

Financial text embeddings for the Russian language: a global vectors-based approach

Kostyantyn A. Malysenko, Dmitriy Anashkin

Big Data Laboratorium, V.I. Vernadsky Crimean Federal University, Simferopol, Russia

Article Info

Article history:

Received Apr 14, 2024

Revised Jul 30, 2024

Accepted Aug 30, 2024

Keywords:

Global vectors

Linguistic embedding

Machine learning

Natural language processing

Russian language

ABSTRACT

The article presents a software implementation of the linguistic embedding method for the Russian language, based on the global vectors for word representation (GloVe) model. The GloVe method allows to obtain word vectors that reflect their semantic and syntactic properties. The resulting vector model can be used in various natural language processing (NLP) tasks, such as machine translation and text clustering. The article describes the architecture of software that implements a method similar to the GloVe algorithm for Russian-language financial texts. The mechanisms used to train the model as well as to compute word vectors are described. Testing with typical classification methods demonstrated that the developed program generates accurate vector representations of Russian-language texts, proving effective in various NLP tasks. This work is one of the first studies devoted to the software implementation of the GloVe method for the Russian language using learning algorithms based on sparse matrices. The results of this study can be used in various NLP tasks, such as machine translation and text clustering.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Kostyantyn A. Malysenko

Big Data Laboratorium, V.I. Vernadsky Crimean Federal University

Republic of Crimea, Simferopol, Russia

Email: docofecon@mail.ru

1. INTRODUCTION

In today's world of computing and artificial intelligence (AI), natural language processing (NLP) algorithms are playing a critical role, particularly in the sensitive realm of financial markets. One of the key tools in this area is linguistic embedding, a technique that represents words as numerical vectors in a multidimensional space. This allows language computer models to understand the semantic meaning of words and their context, leading to more accurate and efficient results in tasks like machine translation, text classification, and sentiment analysis [1].

Modern financial markets are highly sensitive to news, often reacting with significant asset price fluctuations. Unfortunately, this sensitivity also makes them vulnerable to manipulation. Some players may intentionally disseminate false or distorted information to influence the price of stocks or other financial instruments. This is where linguistic embedding and AI come into play. By analyzing news articles, social media posts, and financial reports, AI systems can leverage these word embeddings to detect subtle patterns and linguistic cues often employed in fabricated financial news. This can help identify inconsistencies, biased language, and manipulative narratives, ultimately protecting investors from making detrimental decisions and contributing to a more transparent and trustworthy financial ecosystem. Nevertheless, the field of NLP for the Russian language still lacks models and tools capable of addressing specialized tasks, particularly in the financial sector.

This study presents the results of an experiment on training an embedding model using a corpus of Russian-language online financial news. Financial news is central to this study as a crucial source of information for investors and traders. Analyzing such news helps us understand the current market state and predict potential developments.

Our approach leverages the global vectors (GloVe) algorithm, known for capturing both global and local word co-occurrence statistics, and trains it on a carefully curated corpus of Russian online financial news. Critically, our corpus incorporates not just general financial terms but also stock tickers and the names of prominent global and Russian companies, addressing a key limitation of existing models [2]–[4]. We detail the architecture of our model, highlighting our use of sparse matrices for efficient processing of large textual datasets.

To thoroughly evaluate the quality and effectiveness of our generated embeddings, we employ a three-pronged approach. First, we utilize dimensionality reduction techniques - t-distributed stochastic neighbor embedding (t-SNE), principal component analysis (PCA), and uniform manifold approximation and projection (UMAP) to visualize the embedding space, revealing semantic relationships between words and providing insights into the model's ability to cluster related terms. Second, we calculate semantic similarity scores using the Otiai-Barkman coefficient, quantitatively assessing the model's ability to accurately represent semantic relationships. Finally, and most importantly, we evaluate the practical value of our embeddings in a real-world NLP task: classifying financial news as fake or real. Through these evaluations, we demonstrate the significant advantages of our specialized embedding model over existing approaches, highlighting its potential to enhance a wide range of financial NLP applications.

This research work is organized as follows. The first section begins with a description of the background of the Russian segment of research in the field of creating computer representations of linguistic models of the language. This section ends with motivation and contribution to research work. Russian language embedding models and their disadvantages in the context of the study of financial texts in Russian are further discussed in the second section. The third section consistently presents the methods of visualization and verification of word embedding. In the fourth, an experimental study was conducted considering the use of the developed embedding in a real NLP problem. This ends with a conclusion stating the outcome of this research. This paper aims to:

- Evaluate Russian-language based word embedding model suitable for the financial domain;
- Train the chosen model on the preprocessed financial text corpus;
- Implement methods to assess the quality of the generated word embeddings;
- Analyze the embeddings to understand how they capture financial concepts. This might involve visualizing clusters of semantically related words and exploring how different financial terms are positioned in the embedding space;
- Develop and evaluate NLP tests that leverage the created word embeddings;
- Compare the performance of the Russian-language based financial word embedding with another approaches of text vectorization used in NLP problems.

This research will contribute to the development of advanced NLP tools for the financial sector. The created Russian word embeddings for financial texts will be a valuable resource for researchers and practitioners interested in extracting insights from financial news and documents.

2. RELATED WORK

There are many studies devoted to the creation of embeddings of the Russian language, including works based on the RusCorpora corpus, the RusVectōrēs model, and the RuWordnet lexical database. Below is an overview of some of them.

- RusCorpora is one of the largest corpora of the Russian language, containing a variety of texts, including prose, scientific texts, Internet materials, and others. Many studies use RusCorpora to train word embedding models, as it provides a wide coverage of different styles and genres of texts [2].
- RusVectōrēs is a collection of pre-trained Russian embedding models developed on the basis of various architectures such as Word2Vec, GloVe, and FastText. These models allow you to conduct semantic analysis of words and texts in Russian, as well as use them for various NLP tasks [3].
- RuWordnet is a lexical database based on the concept of WordNet, which contains semantic relationships between words in the Russian language. Some studies use RuWordnet to improve the quality of embeddings by integrating semantic information from the database [4].

Despite their utility in general NLP tasks, these general-purpose embeddings often lack the precision and domain-specific knowledge needed to effectively analyze financial texts. Financial language is characterized by unique terminology, jargon, and stylistic conventions not adequately represented in broader corpora. For example, stock tickers, company names, and financial abbreviations carry significant semantic weight in financial news but might be treated as out-of-vocabulary tokens or assigned inaccurate vectors in general-purpose embeddings.

Since the early 2020s, the rise of semantic recommendation and chat systems has propelled the discussion of representing the Russian language in a vector space within the scientific community. However, current research primarily emphasizes applied NLP problems in politics and education, neglecting the economic domain [5]–[7]. For example, the paper [5] examines the possibilities of vector text models of the Russian language in mathematical education and familiarization of students with the concept of vector and distributive semantics. At the same time, the work [7] examines the use of text embeddings in highlighting and clustering topics of political discussions, which may be important in processing and responding to public opinions in a timely manner.

Despite the increasing popularity of vector representations of the Russian language, some researchers, for example Harman Camper in [8], classify Russian as a zero-resource languages class, since it has a low base of developed algorithms, which can complicate the development of language, acoustic and recommendation systems. Nevertheless, a large number of works [9]–[15] are devoted directly to comparing the methods of vectorization and clustering of national text corpora, which are the first steps in creating a universal and restrictive language model, which can then be implemented in the most important industries and critical state processes. While research on English-language computational linguistics boasts a rich body of work, the study of Russian-language corpora and embeddings is still in its early stages. Several studies have focused on creating general-purpose Russian embeddings, leveraging resources like the RusCorpora corpus, the RusVectōrēs model, and the RuWordnet lexical database. However, these models often fall short in accurately representing the specific language used in financial texts. Moreover, the challenge of optimizing large and complex embedding models for efficient use has led to research on dimensionality reduction techniques. Popular methods like t-SNE [16], PCA [17], [18], and UMAP [18]–[23] have been explored for visualizing and simplifying embeddings while preserving essential information. This paper contributes to the field by:

- Training a specialized embedding model on a preprocessed corpus of Russian financial texts.
- Implementing and analyzing dimensionality reduction techniques for visualizing and verifying the generated embeddings.
- Comparing the performance of the specialized Russian-language financial embedding model with other text vectorization approaches commonly used in NLP.

This research aims to provide researchers and practitioners with a valuable resource for extracting insights from Russian financial news and documents, ultimately contributing to the development of advanced NLP tools for the financial sector.

3. PROPOSED METHODOLOGY

Training effective word embeddings for specialized domains like finance requires careful consideration of both linguistic and computational factors. This section outlines the methodology for visualizing, verifying, and evaluating the performance of our developed Russian-language embedding model. We present a two-pronged approach: utilizing dimensionality reduction techniques to visualize the embedding space and conducting a synthetic experiment to assess the embedding's impact on a real-world NLP task (text classification). This approach provides both qualitative and quantitative insights into the effectiveness of our model in capturing the nuances of Russian financial language.

3.1. Training word embeddings using Python

This section details the implementation of developed GloVe-inspired embedding model, highlighting the rationale behind key programming decisions, potential pitfalls encountered, and strategies employed to ensure scalability and efficiency. Our model is built upon the GloVe algorithm, chosen for its ability to capture both global and local word co-occurrence statistics, crucial for understanding the subtle relationships between terms in financial texts, including technical jargon and evolving slang. We implemented the model using Python, leveraging its extensive ecosystem of NLP and machine learning libraries like Gensim and Scikit-learn, which provided readily available tools for implementing GloVe-like algorithms, preprocessing text data, and evaluating performance. Python's active developer community and comprehensive documentation greatly facilitated the development and debugging process.

Python was chosen as the primary tool for this task for several reasons. First, Python offers a wide range of tools for working with text data, including libraries for NLP, neural networks, and machine learning. This makes it easy to implement the various models and algorithms needed to train embedding. In addition, Python has an active developer community and extensive documentation, making the development and debugging process easy. The embedding model was trained on a corpus of approximately 3,000 Russian financial news articles, scrapped from open sources, containing a diverse range of financial terms and expressions. This corpus, comprising over 2 million tokens, was preprocessed to generate a vocabulary of 35,541 unique words. We trained the embedding model by specifying the following parameters:

- Optimizer: Adam optimizer with a learning rate of 0.001.
- Loss function: Categorical cross-entropy loss, commonly used for multi-class classification tasks like word prediction.
- Number of epochs: The model was trained for 1,000 epochs.
- Hardware/software: Training was performed on a local machine with GeForce RTX 3050 graphical processing unit using the TensorFlow framework in Python.
- Embedding space: 50-dimensional vector space.

Processed 50-dimensional embedding space is not just an arbitrary choice; it is a result of balancing computational efficiency with the need to capture the complex semantic relationships within the specific domain of Russian financial texts. There are several data-related challenges could arise during implementation of embedding models with the same architecture. Firstly, the quality and size of the training corpus are crucial. Secondly, Russian text presents unique preprocessing challenges due to its morphology and slang. Robust techniques like stemming, lemmatization, and handling of special characters are necessary to ensure accurate data representation. Finally, inherent biases within the corpus reflecting societal or economic trends must be actively identified and mitigated during data selection and preprocessing to avoid biased model outputs.

Model training also presents its own set of difficulties. Training for 1,000 epochs on a relatively small dataset could result in overfitting, where the model memorizes the training data instead of learning general patterns. Implementing early stopping, regularization techniques like dropout, and experimenting with different learning rates and batch sizes can help mitigate this issue. Furthermore, training a deep learning model with a large vocabulary can be computationally expensive. Utilizing cloud computing resources or more powerful GPUs can accelerate the training process. Finally, finding the optimal hyperparameters (learning rate and number of layers) can be time-consuming and require extensive experimentation. Employing techniques like grid search or Bayesian optimization can help efficiently explore the hyperparameter space.

Financial corpora often involve large vocabularies, resulting in high-dimensional embedding spaces. Storing and processing these embeddings as dense matrices can quickly become computationally intractable. To address this challenge, we adopted a sparse matrix representation for word co-occurrence statistics. Sparse matrices, by storing only non-zero elements, significantly reduce memory footprint and speed up computations, enabling us to efficiently handle our large corpus. Through extensive benchmarking, "Scipy.sparse" was determined to be the most performant library for our needs, striking a balance between speed and memory efficiency.

To further enhance the expressiveness of our embeddings, we implemented a multi-layered neural network architecture during training. This architecture consisted of the following key components:

- Dropout: A regularization technique to prevent overfitting by randomly "turning off" neurons during training, improving the model's generalization ability.
- One-dimensional convolutional layers (Conv1D): Used to extract local features from sequences of words, automatically capturing important patterns in financial news texts.
- MaxPooling: Reduces dimensionality and identifies the most salient features extracted by the convolutional layers.
- Long short-term memory (LSTM): A type of recurrent neural network layer specifically designed to handle sequential data, allowing the model to learn dependencies between words within the context of a sentence or document.
- Dense (fully connected layers): Integrate the features learned by the previous layers and output the final word embeddings.

This combination of a GloVe-inspired approach, sparse matrix representation, and a carefully designed deep learning architecture allowed us to create rich and informative word embeddings while maintaining computational efficiency. However, it is crucial to emphasize that effectively leveraging sparse matrices requires meticulous attention to detail. Ensuring data structure consistency, choosing compatible algorithms (such as random forest and gradient boosting, which work well with sparse input), and rigorously testing the implementation are essential to avoid errors, performance bottlenecks, and inaccurate results. By addressing these challenges, our approach offers a robust and scalable solution for generating specialized word embeddings for Russian financial text analysis.

There are several advantages to this approach. Firstly, it allows you to efficiently store and process large amounts of text data, which is especially important when working with large news corps. In addition, sparse matrices allow you to take into account the importance of individual words and their interaction in the text, which contributes to the qualitative formation of embedding. This approach to embedding training is a complex method that involves the use of various layers of a neural network, as well as working with sparse matrices to represent textual data. This makes it possible to create effective and informative embeddings, as well as ensures high model performance when directly solving problems of analyzing the text of financial news.

3.2. Embedding visualization and verification

Exploring large, multidimensional datasets often requires tools for simplifying complex patterns. Dimensionality reduction techniques address this need by transforming data from high-dimensional spaces into lower-dimensional representations that are easier to visualize and interpret. Let's consider the cases that allow us to use this method when processing vector representations of the Russian language:

- Visualization: projecting high-dimensional embeddings onto a 2D or 3D plane makes it possible to see hidden structures and relationships between words.
- Data preprocessing: Reducing the number of features can improve the efficiency of machine learning algorithms by removing noisy or redundant information.
- Faster training: models with fewer parameters typically train faster.
- Visualizing word embeddings is crucial for understanding how words relate to each other in the semantic space. Several dimensionality reduction techniques can be used for this purpose:
- t-SNE allows to visualize high-dimensional data in two- or three-dimensional space [16];
- PCA allows to visualize data by projecting it onto the main components [17];
- UMAP allows to visualize data while preserving local and global structures [18].

Visualizing embeddings provides valuable insights into model behavior, allowing for the examination of semantic relationships, identification of potential errors or artifacts, and overall interpretation of the model's representation of language. For example, plotting word embeddings as points on a two-dimensional plane can reveal clusters of semantically similar words, with words of opposing meanings located further apart. This visual analysis helps assess the model's ability to accurately capture semantic relationships within the data [24]. By examining the spatial relationships between words in the embedding visualization, we can gain insights into the features the model utilizes to determine word meaning. In this study, embedding visualization served three primary purposes,

- Analysis of semantic connections between words: visualization allows you to see how words are related to each other in the semantic space;
- Identification of synonyms and antonyms: words that have similar meanings are close to each other in the visualization;
- Evaluation of the quality of the embedding model: visualization allows us to see how well the embedding model preserves the semantic relationships between words.

To accomplish this task, software modules have been consistently developed to reproduce each of the considered methods on the previously presented document containing a list of words and vector representations for the corpus of Russian-language publications based on financial news. Let's present and describe the results of using the considered methods on the previously implemented financial implementation of the Russian language. The PCA and t-SNE implementations from the Scikit-learn library, along with the UMAP library, were utilized to visually represent the 35,541-dimensional embedding. After the necessary parameters for each algorithm were configured, visualizations were generated to explore the structure of the embedding space.

The t-SNE in Figure 1, PCA in Figure 2(a), and UMAP in Figure 2(b) methods were applied sequentially to the embedding dictionary, enabling a visual inspection of the semantic proximity of individual words. Figure 1 presents a two-dimensional t-SNE visualization, while Figure 2 displays three-dimensional representations generated by PCA in Figure 2(a) and UMAP in Figure 2(b). This approach allowed for the exploration of word relationships, with the dictionary index providing a means of verifying observed patterns. For instance, clusters of semantically related words were observed in close proximity within the embedding space, particularly in the UMAP visualization, which exhibited a higher degree of clustering compared to the PCA representation. It can be concluded, that the UMAP method yields the highest density and distinctiveness of word features within the embedding. This is attributed to the fact that PCA focuses on global data patterns, potentially causing closely related words in the high-dimensional space to be dispersed across the PCA plane. Conversely, local data structures are taken into account by UMAP, resulting in closer mappings of semantically similar words [24].

To further validate the semantic relationships captured by our embedding model, the Otiai-Barkman coefficient (1) was calculated for word pairs. This coefficient, which utilizes cosine distance between word vectors, provides a measure of semantic dissimilarity [25], [26]. Given the two n-dimensional attribute vectors M and N and cosine similarity $\cos(\theta)$, Otiai-Barkman coefficient is represented as in (1):

$$k = 1 - \cos(\theta) = 1 - \frac{M \times N}{\|M\| \|N\|} = 1 - \frac{\sum_{i=0}^n M_i N_i}{\sqrt{\sum_{i=0}^n M_i^2} \times \sqrt{\sum_{i=0}^n N_i^2}} \quad (1)$$

To facilitate these calculations and enable the interactive exploration of semantic relationships, a widget was developed using the Python Tkinter library. This widget allows to select a word from a drop-down list and visualize a tag cloud of its nearest neighbors. In Figure 3, the nearest neighbors of the ticker symbol

"AAPL" (Apple Inc.) are showcased, highlighting a prominent cluster of other stock ticker symbols. This observation provides further evidence that domain-specific semantic relationships within the financial lexicon are effectively captured by our embedding model.

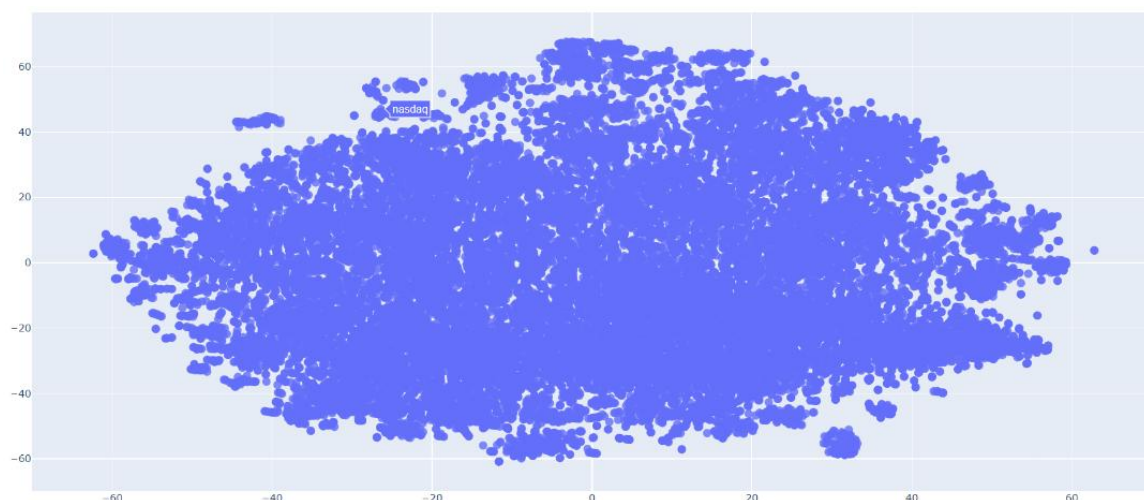


Figure 1. Two-dimensional t-SNE map of the developed embedding of the Russian language

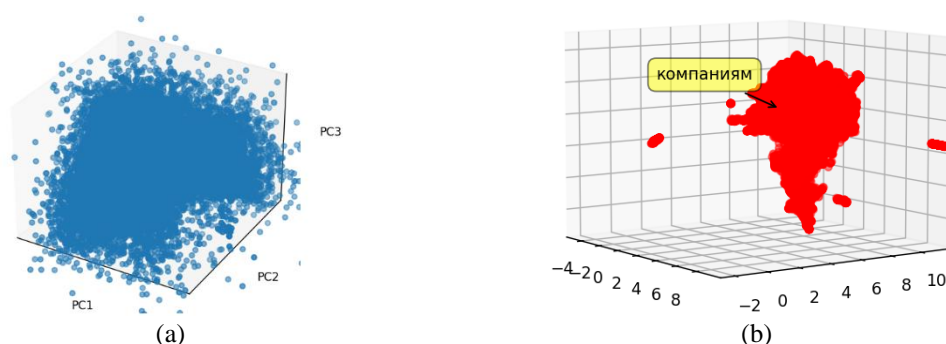


Figure 2. Three-dimensional generated by (a) PCA and (b) UMAP visualization of the original embedding



Figure 3. Financial tickers word cloud based on created linguistic embedding

4. RESULTS AND DISCUSSION

To evaluate the efficacy of our trained word embeddings in a real-world NLP scenario, we designed a binary text classification experiment: distinguishing between fabricated and authentic Russian financial news articles. This task holds particular relevance in today's information landscape, as the proliferation of misinformation can significantly impact financial markets. The experiment was divided into several main parts:

- Preparation of synthetic data. For this stage of the study, a synthetic dataset was prepared containing samples of generated news texts, as well as real news related to the stock market and the stock exchange. The test experimental dataset contained 100 elements, including the names of securities tickers, as well as names and news indicators related to well-known global and local Russian brands and government organizations. For training, the test sample was divided into training (0.8) and validation (0.2) sets.;

- The second stage of the work involved the use of built-in Python language methods for preprocessing and tokenization of text data for their subsequent comparison with the developed vector space of the Russian language with financial topics. To implement the classification task, the preprocessed data array received labels identifying the news as fake with the label "FAKE", as well as real, pre-verified with the label "TRUE". When implementing the classification task, the classification labels were scaled to the interval (0, 1) for computer labeling, respectively. To achieve our goals, we used the Scikit-learn library and the TfidfVectorizer method;
- The next stage of the experiment involved choosing a classification method according to the type of incoming data. To provide a comparative baseline, we employed four widely-used classification algorithms: random forest, decision tree, K-nearest neighbors, and gradient boosting. To implement the appropriate methods, taking into account the prepared 50-dimensional vector space, the corresponding methods from the Scikit-learn library were consistently applied to the input vectorized dataset;
- The final stage of comparing and interpreting the results was the calculation of the F1 metric for each type of classification, as well as the construction of the receiver operating characteristics (ROC) curve and the calculation of the area under the curve (AUC) indicator, which is responsible for the accuracy of the classification model and characterizes the area on the graph located directly under the learning curve.

To visually compare the performance of the four classification methods (random forest, decision tree, K-nearest neighbors, and gradient boosting), we plotted their respective ROC curves and calculated their corresponding AUC values. Figure 4 presents the ROC curves for the classifiers trained on the original term frequency-inverse document frequency (TF-IDF) representation of the data. While Figure 5 displays the results for the classifiers utilizing our 50-dimensional word embeddings.

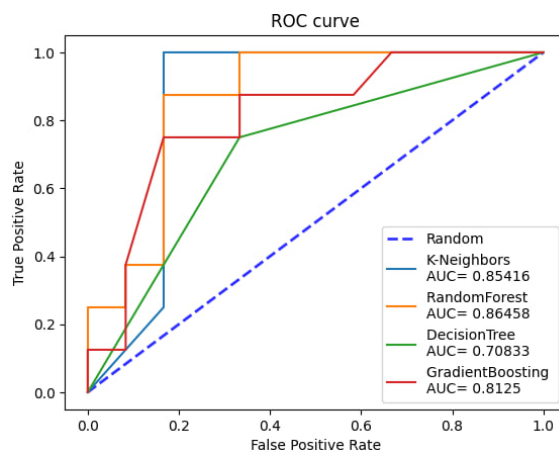


Figure 4. ROC curve for classification models based on original dataset (TF-IDF)

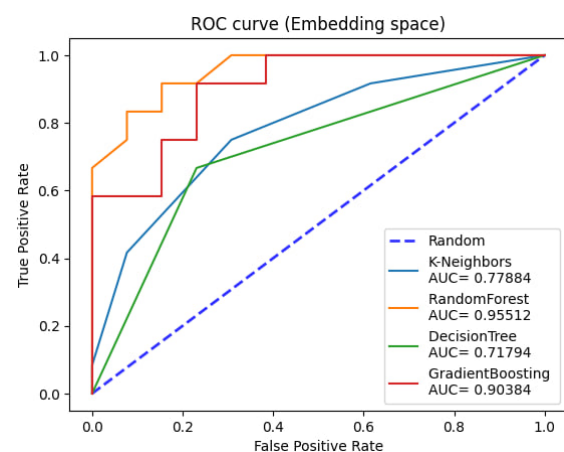


Figure 5. ROC curve for classification models trained on embedding space using tokenizer

As evident from the ROC curves, using our word embeddings for text representation resulted in a noticeable improvement in the performance of random forest and gradient boosting algorithms. Specifically, the F1-score for random forest increased by 9.054%, while gradient boosting showed a 9.134% improvement as shown in Table 1. This suggests that our embeddings capture the nuances of financial language more effectively than a generic TF-IDF approach, leading to better discrimination between fake and real financial news.

Table 1. F1 measure indicator for each classification method before and after applying the vector space

Classifier	F1-score for typical classification	F1-score for classification using word embedding
K-Neighbors	0.9	0.68
Random Forest	0.8	0.9
Decision Tree	0.75	0.72
Gradient Boosting	0.75	0.85

To quantitatively compare the performance of each classification method before and after applying our specialized word embeddings, we used the F1-score as the primary evaluation metric. Table 1 presents the F1-scores achieved on the validation set (X_{test} , y_{test}) using both the standard TF-IDF representation and our 50-dimensional embeddings as input. The F1-score was calculated using the `clf.score(*args)` function from

the scikit-learn library, where `clf` represents the trained classification model. The Table 1 shows a comparison of the F1 measure indicator for each classification method before and after applying the vector space. F1 measure was calculated using the `clf.score(X_test, y_test)` construct from the scikit-learn library, where “`clf`” means the selected classification model and “`X_test`”, “`y_test`” are validation samples.

Thus, it can be noted that in comparison with the basic classification model, the embedding weights can significantly increase its accuracy for such models as gradient boosting and random forest. In the study, where the method of textual classification of news was applied on the newly developed text vector space of the Russian language, true positive (TP) and true negative (TN) were used to compare various classification algorithms, assess the impact of various parameters on the quality of the model, as well as limit the optimal classification threshold. A scheme illustrating the process of solving the classification problem using pre-trained embeddings for text tokenization is shown in Figure 6. This scheme highlights the core components of our approach and how the pre-trained embeddings contribute to the overall effectiveness of the classification system. The process begins with a raw input text, which in our case would be a Russian financial news article. This raw text is first passed through a preprocessing stage, which typically involves tasks like tokenization (splitting the text into individual words or punctuation marks), lowercasing, and removing stop words (common words like “a,” “the,” “is” that carry little semantic meaning).

Following preprocessing, the tokenized text is fed into the embedding layer. This is where our pre-trained 50-dimensional embeddings come into play. Each word in the tokenized text is mapped to its corresponding 50-dimensional vector representation. These vectors, having been trained on a massive corpus of financial text, encapsulate rich semantic information about each word within the financial domain.

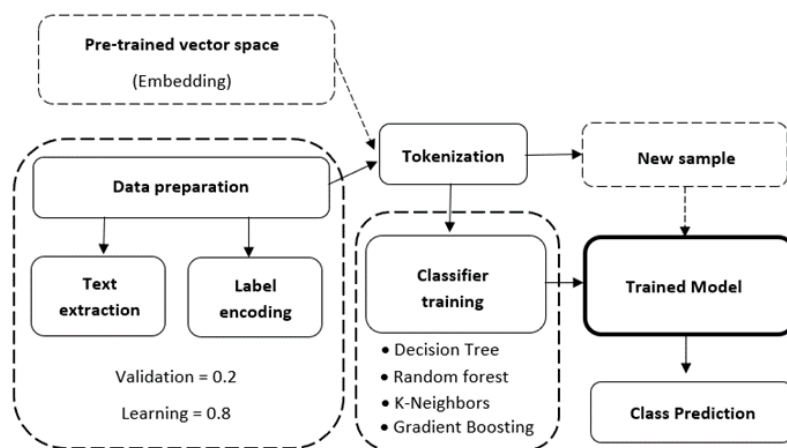


Figure 6. Text classification algorithm using pre-trained embedding

Our experimental results strongly suggest that ensemble learning methods, particularly gradient boosting and random forest, are highly effective for text classification tasks when coupled with our specialized word embeddings. These algorithms, known for their ability to capture complex interactions among features, seem to benefit significantly from the rich semantic information encoded in the 50-dimensional embedding space. This finding aligns with the broader trend in NLP, where pre-trained embeddings have become instrumental in boosting the performance of various downstream tasks.

The observed improvement in classification accuracy can be attributed to the pre-marked vector space serving as a powerful mechanism for tokenization. Unlike traditional methods like TF-IDF, which treat words as isolated units, our embeddings encode the semantic relationships between words based on their co-occurrence patterns in a vast financial corpus. This allows the classification models to “understand” the context of words within financial news articles more effectively, leading to more informed and accurate predictions.

This research contributes valuable insights into the development and application of specialized word embeddings for Russian, specifically targeting the underexplored domain of financial text analysis. Existing research on Russian language embeddings often focuses on general-purpose applications or topics like politics and education, neglecting the unique challenges posed by financial texts. This work directly addresses this gap by creating and rigorously evaluating embeddings tailored for the specific linguistic features of Russian financial discourse.

The methodological choices made throughout this study demonstrate a sound understanding of both linguistic and computational considerations. Choosing GloVe as the foundation is well-justified, as its ability to capture both global and local word co-occurrence statistics is crucial for understanding the subtle

relationships between financial terms, including technical jargon and evolving slang. This is further enhanced by the use of a multi-layered neural network architecture, showcasing a well-designed combination of components like convolutional layers, LSTM, and dropout, known for their effectiveness in capturing complex patterns in sequential data. Addressing the computational challenges posed by large vocabularies in financial corpora, the authors' adoption of a sparse matrix representation for word co-occurrence statistics is commendable. This choice significantly improves computational efficiency without sacrificing model expressiveness, enabling the processing of large financial datasets, a crucial factor for real-world applications.

The paper goes beyond simply creating embeddings by employing a comprehensive evaluation framework. Utilizing dimensionality reduction techniques like t-SNE, PCA, and UMAP provides valuable visual insights into the embedding space, confirming the model's ability to cluster semantically related terms, including stock tickers and company names. This qualitative analysis is further strengthened by the quantitative assessment using the Otai-Barkman coefficient, showcasing the model's effectiveness in capturing semantic similarity. The real-world value of the developed embeddings is convincingly demonstrated through the text classification experiment. The significant improvement in accuracy when classifying fake vs. real financial news, particularly for models like gradient boosting and random forest, highlights the practical benefits of using these specialized embeddings in a real-world NLP application.

Furthermore, the inclusion of company names and stock tickers directly into the embedding vocabulary is a noteworthy contribution. This approach allows the model to capture crucial contextual information often overlooked by general-purpose embeddings, leading to a richer and more nuanced understanding of financial texts. In conclusion, this research provides a strong foundation for future work on Russian financial text analysis. The developed embeddings, combined with the rigorous evaluation methodology presented, offer valuable resources for researchers and practitioners seeking to leverage the power of NLP in the financial domain.

5. CONCLUSION

This research introduces a novel approach to address the scarcity of specialized word embeddings for Russian financial texts. We developed a 50-dimensional embedding model, trained on a curated corpus, that effectively captures semantic relationships within this domain. Our model, inspired by GloVe, leverages sparse matrix representations and a multi-layered neural network architecture to efficiently handle a large vocabulary and capture complex word relationships. To validate our model's effectiveness, we conducted a binary text classification experiment on fake and real financial news. Our embeddings, when used as input features for ensemble learning algorithms, significantly outperformed a standard TF-IDF baseline, demonstrating their ability to encode subtle semantic nuances crucial for accurate fake news detection. Our work offers a valuable resource for the NLP community and has the potential to enhance various financial applications, including sentiment analysis, risk assessment, and algorithmic trading. Future research will focus on expanding the dataset, exploring different embedding dimensions and neural network architectures, and investigating the model's applicability to other financial NLP tasks.

ACKNOWLEDGEMENTS

The research was carried out at the expense of the grant of the Russian Science Foundation No. 23-28-00946: <https://rscf.ru/en/project/23-28-00946/>.




REFERENCES

- [1] M. G. Shishaev, "Neural network models in semantic analysis of natural language texts," *Proceedings of the Kolsky Science Center of the Russian Academy of Sciences*, vol. 11, no. 8–11, pp. 91–100, 2020, doi: 10.37614/2307-5252.2020.8.11.008.
- [2] I. M. Boguslavsky *et al.*, "Constructing a semantic corpus for Russian: SemOntoCor," *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2023"*, pp. 1–14, 2023, doi: 10.28995/2075-7182-2023-22-12-25.
- [3] A. Bolshina and N. Loukachevitch, "All-words word sense disambiguation for Russian using automatically generated text collection," *Cybernetics and Information Technologies*, vol. 20, no. 4, pp. 90–107, 2020, doi: 10.2478/cait-2020-0049.
- [4] L. Usmanova *et al.*, "Analysis of the semantic distance of words in the RuWordNet thesaurus," *Data Analytics and Management in Data Intensive Domains: 22nd International Conference, DAMDID/RCDL 2020*, Voronezh, Russia, vol. 22, pp. 60–73, 2020.
- [5] M. P. Kontsevov, "Service of semantic calculations Rusvectōrēs in modern mathematical education," *Web-programming and Internet technologies WebConf2021: materials of the 5th International Scientific and Practical Conference*, Minsk, Belarusian State University, 2021, pp. 256–256. Accessed: Oct. 12, 2024. [Online]. Available: <https://elib.bsu.by/bitstream/123456789/261736/1/256.pdf>
- [6] S. D. Belov, P. V. Zrellov, A. V. Ilyina, V. V. Korenkov, and V. A. Tarabrin, "The use of neural network language models to study the demand for professional competencies of higher education in the labor market," *System Analysis in Science and Education*, vol. 3, pp. 13–25, 2023. Accessed: Oct. 12, 2024. [Online]. Available: <https://sanse.ru/index.php/sanse/article/view/585>
- [7] A. Indukaev, "Studying ideational change in Russian politics with topic models and word embeddings," *The Palgrave Handbook of Digital Russia Studies*, 2020, pp. 443–464, doi: 10.1007/978-3-030-42855-6_25.
- [8] H. Kamper, Y. Matusevych, and S. Goldwater, "Multilingual acoustic word embedding models for processing zero-resource




- languages,” *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6414–6418, 2020, doi: 10.1109/ICASSP40776.2020.9054202.
- [9] O. Korogodina *et al.*, “Evaluation of vector transformations for russian static and contextualized embeddings,” *Conference on Computer Graphics and Machine Vision: GraphiCon*, vol. 2021, pp. 349–357, 2021.
- [10] A. A. Antipenko and O. A. Mitrofanova, “Comparative study of word associations in social networks corpora by means of distributional semantics models for Russian,” *International Journal of Open Information Technologies*, vol. 8, no. 1, pp. 27–33, 2020.
- [11] O. Mitrofanova *et al.*, “Topic modelling of the russian corpus of pikabu posts: author-topic distribution and topic labelling,” *International Conference “Internet and Modern Society” (IMS-2020). CEUR Proceedings*, pp. 101–116, 2020.
- [12] P. Zhang, V. P. Zakharov, “Computerized visualization of the Russian language picture of the world,” *International Journal of Open Information Technologies*, vol. 8, no. 1, pp. 58–63, 2020.
- [13] D. V. Borodaenko and A. S. Pogudina, “Comparison of methods of vectorization of texts with preservation of semantic proximity,” *Electronic Scientific Journal ‘Diary of Science’*, vol. 5, pp. 24–24, 2020. Accessed: Oct. 12, 2024. [Online]. Available: http://dnevniknauki.ru/images/publications/2020/5/technics/Borodaenko_Pogudina.pdf
- [14] E. Artemova, “Deep learning for the Russian language,” *The Palgrave Handbook of Digital Russia Studies*, Palgrave Macmillan, Cham: Springer, 2021, pp. 465–481, doi: 10.1007/978-3-030-42855-6_26.
- [15] D. O. Zhakysbaev and G. N. Mizamova, “Natural language processing algorithms for understanding text semantics,” *Transactions of ISP RAS*, vol. 34, pp. 141–150, 2022, doi: 10.15514/ISPRAS-2022-34(1)-10.
- [16] S. D. Erokhin, B. B. Borisenko, I. D. Martishin, and A. S. Fadeev, “Analysis of existing methods for reducing the dimensionality of input data,” *T-Comm - Telecommunications and Transportation*, vol. 16, no. 1, pp. 30–37, 2022, doi: 10.36724/2072-8735-2022-16-1-30-37.
- [17] W. Song, L. Wong, P. Liu *et al.*, “Improved t-SNE based manifold dimensional reduction for remote sensing data processing,” *Multimedia Tools and Applications*, vol. 78, pp. 4311–4326, 2019, doi:10.1007/s11042-018-5715-0.
- [18] E. Becht, L. McInnes, J. Healy *et al.*, “Dimensionality reduction for visualizing single-cell data using UMAP,” *Nature Biotechnology*, vol. 37, pp. 38–44, 2019, doi:10.1038/nbt.4314.
- [19] V. O. Malich and S. A. Nesterov, “Dimensionality reduction methods for data analysis tasks,” *System Analysis in Design and Control*, vol. XXVI, no. 3, pp. 437–443, 2023, doi: 10.18720/SPBPU/2/id23-510.
- [20] F. Krasnov, “Comparative analysis of the accuracy of methods for visualizing the structure of a collection of texts,” *International Journal of Open Information Technologies*, vol. 9, no. 4, pp. 79–84, 2021.
- [21] N. Migenda, R. Möller, and W. Schenck, “Adaptive dimensionality reduction for neural network-based online principal component analysis,” *PLoS ONE*, vol. 16, no. 3, 2021, doi: 10.1371/journal.pone.0248896.
- [22] J. Li and Y. Wang, “nPCA: a linear dimensionality reduction method using a multilayer perceptron,” *Frontiers in Genetics*, vol. 14, 2024, doi: 10.3389/fgene.2023.1290447.
- [23] M. A. Salam, A. T. Azar, M. S. Elgendy, and K. M. Fouad, “The effect of different dimensionality reduction techniques on machine learning overfitting problem,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 4, pp. 641–655, 2021, doi: 10.14569/IJACSA.2021.0120480.
- [24] T. A. Litvinova, P. V. Panicheva, E. S. Kotlyarova, and V. V. Zavarzina, “Visualizing embeddings to study gender-related differences in word meaning,” *International Journal of Open Information Technologies*, vol. 10, no. 11, pp. 47–53, 2022.
- [25] E. O. Chernousov and N. S. Chikunov, “Research and development of an intelligent decision support system for remote technical support based on word-embedding methods,” *International Scientific Journal ‘Innovative Science’*, no. 12, pp. 66–70, 2017. Accessed: Oct. 12, 2024. [Online]. Available: <https://cyberleninka.ru/article/n/issledovanie-i-razrabotka-intellektualnoy-sistemy-podderzhki-prinyatiya-resheniy-dlya-sluzhby-udalennoy-tehnicheskoy-podderzhki-na/pdf>
- [26] O. N. Polshchikova, “The scope of the concept of “terminology of computational linguistics”,” *Vestnik of the Mari State University*, vol. 17, no. 1, pp. 89–94, 2023, doi: 10.30914/2072-6783-2023-17-1-89-94.

BIOGRAPHIES OF AUTHORS



Kostyantyn A. Malysenko    began his professional career in September 1995 as an assistant in the Department of Industrial Economics at the East Ukrainian State University, where he worked until December 1996. After completing his postgraduate studies at the East Ukrainian State University (from December 1996 to November 1999), he worked as an assistant and senior lecturer in the Department of Enterprise Economics at the East Ukrainian National University named after Vladimir Dal from December 1999 to June 2002. Since September 2002 and up to the present day, he has been employed at the Humanitarian Pedagogical Academy (Branch) of the Crimean Federal University named after V.I. Vernadsky in Yalta. He is the author of over 130 scholarly works and instructional materials. He can be contacted at email: docofecon@mail.ru.



Dmitriy Anashkin    holds a Bachelor in Education (Mathematics (B.Ed.)). He has published over 19 papers from 2020 to 2024. He participated in the competition for the best projects of fundamental research in the field of socio-political sciences, conducted jointly by the Russian Foundation for Basic Research and the EISI (2020) with project No. 20-011-31388 opn as the main performer. The performer of Russian Science Foundation Project No. 23-28-00946. Up to the present day, he has been a student at the Humanitarian Pedagogical Academy (Branch) of the Crimean Federal University named after V.I. Vernadsky in Yalta. His research areas of interest include mathematics, artificial intelligent, informatics, and social media research. He can be contacted at email: anashkin.dima30.08@gmail.com.