International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

Exploring Word Embeddings for Text Classification: A Comparative Analysis

Satya Mohan Chowdary G[1], T Ganga Bhavani[2] , D Konda Babu[3] , B Prasanna Rani[4], K Sireesha[5]

[1] *Assistant Professor, Information Technology , Pragati Engineering College(A), Surampalem, Andhra Pradesh*

[2] *Assistant Professor, Information Technology , Pragati Engineering College(A), Surampalem, Andhra Pradesh*

[3] *Assistant Professor, Information Technology , Pragati Engineering College(A), Surampalem, Andhra Pradesh*

[4] *Assistant Professor, CSE , VSM College of Engineering, Ramachandrapuram, Andhra Pradesh*

[5] *Teaching Assistant, Information Technology , Pragati Engineering College(A), Surampalem, Andhra Pradesh*

Corresponding Author Orcid ID : https://orcid.org/0000-0002-1952-6403

ABSTRACT

For language tasks like text classification and sequence labeling, word embeddings are essential for providing input characteristics in deep models. There have been many word embedding techniques put out in the past ten years, which can be broadly divided into classic and context-based embeddings. In this study, two encoders—CNN and BiLSTM—are used in a downstream network architecture to analyze both forms of embeddings in the context of text classification. Four benchmarking classification datasets with single-label and multi-label tasks and a range of average sample lengths are selected in order to evaluate the effects of word embeddings on various datasets. CNN routinely beats BiLSTM, especially on datasets that don't take document context into account, according to the evaluation results with confidence intervals. CNN is therefore advised above BiLSTM for datasets involving document categorization where context is less predictive of class membership. Concatenating numerous classic embeddings or growing their size for word embeddings doesn't greatly increase performance, while there are few instances when there are marginal gains. Contrarily, context-based embeddings like ELMo and BERT are investigated, with BERT showing better overall performance, particularly for longer document datasets. On short datasets, both context-based embeddings perform better, but on longer datasets, no significant improvement is seen.In conclusion, this study emphasizes the significance of word embeddings and their impact on downstream tasks, highlighting the advantages of BERT over ELMo, especially for lengthier documents, and CNN over BiLSTM for certain scenarios involving document classification.

Keywords— Text Classification, Word Embeddings, Similarity in Words, NLP, Distributed Word Representation.

1. Introduction

In a variety of Natural Language Processing (NLP) tasks, the representation of natural language has emerged as a key topic. One-hot encoding, Bag of Words (BoW), term-frequency, and other straightforward methods[1] are some of the simplest ways to represent natural language in textual data. The relevant information isn't included in the mapped space of the representations using these methods, which is a weakness. The relative closeness of the words is lost in representations like one-hot encoding, while BoW representations fail to capture the context and word order. These representations produce vector representations that are huge (equal to the size of the vocabulary).

DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

The usage of distributed vector representations of textual data in several algorithms has been demonstrated. Based on the mutual information between words in the corpus, each word vector in a particular corpus is represented [2]. Vector representation can be computed at different levels including: characters [3], [4], words [5], [6], [7], phrases [8], [9], sentences [10], document [11],etc. Word representations in a continuous Rn space are called word embeddings. Keeping the information about the meanings and commonalities of the words, each word is projected in n-dimensional vector space as a real-valued vector. In the planned space, words with similar meanings will be mapped closer together. As an example, the operation "King - Man + Woman" yields a vector close to "Queen" [5], [12]. Word vectors represent the syntactic and semantic regularities in language.
Today, several word embedding systems integrate various word representation methods. It takes a lot of computational time, huge corpora, and computational capacity to train word embeddings. Therefore, the idea of transfer learning—using already-trained embeddings and then training the NLP model with them—is the preferable method. The projected embedding space is not apparent to humans, therefore choosing the suitable word embeddings is a challenging task. However, choosing the pre-trained word embeddings used as input in the NLP model is the first and most important step, as the standard of the word representation significantly affects the model's overall performance. As a result, researchers may find it useful to provide comparison experimental data for the approaches in order to help them choose the best model embeddings based on the comparative analysis. Word embeddings are not uniformly evaluated. The evaluation techniques can often be divided into two categories: intrinsic and extrinsic evaluation [13]. Through direct examination of the syntactic or semantic links between words, intrinsic evaluations gauge a representation's quality without reference to any particular tasks involving natural language processing. In the comparing process of intrinsic evaluation, human-created benchmark datasets produced with a specific aim of comparing words by similarity, relatedness, or similar, are significant. Extrinsic evaluation of word vectors refers to the evaluation of the embeddings on a genuine NLP task, such as part-of- speech tagging [14], named-entity recognition [15], sentiment analysis [16], and neural machine translation [17].
When performing the assessments, the main goal is to identify the highest quality word embeddings. Additionally, a key component of the method's ability to encode significant information in the projected space is the word embeddings' dimensionality. More information can be incorporated into a space with higher dimensionality, but at a certain point, the marginal advantage starts to decline [5]. The vectors are typically set to have a dimensionality of between 50 and 1,000, with 300 being the most common value. As a result, further research is needed to determine the best dimensionality before analyzing the word embeddings.
In this research, we concentrate on various pre-trained word embeddings utilized in cutting-edge models for some NLP tasks and evaluate their performance on word similarity detection with benchmark datasets for word pairs1. To compare the most recent word embeddings, experiments were conducted using the datasets WordSim353 [18], SimLex999 [19], and SimVerb3500 [20].
The rest of the paper is divided into the following sections. The word embedding strategies employed for comparison analysis are described in Section 2. The research that has been done on the issue is summarized in Section 3. Section 4 contains the comparison analysis as well as a discussion of the test results. The paper is concluded and suggestions for further research are provided in Section 5.

2. Word Embeddings

Word embeddings are crucial in natural language processing tasks, as they allow representing words in a continuous vector space where semantic relationships are preserved. The provided text introduces three main categories of word embedding methods: neural network-based, word matrix-based, and ensemble methods. Let's analyze each category and the specific techniques mentioned within them.

DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

2.1 Neural Network-Based Methods: Word2Vec and FastText

Neural network-based methods aim to learn word embeddings through neural networks. Word2Vec and FastText are two prominent examples.

Word2Vec:

Word2Vec introduced two models - Continuous Bag Of Words (CBOW) and Skip-gram. These models use a hidden layer with N neurons (dimensionality of word embeddings).CBOW predicts the target word from its context words. Skip-gram predicts context words given the target word. Word2Vec offers efficient training techniques like Hierarchical Softmax and Skip-Gram Negative Sampling (SGNS).Pre-trained Word2Vec embeddings are available, often using 300-dimensional vectors trained on massive corpora.

FastText:

FastText is derived from Skip-gram with a subword model. It considers the internal structure of words using character n-grams. Each word is represented as a sum of its n-gram vectors. This approach enables learning representations for rare words and sharing representations across words. Pre-trained models are available, trained with subword information on large corpora.

2.2 Word Matrix-Based Methods: GloVe and LexVec

Word matrix-based methods utilize matrix factorization techniques on word co-occurrence information to create embeddings.

GloVe (Global Vectors for Word Representation):

GloVe combines global matrix factorization and local context window methods. It optimizes embeddings so that dot products reflect word co-occurrence ratios. Pre-trained GloVe vectors are available, trained on various datasets and dimensions.

LexVec:

LexVec factors the Positive Pointwise Mutual Information (PPMI) matrix using a reconstruction loss function. It weighs errors based on co-occurrence frequency. Pre-trained LexVec models exist, with comparisons to state-of-the-art methods in word similarity and analogy tasks.

2.3 Ensemble Methods: ConceptNet Numberbatch

Ensemble methods integrate different embeddings and structured knowledge to improve word representations.

ConceptNet Numberbatch:

This method combines GloVe, Word2Vec embeddings, and knowledge from ConceptNet and PPDB. It uses retrofitting to adjust embeddings using knowledge graphs, enhancing embeddings' semantic relationships. Pre-trained ConceptNet Numberbatch embeddings are available.

2.4 Comparison and Insights:

Neural Network vs. Word Matrix:

Neural network-based methods like Word2Vec and FastText capture syntactic and semantic information efficiently. They excel at capturing contextual relationships. Word matrix-based methods like GloVe and LexVec focus on co-occurrence statistics, capturing both global and local semantic information.

Subword Information:

FastText leverages subword information, making it effective for handling out-of-vocabulary words and rare words. This approach helps in learning meaningful embeddings for morphologically rich languages.

Data and Model Sizes:

Word2Vec, FastText, GloVe, and LexVec all provide pre-trained embeddings with different dimensions and trained on diverse datasets.Selection depends on the specific task, dataset, and computational resources available.

Ensemble Approaches:

Ensemble methods like ConceptNet Numberbatch offer improved embeddings by combining different sources of information. These methods are effective for capturing both intrinsic and extrinsic word relationships.

3. Related Work

Word embeddings have become a pivotal tool in natural language processing (NLP) and machine learning, enabling the representation of words as continuous vectors in a semantic space. Researchers have extensively studied the performance of various word embedding methods on different languages and datasets to assess their effectiveness and applicability. This report presents a summary and analysis of several studies that have compared different word embedding methods in diverse linguistic contexts and domains.

Berardi et al. [30] conducted a comprehensive comparison of the Skip-gram model from Word2Vec and GloVe in the Italian language. They employed two distinct datasets, the Italian Wikipedia dump and a collection of Italian novels. Despite the diversity in purpose and style, the Skip-gram model consistently outperformed GloVe on both datasets. The authors noted that the performance of the Italian-trained models did not match those trained on English data, potentially reflecting the inherent complexity of the Italian language.

In [31], a cross-linguistic comparison was carried out, focusing on Hungarian and English. A Skip-gram model trained on the Hungarian Webcorpus was compared to an English model. Morphological questions yielded similar results, but the English model dominated in semantic questions. Additionally, a proto dictionary comparison was conducted using CBOW Word2Vec and GloVe models for Hungarian, Slovenian, Lithuanian, and English languages.

Baroni et al. [32] evaluated four word embedding models—Word2Vec's CBOW, DISSECT, Distributional Memory model, and Collobert and Weston—on a large English corpus. They found that Word2Vec's CBOW model consistently outperformed other methods across various tasks. Similarly, Ghannay et al. [33] assessed CSLM word embeddings, dependency-based embeddings, combined embeddings, and Word2Vec's Skip-gram on NLP tasks. The dependency-based embeddings demonstrated the best performance, with a suggestion that combined embeddings could yield further improvements.

Another study [13] compared Word2Vec's CBOW model, GloVe, TSCCA, C&W embeddings, Hellinger PCA, and Sparse Random Projections across 14 test datasets. The conclusion was that Word2Vec's CBOW model excelled on 10 out of the 14 datasets, showcasing its overall robustness. Moreover, the assessment extended beyond general NLP, as word embeddings were compared within specific domains like biomedical domains [40], [41].

The studies mentioned above emphasize the significance of choosing appropriate word embedding methods based on the language, dataset, and task. While Word2Vec's CBOW model often outperforms others in English contexts, its performance varies in other languages and domains. The complexity of languages, as observed in the Italian case, might influence model performance. Cross-linguistic comparisons highlight the interplay between morphological and semantic nuances. Additionally, the potential for combining embeddings and adapting methods to specific domains is evident in the studies.

Comparative studies of word embeddings across languages and domains reveal the contextual dependence of embedding methods' performance. While Word2Vec's CBOW model often demonstrates superiority in English, its efficacy differs in other languages. Cross-linguistic assessments and domain-specific evaluations emphasize the need for tailored approaches. Researchers must consider linguistic intricacies, training data diversity, and task requirements when selecting word embedding methods for various applications in the broader NLP landscape.

4. Comparative Analysis

There are two methods for evaluating word vectors. Extrinsic evaluation involves actual NLP tasks like sentiment analysis or natural language inference. Word vectors are incorporated into a deep neural network's embedding layer for that reason. After that, the model is trained on a particular NLP task. If the results are poor, it is unclear whether the word vectors do not adequately capture the meaning of the words or whether the model is just inadequate for the task.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

The previous concern demonstrates the necessity of intrinsic evaluation. Word vectors are evaluated directly on various tasks in this sort of evaluation, which is an intermediate evaluation. Tasks like word vector analogies and word similarities are included in the intrinsic evaluation.
By measuring how well a word vector's cosine distance after addition captures common semantic and syntactic comparison queries, word vectors are assessed. For instance, the operations "King - Man + Woman" and "Windows - Microsoft + Google" produce vectors that are similar to "Queen" and "Android," respectively.
Values used to calculate word similarity are used to compare word pairs. These values are gathered in several human-generated datasets used as a resource for comparing the accuracy of word mapping in various word representations in a given space. By combining or isolating word similarity with word relatedness or word association, different datasets depict word similarity differently.

Data:

353 noun pairings with human similarity ratings make up the WordSim35312 dataset [18]. The dataset is regarded as the gold standard in the computation of word similarity and relatedness. The word pairings in this resource's similarity scores serve as a typical indicator of how similar words are based on their relatedness or associations. Later, the dataset is annotated with semantic relations by differentiating between similarity and relatedness [42].
SimLex99913 has lately gained popularity as a lexical resource for monitoring word similarity computation development. Instead than focusing on relatedness or association, this benchmark dataset measures similarity [19]. There are 222 verb-verb pairings, 111 adjective-adjective combinations, and 666 noun-noun pairs in all. The dataset's annotation criteria limit synonymy's degree of similarity [43]. The word embedding models' capacity to capture word similarity (without the influence of relatedness or association) while simultaneously removing the dependence on relatedness or association can be measured using SimLex999.
3500 verb-verb pairs make up the SimVerb350014 verb similarity dataset [20]. There is a distinction between similarity and relatedness, and this dataset was created using the same annotation criteria as the SimLex-999. As a result, this technique is a comprehensive resource for verb similarity.

Word Similarity Experiments:

Words are represented by word embeddings so that words with similar meanings have comparable vector representations. The experiments in this part test various pre-trained word embeddings' capacity to recognize word similarity. The word embedding vectors' cosine similarity is calculated for each word pair in the datasets. In Figure 1, the average cosine similarity for each dataset is presented.
For the WordSim353, SimLex999, and SimVerb3500 datasets, respectively, the average similarity determined by human assessments is 5.86, 4.56, and 4.29. GloVe and FastText word embeddings had the highest average cosine similarity for each dataset. For the WordSim353, SimLex999, and SimVerb3500 datasets, respectively, the average cosine similarity for GloVe embeddings are 5.37, 4.62, and 3.79. For the WordSim353, SimLex999, and SimVerb3500 datasets using the FastText embeddings, the average cosine similarity is 4.69, 4.81, and 4.12, respectively. These results suggest that GloVe and FastText are more effective in identifying word similarity.
Higher dimensional word vectors are anticipated to have a more noticeable quality of representation in terms of dimensionality. When analyzing the GloVe embeddings with various dimensionalities, the SimLex999 dataset and the SimVerb3500 dataset both provide strong support for this claim. For the SimLex999 dataset, 200-dimensional embeddings pre-trained on Twitter and 300-dimensional embeddings pre-trained on Wikipedia, the average cosine similarity is more in line with the ground truth similarity. The claim only holds for 200-dimensional embeddings pre-trained on Twitter for the SimVerb3500 dataset, although similarities for vectors pre-trained on Wikipedia are more comparable to ground truth similarities when the dimensionality is 50. Less dimensionality in the

DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

vector space means that the WordSim353 dataset is more comparable to the ground truth in terms of cosine.

Correlation Analysis

Since similarities determined by one metric may be higher than similarities determined by another, average similarity may not necessarily offer useful insight into word similarities. Although these similarity distributions have different average values, they might be connected. As a result, we determine the correlation coefficient between word vector cosine similarity and ground truth similarity for all pre-trained word embedding vectors.

For each pair of ground-truth similarities and cosine similarities, the Spearman correlation coefficient, Pearson correlation coefficient, and Kendall's tau correlation coefficient are calculated. The results are shown in Figure 2 for the WordSim353 dataset, Figure 3 for the SimLex999 dataset, and Figure 4 for the SimVerb3500 dataset. They are summarized in Tables 1, Table 2, and Table 3 for the WordSim353, SimLex999, and SimVerb3500 datasets, respectively.

The correlation analysis' findings conflict with those of average similarity. In terms of average cosine similarity, GloVe and FastText word embeddings performed better at identifying word similarities. But their correlation coefficient is the lowest. The number is close to 0, indicating that there is no relationship between ground truth and word vector cosine similarity. Despite being created to address Word2Vec's shortcomings, FastText does not demonstrate an improved ability to capture word similarity, as seen by correlations of roughly 0.1 for SimLex999 and SimVerb3500 and approximately 0.2 for WordSim353. GloVe performs poorly on SimLex999 and SimVerb3500 as well, despite the fact that it is a popular starting point for many high-performance deep learning models used for various NLP applications. The correlation on WordSim353, on the other hand, is significantly stronger, with a maximum Spearman correlation coefficient of 0.61.

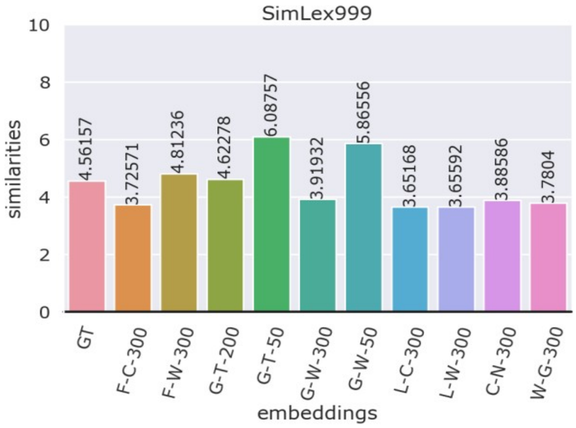
ConceptNet Numberbatch word embeddings achieve the highest correlation for all three datasets, demonstrating a positive correlation between similarity distributions. The Word2Vec and LexVec word embeddings have the second-highest correlation coefficient, which is approximately 0.7 for the WordSim353 dataset, approximately 0.4 for the SimLex999 dataset, and nearly 0.3 for the SimVerb3500 dataset.

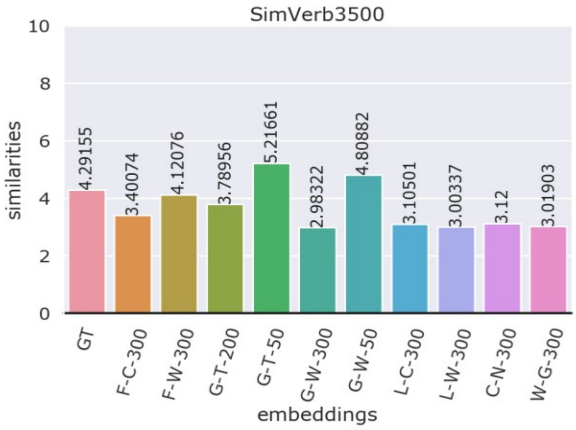
The correlation results are consistent with the hypothesis that higher dimensional word vectors are of higher quality. The SimLex999 and SimVerb3500 dataset correlation coefficients for GloVe word embeddings are low, but they do demonstrate that increasing dimensionality leads to stronger correlation. We can also deduce that word vectors with higher dimensions are better at capturing word similarity.

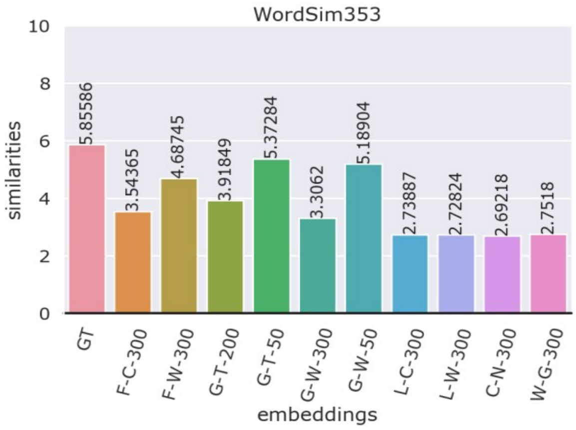
Figure 1. Cosine similarities that are typical. likenesses between GT and ground truth. FastText embeddings in 300 dimensions, pre-trained on the Crawl dataset, are designated as F-C-300. Wikipedia dataset served as the training set for the 300-dimensional FastText embeddings known as F-W-300. Pre-trained 200-dimensional GloVe embeddings on the Twitter dataset are designated as G-T-200. Pre-trained 50-dimensional GloVe embeddings using the Twitter dataset are designated as G-T-50. Pre-trained 300-dimensional GloVe embeddings using Wikipedia data are designated as G-W-300. Wikipedia was used as the training data for the 50-dimensional GloVe embeddings known as G-W-50. 300-dimensional LexVec embeddings in L-C-300 that were pre-trained using the Crawl dataset. 300-dimensional LexVec embeddings that were pre-trained using the Wikipedia dataset are called L-W-300. 300-dimensional ConceptNet Numberbatch embeddings are designated as N-C-300. Pre-trained 300-dimensional Word2Vec embeddings using the Google News dataset are designated as W-G-300.

Table 1. Correlation coefficients between ground truth similarities and word vector cosine similarities for the WordSim353 dataset. S - Spearman correlation coefficient. Pr - Pearson correlation coefficient. K - Kendall’s tau correlation coefficient.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621







Word Embeddings	S	Pr	K
FastText-Crawl-300	0.25	0.26	0.17
FastText-Wikipedia-300	0.19	0.19	0.13
GloVe-Twitter-200	0.52	0.53	0.36
GloVe-Twitter-50	0.46	0.46	0.32
GloVe-Wikipedia-300	0.61	0.60	0.45
GloVe-Wikipedia-50	0.50	0.51	0.36
LexVec-Crawl-300	0.72	0.68	0.53
LexVec-Wikipedia-300	0.66	0.63	0.48
ConceptNet-Numberbatch-300	0.81	0.75	0.63
Word2Vec-GoogleNews-300	0.69	0.65	0.51

Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

Table 2. Correlation coefficients between ground truth similarities and word vector cosine similarities forthe SimLex999 dataset. S - Spearman correlation coefficient. Pr - Pearson correlation coefficient. K - Kendall's tau correlation coefficient.

Word Embeddings	S	Pr	K
FastText-Crawl-300	0.16	0.16	0.11
FastText-Wikipedia-300	0.09	0.07	0.06
GloVe-Twitter-200	0.13	0.14	0.08
GloVe-Twitter-50	0.10	0.10	0.06
GloVe-Wikipedia-300	0.37	0.39	0.30
GloVe-Wikipedia-50	0.26	0.29	0.18
LexVec-Crawl-300	0.44	0.45	0.31
LexVec-Wikipedia-300	0.38	0.39	0.27
ConceptNet-Numberbatch-300	0.63	0.65	0.46
Word2Vec-GoogleNews-300	0.44	0.45	0.31

Table 3. Correlation coefficients between ground truth similarities and word vector cosine similarities for the SimVerb3500 dataset. S - Spearman correlation coefficient. Pr - Pearson correlation coefficient. K - Kendall's tau correlation coefficient.

Word Embeddings	S	Pr	K
FastText-Crawl-300	0.11	0.11	0.07
FastText-Wikipedia-300	0.03	0.02	0.02
GloVe-Twitter-200	0.06	0.07	0.04
GloVe-Twitter-50	0.03	0.04	0.02
GloVe-Wikipedia-300	0.23	0.23	0.16
GloVe-Wikipedia-50	0.15	0.16	0.10
LexVec-Crawl-300	0.30	0.31	0.21
LexVec-Wikipedia-300	0.28	0.28	0.19
ConceptNet-Numberbatch-300	0.57	0.59	0.41
Word2Vec-GoogleNews-300	0.36	0.38	0.25

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

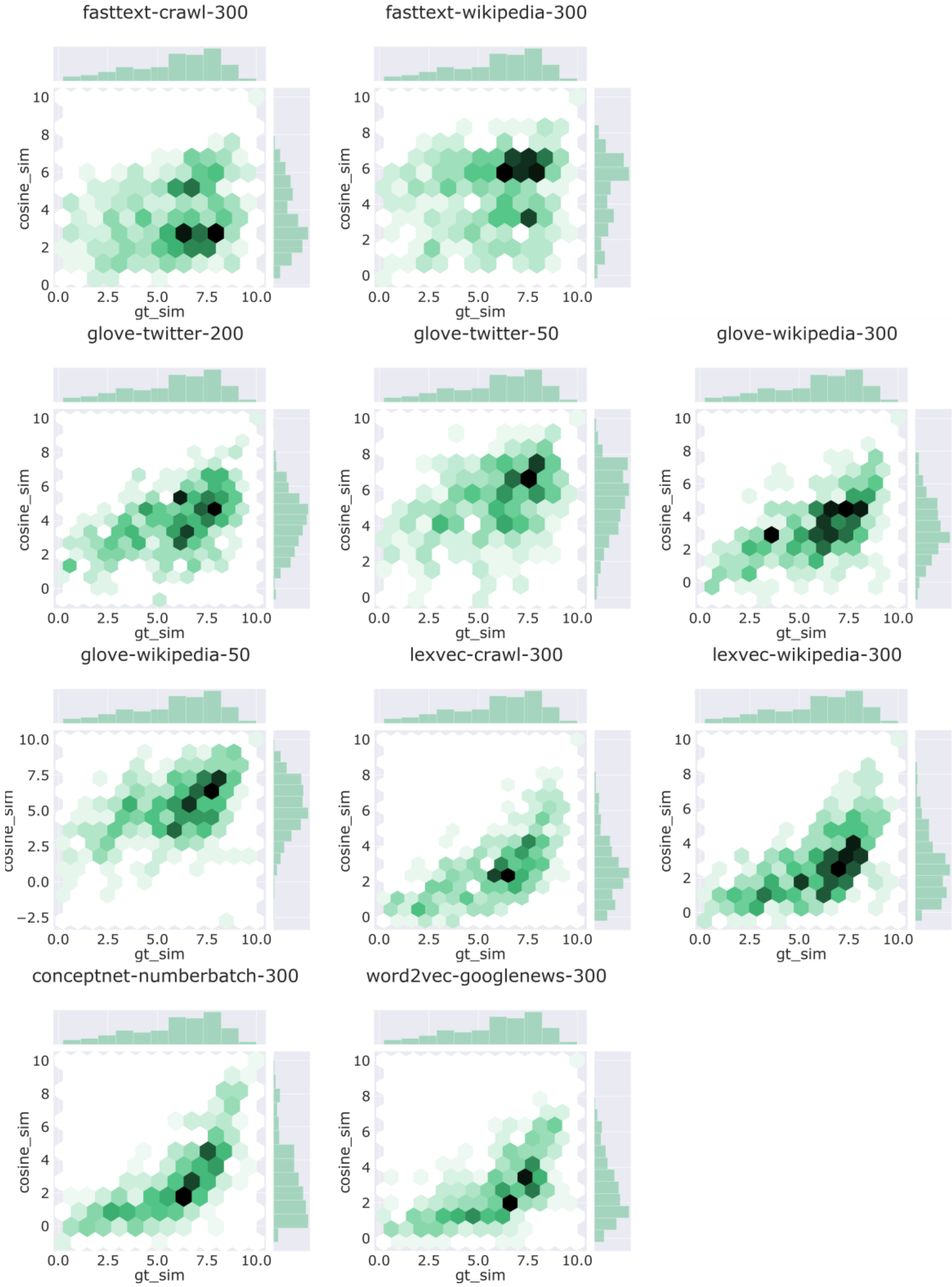


Figure 2. Correlation between similarities found using various word embedding techniques and the ground truth for the WordSim353 dataset. The ground truth similarity measure (Gt_sim) is determined by human judgments. Word embeddings with similar cosines are known as cosine_sim.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

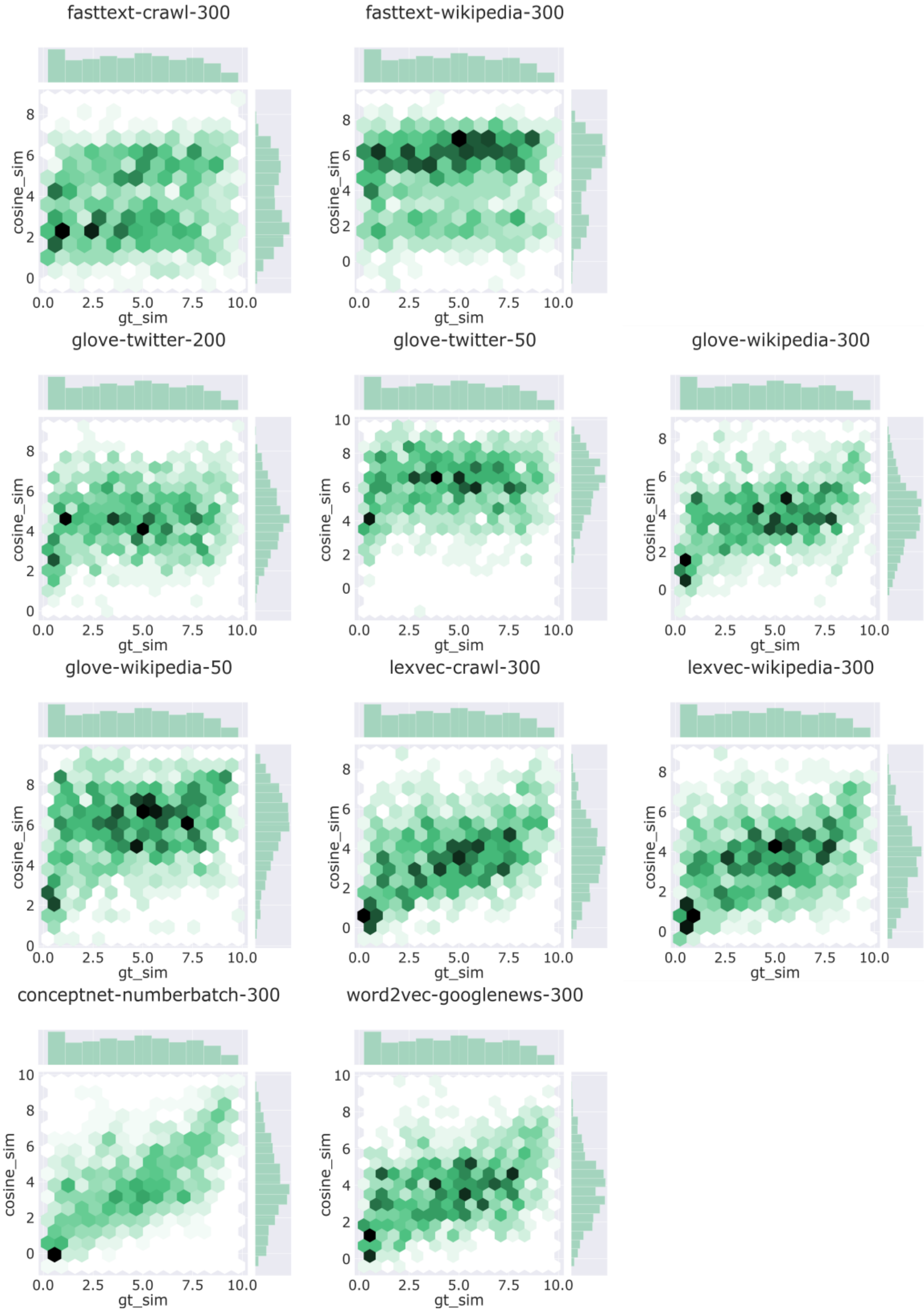


Figure 3. Comparison of the SimLex999 dataset's ground truth with comparisons made using various word embedding techniques. Ground truth similarities determined by human ratings are obtained using Gt_sim. When comparing word embeddings, use the cosine_sim function.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.7 September - October – 2023
DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

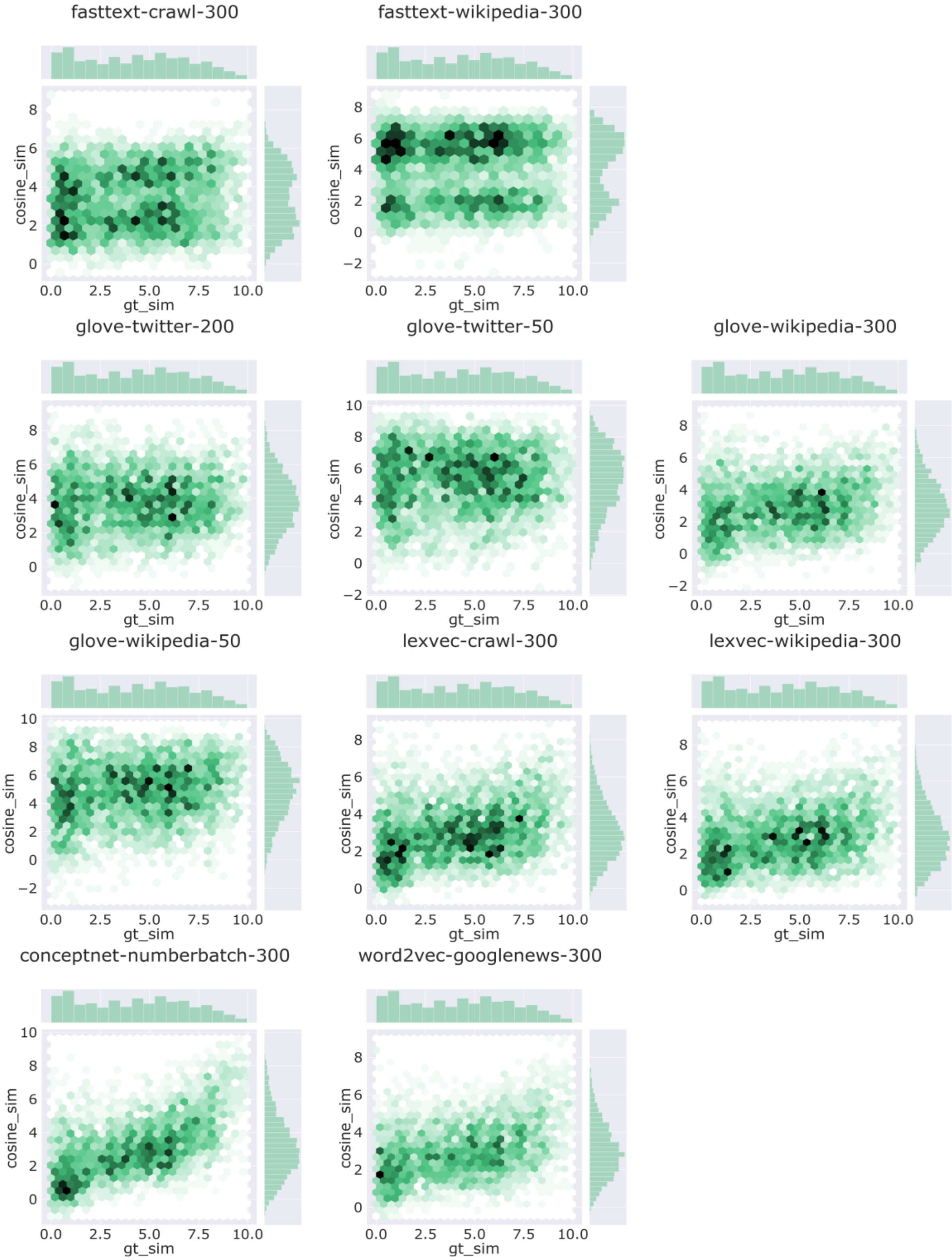


Figure 4. Correlation between similarities found using various word embedding techniques and similarities found using the SimVerb3500 dataset. Gt_sim is a measure of ground truth similarity determined by human judgments. Cosine_sim - Cosine similarity between word embeddings.

5. Conclusion and Future work

We have examined various word embedding techniques in this research. According to the process of creation, these methods might be broadly categorized into two groups: neural network-based and matrix-based. We also looked at an amalgam of the two methods called an ensemble method.For each of the techniques under consideration, word embedding similarity studies were carried out. On three separate datasets, these studies test the cosine similarity measure's capacity to identify word similarities between pairs of words. It was determined that GloVe and FastText beat the other word embedding methods after analyzing only the average similarity data and comparing these values with the average value of the human ratings.

We also conducted correlation analysis after running the word embedding similarity studies. For each pair of ground truth similarities and cosine similarities of the word embeddings, the Spearman correlation coefficient, Pearson correlation coefficient, and Kendall's tau correlation coefficient were calculated. Using the information gathered during this research, we came to the conclusion that, despite GloVe and FastText having high cosine similarity values, there is no association between ground truth and the cosine similarities of word vectors. Despite having low average similarity values, ConceptNet Numberbatch word embeddings have the highest correlation coefficients.

Including the textual context of the words allows for a more thorough analysis of word similarity. Recent word embedding techniques, such as ELMO [44] and BERT [45], make use of the context of the words as they appear in the text to produce deeply contextualized word representations. As part of our ongoing research, we will compare and contrast these various word mappings. Furthermore, it would be more advantageous to compare the various word representations by merging the various concepts from intrinsic and extrinsic assessments.

References

1. Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. “Scoring, term weighting and the vector space model.” Introduction to information retrieval 100 (2008): 2-4.

2. Mikolov, Tomáš, Wen-tau Yih, and Geoffrey 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. 2013.

3. Santos, Cicero D., and Bianca Zadrozny. “Learning character-level representations for part-of-speech tagging.” Proceedings of the 31st international conference on machine learning (ICML- 14). 2014.

4. dos Santos, Cıcero, et al. “Boosting Named Entity Recognition with Neural Character Embeddings.” Proceedings of NEWS 2015 The Fifth Named Entities Workshop. 2015.

5. Mikolov, Tomas, et al. “Efficient Estimation of Word Representations in Vector Space.” ICLR (Workshop Poster). 2013.

6. Mikolov, Tomas, et al. “Distributed representations of words and phrases and their compositionality.” Advances in neural information processing systems. 2013.

7. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. “Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

8. Shubham Agarwal; Nishant Moghe; Vaishali Wadhe. "Big Data Analytics for Supply Chain Optimization: A Review of Methodologies and Applications". International Research Journal on Advanced Science Hub, 5, 07, 2023, 215-221. doi: 10.47392/irjash.2023.046

9. Hanumanthappa S; Guruprakash C D. "Feature Extraction from Brain MR Images for Detecting Brain Tumor using Deep Learning Techniques". International Research Journal on Advanced Science Hub, 5, 07, 2023, 242-247. doi: 10.47392/irjash.2023.049

10. Shakti Punj; Lavkush Sharma; Brajesh Kumar Singh. "Enhancing Face Mask Detection Using Convolutional Neural Networks: A Comparative Study". International Research Journal on Advanced Science Hub, 5, 08, 2023, 280-289. doi: 10.47392/irjash.2023.054

11. Sirajudeen S; Sudha S. "A Review - Smoke-Fire Detection and YOLO (You Only Look Once)". International Research Journal on Advanced Science Hub, 5, 08, 2023, 248-256. doi: 10.47392/irjash.2023.051

12. Mr. Buddesab; Nanda Ashwin; Shruthi M; Rekha P; Pavan Mulgund. "Real time eye based Password Authentication by Eye Blinking System". International Research Journal on Advanced Science Hub, 5, Issue 05S, 2023, 1-6. doi: 10.47392/irjash.2023.S001

13. Rekha P; Mr. Buddesab; Shruthi K; Sathya M; Nanda Ashwin. "Tabib : Chatbot for Healthcare Automation with Audio Assistance using Artificial Intelligence". International Research Journal on Advanced Science Hub, 5, Issue 05S, 2023, 7-14. doi: 10.47392/irjash.2023.S002

14. Krishnan S; Pranay Varma; Aravind J; Indra Gandhi K. "Analysis of Supervised and Unsupervised Deep Learning Approaches for Identifying and Localizing Image Forgeries". International Research Journal on Advanced Science Hub, 5, Issue 05S, 2023, 15-25. doi: 10.47392/irjash.2023.S003

15. Yu, Mo, and Mark Dredze. "Learning composition models for phrase embeddings." Transactions of the Association for Computational Linguistics 3 (2015): 227-242.

16. Zhou, Zhihao, Lifu Huang, and Heng Ji. "Learning Phrase Embeddings from Paraphrases with GRUs." Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora. 2017.

17. Kiros, Ryan, et al. "Skip-thought vectors." Advances in neural information processing systems. 2015.

18. Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." International conference on machine learning. 2014.

19. Levy, Omer, and Yoav Goldberg. "Linguistic regularities in sparse and explicit word representations." Proceedings of the eighteenth conference on computational natural language learning. 2014.

20. Schnabel, Tobias, et al. "Evaluation methods for unsupervised word embeddings." Proceedings of the 2015 conference on empirical methods in natural language processing. 2015.

21. Ratnaparkhi, Adwait. "A maximum entropy model for part-of-speech tagging." Conference on Empirical Methods in Natural Language Processing. 1996.

22. Yadav, Vikas, and Steven Bethard. "A Survey on Recent Advances in Named Entity Recognition from Deep Learning models." Proceedings of the 27th International Conference on Computational Linguistics. 2018.

23. Liu, Bing. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press, 2015.

24. Cho, Kyunghyun, et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.

25. Finkelstein, Lev, et al. "Placing search in context: The concept revisited." Proceedings of the 10th international conference on World Wide Web. 2001.

26. Hill, Felix, Roi Reichart, and Anna Korhonen. "Simlex-999: Evaluating semantic models with (genuine) similarity estimation." Computational Linguistics 41.4 (2015): 665-695.

27. Gerz, Daniela, et al. "SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity." Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

28. Bojanowski, Piotr, et al. "Enriching word vectors with subword information." Transactions of the Association for Computational Linguistics 5 (2017): 135-146.

29. Joulin, Armand, et al. "Bag of Tricks for Efficient Text Classification." Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 2017.

30. A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jegou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," arXiv preprint arXiv:1612.03651 (2016).

31. Salle, Alexandre, Aline Villavicencio, and Marco Idiart. "Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.

DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

32. Bullinaria, John A., and Joseph P. Levy. “Extracting semantic representations from word co-occurrence statistics: A computational study.” Behavior research methods 39.3 (2007): 510-526.

33. Speer, Robyn, and Catherine Havasi. “Representing General Relational Knowledge in ConceptNet 5.” Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). 2012.

34. Ganitkevitch, Juri, Benjamin Van Durme, and Chris Callison-Burch. “PPDB: The paraphrase database.” Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013.

35. R. Speer and J. Chin, “An ensemble method to produce high-quality word embeddings.” arXiv preprint arXiv:1604.01692 (2016).

36. Speer, Robyn, Joshua Chin, and Catherine Havasi. “Conceptnet 5.5: An open multilingual graph of general knowledge.” Thirty-First AAAI Conference on Artificial Intelligence. 2017.

37. Berardi, Giacomo, Andrea Esuli, and Diego Marcheggiani. “Word Embeddings Go to Italy: A Comparison of Models and Training Datasets.” IIR. 2015.

38. Makrai, Márton, et al. “Comparison of distributed language models on medium-resourced languages.” XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2015) (2015).

39. Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors.” Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014.

40. Ghannay, Sahar, et al. “Word embedding evaluation and combination.” Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). 2016.

41. Schwenk, Holger. “CSLM-a modular open-source continuous space language modeling toolkit.” INTERSPEECH. 2013.

42. Levy, Omer, and Yoav Goldberg. “Dependency-based word embeddings.” Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014.

43. Dhillon, Paramveer S., et al. “Two step CCA: a new spectral method for estimating vector models of words.” Proceedings of the 29th International Conference on International Conference on Machine Learning. 2012.

44. Collobert, Ronan, et al. “Natural language processing (almost) from scratch.” Journal of machine learning research 12.Aug (2011): 2493-2537.

45. Lebret, Rémi, and Ronan Collobert. “Word Embeddings through Hellinger PCA.” Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. 2014.

46. Li, Ping, Trevor J. Hastie, and Kenneth W. Church. “Very sparse random projections.” Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006.

47. Pakhomov, Serguei VS, et al. “Corpus domain effects on distributional semantic modeling of medical terms.” Bioinformatics 32.23 (2016): 3635-3644.

48. Wang, Yanshan, et al. “A comparison of word embeddings for the biomedical natural language processing.” Journal of biomedical informatics 87 (2018): 12-20.

49. Agirre, Eneko, et al. “A Study on Similarity and Relatedness Using Distributional and WordNet- based Approaches.” Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2009.

50. Kliegr, Tomáš, and Ondřej Zamazal. “Antonyms are similar: Towards paradigmatic association approach to rating similarity in SimLex-999 and WordSim-353.” Data & Knowledge Engineering 115 (2018): 174-193.

51. Peters, Matthew E., et al. “Deep Contextualized Word Representations.” Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018.

DOI:10.46647/ijetms.2023.v07i05.007 ISSN: 2581-4621

52. Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019.