumps.wikimedia.org/trwiki/20230620/
[14] R. Rehurek and P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.
[15] Glove. [Online]. Available: https://nlp.stanford.edu/projects/glove/
[16] fasttext. [Online]. Available: https://fasttext.cc/
[17] J. Camacho-Collados, M. T. Pilehvar, N. Collier, and R. Navigli, "Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, August 2017, pp. 15–26. [Online]. Available: http://www.aclweb.org/anthology/S17-2002
[18] M. Baroni, S. Evert, and A. Lenci, "Esslli 2008 workshop on distributional lexical semantics." Hamburg, Germany: Association for Logic, Language and Information, 2008.