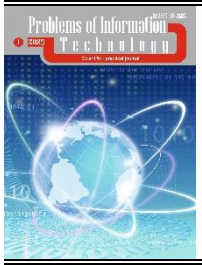

Available online at www.jpit.az

16 (1)
2025

Experimental comparison of word embedding models for fake news detection

Jalal Mehdiyev[1,2]

[1]Azerbaijan Technical University, H. Javid ave., 25, AZ1073 Baku, Azerbaijan

[2]Institute of Information Technology, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

jalal.mehdiyev.s@gmail.com

https://orcid.org/0009-0005-1386-9025

ARTICLE INFO

Keywords:

Fake news detection
Word embeddings models
Contextualized embedding models
Support Vector Machine

ABSTRACT

Detecting fake news in text data is a challenging task in today's world of generative AI, where it helps third parties generate fake news on a single request, making this problem increasingly relevant. Word embedding has been shown to be an effective model for solving classification problems. The purpose of this study and the experiments conducted is to find the strengths and weaknesses of each embedding model to solve the classification problem of finding falsified data. We consider three categories: traditional models (TF-IDF, LSA), predictive models (Word2Vec, GloVe, FastText), and contextualized models (BERT). The assessment is carried out using a test on three datasets - LIAR, ISOT, and COVID-19. In order to achieve a fair experiment, a single SVM classification method was chosen. The models are compared based on the metrics Accuracy, F1-score, and CPU time. The results of the research will help in selecting the efficient algorithm for researchers.

1. Introduction

We are spending more and more time on social media. Statistics show that every second a huge amount of information is published on social networks. Facebook Instagram statistics include 56,000 Facebook posts, 5,700 Twitter tweets, and 1,000 Instagram posts. The number of people who are looking for information about news on social media is growing, which is due to the fact that the creators of this content do not check the information they publish. This is justified by the fact that searching for information on social networks is much easier than combing through inconvenient news sites. The purpose of fake information is often to mislead people, gain clicks or advertising revenue, as well as influence public opinion.

Disinformation has become a major issue and an important area of research in the modern digital age. The consequences of disinformation can be dramatic, leading to confusion, conflict, and sometimes violence. In recent years, disinformation is connected to events, such as the spread of false information during the 2016 US presidential election and misleading claims about COVID-19 and 5G networks. In each case, there was a spread of misinformation and conspiracy theories.

A critical aspect of detecting fake news is representing text data and preparing the input for a machine learning classification model. How well a classification model can detect fake news depends on how much information it can extract from the text and recognize from the features. Embedding models have evolved very quickly. An

Received 7 October 2024, Received in revised form 9 December 2024, Accepted 23 December 2024
http://doi.org/10.25045/jpit.v16.i1.03

embedding word model converts text into number representation and compares them to find patterns. This process is similar to how a library organizes books, allowing users to find what they are looking for more quickly.

The purpose of this work is to compare the existing features of word embeddings. This will be achieved by conducting an experimental comparison using existing datasets that contain fake information. Initially, it is necessary to conduct an experiment on data analysis and the results. Definition of conditions and the negative side: Each of the methods that facilitate the task in the appropriate, necessary cases can be embedded depending on the datasets.

The rest of the paper is organized as follows. Section 2 provides a literature review. Popular word embedding models are reviewed in Section 3. In Section 4, we discuss the details of the selected datasets, evaluation metrics, setup, and experimental results.

2. Related work

Fake news detection is an actively developed problem in natural language processing. There is also increasing attention to the optimization and development of word embedding methods. Early methods are based on linguistic principles and traditional text representations such as Bag of Words (BoW) and TF-IDF. However, these methods often fail to capture the contextual relationships between words, limiting their effectiveness in detecting deceptive content. Recent advances in deep learning-based embeddings largely limit fake news classification to computing the encoding of semantic and syntactic dependencies in text data.

Verma et al. (2021) proposed a two-stage benchmark model called WELFake for using machine learning to detect fake news. It is based on word embedding (WE) over linguistic variables. In the first stage, the dataset is preprocessed and the language features are used to check the correctness of the news stories. In the second stage, voting classification is performed while the language feature sets are combined with WE. Experiments show that the WELFake model can correctly identify whether the news is true or false in 96.73% of cases.

Truică et al. (2023) presented a new way to find fake news using document embeddings (DOCEMBs) in the study. They suggested a standard to find out the best ways for detecting fake information. The novel approach used several machine learning methods and DOCEMBs created with either TF-IDF or word and transformer embeddings to find fake news.

Nassif et al. (2022) contributed to the area with respect to the enrichment of fake news detection in Arabic language. Their contribution is twofold: first, they built a large and diverse dataset of Arabic fake news. Second, they developed and evaluated transformer-based classifiers for fake news detection using eight state-of-the-art Arabic contextualized embedding models.

Samadi et al. (2023) suggest three classifiers, each with a different set of pre-trained models for embedding news articles in this study. After the embedding layer, we connect a single-layer perceptron (SLP), a multilayer perceptron (MLP), and a convolutional neural network (CNN). The CNN is made up of new pre-trained models like BERT, RoBERTa, GPT-2, and Funnel Transformer. We do these exercises to get the deep contextualized representation and deep neural classifications that these models offer.

Summing up the previous research we can see that, when it comes to classification accuracy, deep contextualized embeddings always outperform better than simple embeddings. When it comes to detecting fake news, hybrid models that combine word embeddings with linguistic and statistical features perform better. Document-level representations (Doc2Vec, Sentence-BERT) make it easier for models to find false information at a broader conversational level. Deep embeddings require more computational power than older methods like TF-IDF and LSA, so computational tradeoffs are still a big issue.

In this work, we will make experimental comparisons between embedding models of different types. Unlike the previous studies, which were focused mainly on the specific problems, we will examine the performance, efficiency, and trade-offs of traditional, predictive, and contextualized embeddings in the context of fake news detection.

3. Word Embedding Models

The study covers three-word embedding subcategories: traditional, static and contextual.

3.1. Traditional methods

Traditional models are based on the frequency of words used in the whole document. These are document-oriented models that read the whole

document and prepare the frequency of each word used as the numerical model to input to the ML model.

One of the first models is bag-of-words. The bag-of-words model is based on a simple technique, which is calculating the frequency of the words in the document. These frequencies are then converted into numerical values that machine learning algorithms can process and use to extract features from the text.

A widely used traditional model is TF-IDF. The TF part of the model is term frequency, which is the number of times a term appears in a set of documents. In this model the meaningless words such as "a", "the" and etc will ruin the statistics of the model. This is where inverse document frequency comes into play, which, unlike TF, minimizes the weight of frequently used words and increases the weight of words that are important but rarely used.

3.2. Static Word Embeddings

Static models, as the name suggests, are fixed forms. These models capture the meaning of a word when learning a particular corpus and do not change until new learning occurs. This can lead to problems because the same word in human language can be expressed in different contexts.

The Word2Vec model represents words as a one-dimensional vector. This model uses neighboring words in sentences to form a representation vector for a given word. According to the logic of the model, similar words will have similar vectors. Word2Vec is a 2-layer neural network. The input contains all the documents/texts in our training set. In order for the network to process these texts, they are represented in a 1-hot encoding of the words. The number of neurons present in the hidden layer is equal to the length of the embedding we want. We should mention two subtypes of this model. The Continuous Bag-of-Words Model is aimed to predict the word based on the neighbor values in the sentence. The context consists of several words before and after the current word. This architecture is called a bag-of-words model because the order of words is not important. The Continuous Skip-Gram Model predicts words in a certain range before and after the current word in a single sentence.

The next static embedding word model is GloVe. GloVe works differently than Word2Vec. Instead of predicting words, it computes all the statistics about how words appear together in a corpus. It builds connections between words based on the number of times they occur.

The following FastText model is a modification of the Word2Vec model so that it does not create vector representations of specific words but instead works with n-grams.

3.3. Contextualized Word Embeddings

BERT is a neural network transformer model from Google that is currently used by most automatic language processing tools. The model appeared in early 2018. BERT models for English are ready to work with large amounts of data. Developers can download and implement their natural language processing projects using a ready-made tool, without wasting time learning neural networks from scratch. Worth to mention that transformer-based models have a better performance on GPU rather than CPUs. BERT uses a transformer, an "attention" mechanism that learns contextual relationships between words. In its original form, the transformer includes two secondary mechanisms - an encoder that reads the input text, and a decoder that makes a prediction for the task. Since the goal of BERT is to create a language model, it only needs an encoder.

4. Experimental results

To conduct a fair experiment, the Support Vector Machine model was chosen. The main task of the algorithm is to find the most correct line, or hyperplane, dividing the data into two classes. SVM is an algorithm that receives data as input and returns such a dividing line.

To keep a fair comparison, all embedding models and the SVM classifier were run with default settings. The following parameters were used for the SVM classifier: SVC with a linear kernel and a C parameter of 1.0. No hyperparameter tuning was performed to focus only on the embedding performance.

All data was preprocessed before embedding. Each dataset was converted to lowercase and tokenized using simple space tokenization. Stopword removal was implemented using the public open source repository https://github.com/6/stopwords-json. To keep the experiment simple, no stemming or lemmatization was applied. Any missing or malformed data was removed using dropna().

The following hardware parameters were used to implement the experiment: Apple M2 with 8 cores operating at 3.50 GHz and 16 GB of RAM. The experiment was conducted on Python 3.9.

4.1. Datasets

For implementing the experiment with the aim of resolving the fake news detection three datasets that cover the problem selected. The details of the datasets are described in Table 1.

The LIAR dataset is a new one for detecting fake news. It contains 12.8 thousand short phrases that have been manually labeled by truthfulness, topic, location or context, speaker, rank, party, and past date. The dataset includes short statements from politifact.com that are more than a decade old and come from various settings. Each case has a full analysis report and links to the original documents.

Table 1. Description of datasets

Dataset	Number of samples	Number of features
LIAR (Wang et al., 2017)	12,800	13
Fighting an Infodemic: COVID-19 Fake News Dataset (Patwa et al. 2021)	10,700	2
ISOT Fake News (Ahmed et al. 2017)	44,898	3

The ISOT fake news dataset is a dataset that consists of articles scraped from various legitimate news sites and sites labeled as unreliable by Politifact.com. The ISOT dataset contains relatively bigger samples than the other two.

The COVID-19 fake news dataset contains misinformation and factual news related to the COVID-19 pandemic. Dataset contains tweets which led to the misconfusion of the people at the time of the pandemic.

4.2. Evaluation Metrics

We used three mostly used metrics for evaluation of the models.

Accuracy is the mostly effective performance metric used to evaluate a binary classification model. It quantifies the ratio of accurate predictions generated by the model to the total number of predictions made. A high accuracy score means that the model produces a significant proportion of correct predictions, while a low accuracy score indicates an excessive number of incorrect predictions. Accuracy is calculated as True Positives plus True Negatives divided by the sum of True Positives, True Negatives, False Positives, and False Negatives.

The F1 score is a performance metric that combines recall and precision to provide a comprehensive assessment of the effectiveness of a binary classification model. It determines the precision and recall harmonic means, giving each measurement equal weight. The F1-score is calculated by dividing the sum of precision and recall by the product of precision and recall.

Computational efficiency (CPU time): assesses the duration necessary for training and classifying text with each embedding technique.

4.3. Result analysis

Bert demonstrated the best results in the accuracy metric, according to our analysis of the experiment's data. This outcome can be explained by the BERT model's utilization of the text's contextual information. In spite of this, the Word2Vec model performed the best, while the static embedding model also produced a steady outcome. The utilization of the text's semantic meaning by static models explains this. It is evident from the results that conventional models like TF-IDF or LSA also produced good results comparable to the BERT model; yet, one should not be duped, as this is supported by well-structured datasets. For example, in the Covid-19 dataset model, the predominant terms used were “5G”, “results”, and “Covid”, indicating that count-based models can recognize the authenticity of sentences without relying on conceptual or semantic information.

Table 2. COVID-19 Dataset Results

Parameter	CPU Time (Sec)	Accuracy	F1-Score
TF-IDF	2.18	0.9042	0.9041
LSA	0.6	0.8855	0.8854
Word2Vec	1.3	0.8935	0.8933
GloVe	0.92	0.8192	0.8188
FastText	1.92	0.8668	0.8665
BERT	2.24	0.9542	0.9542

In terms of CPU time and computational cost, we can conclude that the most expensive model is BERT. BERT is based on the transformer model, which is computationally expensive and mostly runs on GPU. In terms of CPU time, we can safely say that static embedding models give the best results despite the loss in accuracy.

Table 3. LIAR Dataset Results

Parameter	CPU Time (Sec)	Accuracy	F1-Score
TF-IDF	62	0.6225	0.6165
LSA	56	0.6189	0.6197
Word2Vec	58	0.6330	0.6222
GloVe	58	0.6412	0.6278
FastText	61	0.6260	0.6118
BERT	65	0.6248	0.6198

Table 4. ISOT Dataset Results

Parameter	CPU Time (Sec)	Accuracy	F1-Score
TF-IDF	31.23	0.9922	0.9922
LSA	7.70	0.9700	0.9700
Word2Vec	21.76	0.9863	0.9863
GloVe	11.61	0.9665	0.9603
FastText	52.07	0.9765	0.9765
BERT	16.55	0.9952	0.9952

We should mention that each dataset has different types of features, considering this fact we should mention the embedding model with the best results for each dataset. COVID-19 Dataset (social media/textual misinformation) BERT performed best as contextualized embeddings effectively captured patterns of deceptive phrases. ISOT Dataset (Long News Articles) BERT performed well, showing that both statistical and deep learning models are suitable for detecting structured fake news. LIAR Dataset (Short Political Statements) GloVe and Word2Vec performed better, showing that traditional embeddings are better at classifying factual statements.

In the summary it's worth mentioning that, while BERT provided the most accurate result, its computational demands make it challenging for small and medium applications. TF-IDF and LSA are count-based models which are very useful with well-structured datasets, and suitable for tasks where efficiency is critical. Predictive embeddings (Word2Vec, GloVe, FastText) strike a balance between accuracy and computational efficiency. FastText, on the other hand, performed well with datasets containing infrequent words. However, due to its increased computational requirements, it is not suitable for classification tasks done on larger scales.

5. Conclusion

The impact of word embedding models on fake news detection is examined in this work. We can classify word embeddings as "contextualized," "static," or "traditional," where "contextualized" refers to context-based, "static" refers to prediction-based, and "traditional" refers to frequency-based. The findings show that modern contextual models that have just become accessible, like BERT and GPT, now perform better at spotting false information. Even though these strategies sometimes work, they are not always effective and waste time and money because they need to use a lot of computing power on small datasets. It is crucial to keep in mind the trade-offs between accuracy, computational cost and model complexity, while choosing the model.

In future research, we plan to provide more advanced research mainly focusing on transformer models. In addition, we will explore new approaches that improve contextual understanding, computational efficiency, and adaptability to changing patterns of disinformation.

References

Ahmed, H., Traoré, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. Security and Privacy, 1(1). https://doi.org/10.1002/spy2.9

Barushka, A., & Hajek, P. (2019). Review Spam Detection Using Word Embeddings and Deep Neural Networks. In: MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E. (Eds.) Artificial Intelligence Applications and Innovations. AIAI 2019. IFIP Advances in Information and Communication Technology, vol. 559. Springer, Cham. https://doi.org/10.1007/978-3-030-19823-7_28

Ghannay, S., Favre, B., Estève, Y., & Camelin, N. (2016). Word embedding evaluation and combination. In Proceedings of the Tenth International Conference on Language Resources and Evaluation, 16 (pp. 300–305).

Joseph, P., & Yerima, S.Y. (2022). A comparative study of word embedding techniques for SMS spam detection. In 14th International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 149-155). https://doi.org/10.1109/CICN56167.2022.10008245

Naseem, U., Razzak, I., Khan, S. K., & Prasad, M. (2021). A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. ACM Transactions on Asian and Low-Resource Language Information Processing, 20(5), Article 74, 1-35. https://doi.org/10.1145/3434237

Nassif, A.B., Elnagar, A., Elgendy, O., & Afadar, Y. (2022). Arabic fake news detection based on deep contextualized embedding models. Neural Computing and Applications, 34, 16019–16032. https://doi.org/10.1007/s00521-022-07206-4

Neelima, A., & Mehrotra, S. (2023). A Comprehensive Review on Word Embedding Techniques. In 2023 International Conference on Intelligent Systems for Communication, IoT and Security (pp. 538-543). https://doi.org/10.1109/ICISCoIS56541.2023.10100347

Omar, E. (2023). Towards a Self- sustained House: Development of an Analytical Hierarchy Process System for Evaluating the Performance of Self-Sustained Houses. MSA Engineering Journal, 2(2), 56-79. https://doi.org/10.21608/msaeng.2023.291864

Patwa, P., Sharma, S., Pykl, S., Guptha, V., Kumari, G., Akhtar, M. S., Ekbal, A., Das, A., & Chakraborty, T. (2021). Fighting an infodemic: COVID-19 fake news dataset. In: Chakraborty, T., Shu, K., Bernard, H.R., Liu, H., & Akhtar, M.S. (Eds.) Combating online hostile posts in regional languages during emergency situations, vol. 1402. Springer, Cham. https://doi.org/10.1007/978-3-030-73696-5_3

Rodriguez, P.L., & Spirling, A. (2021). Word Embeddings: What Works, What Doesn't, and How to Tell the Difference for Applied Research. The Journal of Politics, 84, 101-115. https://doi.org/10.1086/715162

Samadi, M., Mousavian, M., & Momtazi, S. (2021). Deep contextualized text representation and learning for fake news detection. Information Processing & Management, 58(6), 102723. https://doi.org/10.1016/j.ipm.2021.102723

Srinivasan, S., Ravi, V., Alazab, M., Ketha, S., Al-Zoubi, A.M., & Kotti Padannayil, S. (2021). Spam emails detection based on distributed word embedding with deep learning. In: Maleh, Y., Shojafar, M., Alazab, M., & Baddi, Y. (Eds.), Machine intelligence and big data analytics for cybersecurity applications, vol. 919. Springer, Cham. https://doi.org/10.1007/978-3-030-57024-8_7

Torregrossa, F., Allesiardo, R., Claveau, V., Kooli, N., & Gravier, G. (2021) A survey on training and evaluation of word embeddings. International Journal of Data Science and Analytics, 11(2), 85–103. https://doi.org/10.1007/s41060-021-00242-8

Truică, C.-O., & Apostol, E.-S. (2023). It's All in the Embedding! Fake News Detection Using Document Embeddings. Mathematics, 11(3), 508. https://doi.org/10.3390/math11030508

Verma, P.K., Agrawal, P., Amorim, I., & Prodan, R. (2021). WELFake: Word Embedding Over Linguistic Features for Fake News Detection. IEEE Transactions on Computational Social Systems, 8(4), 881-893. https://doi.org/10.1109/TCSS.2021.3068519

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C.-C. J. (2019). Evaluating word embedding models: methods and experimental results. APSIPA Transactions on Signal and Information Processing, 8, e19. https://doi.org/10.1017/ATSIP.2019.12

Wang, W.Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. In Proc. of the 55th Annual Meeting of the Association for Computational Linguistics, 2 (pp. 422–426). Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-2067

Yang, W., Li, L., Zhang, Z., Ren, X., Sun, X., & He, B. (2021). Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In Proc. of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 2048–2058). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.165