

# Learning Word Embeddings using Lexical Resources and Corpora

Ranka Stanković<sup>1</sup>[0000-0001-5123-6273], Jovana Rađenović<sup>1</sup>[0000-0002-2707-3870],  
Mihailo Škorić<sup>1</sup>[0000-0003-4811-8692] and Marko Putniković<sup>2</sup>[0000-0003-2040-796X]

<sup>1</sup> University of Belgrade, Faculty of Mining and Geology, Đušina 7, 11000 Belgrade, Serbia

<sup>2</sup> University of Belgrade, Program for Multidisciplinary Research, Studentski trg 1, 11000 Belgrade, Serbia

**Abstract.** Learning word embeddings on large, unlabeled corpora has proven effective for many natural language tasks. However, these representations can be further improved by incorporating external lexical resources. Previous research has demonstrated that lexical vector representation (embeddings, e.g., Dict2vec) trained on both text and lexical data (e.g., WordNet and/or monolingual dictionaries) gives improved results for English. The existence of the Serbian Wordnet and several Serbian electronic dictionaries enables testing this approach for Serbian within this project. In this paper, we adapt the original Dict2vec project for Serbian language resources. We present the textual, lexical, and vector resources prepared and used for training and evaluation, describe the training pipeline, and discuss preliminary evaluation results. We conclude this paper by outlining ongoing work and future steps.

**Keywords:** Word Embeddings, Dictionaries, Lexical Resources, Word2Vec, Dict2vec.

## 1 Introduction

### 1.1 Motivation

The early twenty-first century's proliferation of textual data and increased computing power spurred deep learning research, leading to the transformer architecture (Vaswani et al. 2017) in Natural Language Processing (NLP). Using pre-trained word embeddings, such as those derived from Word2vec or BERT, enables transfer learning, which is especially beneficial for tasks with limited training data. Pre-trained models often lead to improved performance in diverse NLP applications. The application of these embeddings allows models to grasp semantic relationships more effectively than traditional methods, enhancing their ability to process and understand language. Transformers employ encoders for text analysis and decoders for synthesis. Encoder-only models, often referred to as static models, have achieved breakthroughs in text vectorization, word annotation, and classification (e.g., BERT by Devlin et al. 2018, RoBERTa by Liu et al. 2019, and DeBERTa by He et al. 2020). In parallel, decoder-only models like GPT (Generative Pre-trained Transformer) (Radford et al. 2018, 2019) popularized generative tasks through their ability to produce coherent and context-aware language. Encoder-decoder models like BART (Lewis et al. 2020) and T5 (Raffel et al. 2020), while powerful for text transformation, have seen comparatively less widespread adoption.

Learning Word Embeddings using Lexical Resources refers to the process of enhancing word representations by integrating information from lexical databases and semantic networks. This approach aims to improve the effectiveness of word embeddings—dense vector representations of words—in various NLP tasks, such as sentiment analysis, machine translation, and information retrieval. The significance of this topic lies in its potential to provide a richer semantic context, which can lead to a more accurate understanding and processing of language by computational models.

Word embeddings are essential for many NLP tasks. Traditional embeddings (e.g., those obtained via Word2vec) rely on large corpora but lack external linguistic knowledge. Strong semantic relations between words, such as synonymy or meronymy, rarely occur within the same context window, making them difficult to capture in word embeddings. Lexical resources (e.g., WordNet, dictionaries) provide structured word relationships.

Lexical resources, such as WordNet and other semantic networks, play a pivotal role in this enhancement by defining relationships among words, including synonyms, antonyms, and contextual associations. Researchers have developed techniques like retrofitting and "sprinkling," which involve adjusting pre-trained embeddings to better reflect the information derived from these lexical databases. Such methods have demonstrated improved performance in semantic similarity tasks compared to embeddings learned solely from large text corpora, indicating the value of incorporating external lexical knowledge into embedding models (Srinivasan et al 2019; Wang et al. 2019; Rodriguez and Spirling 2022). Our goal is not only to train static word embeddings, but to improve Serbian word embeddings using lexical resources.

## 1.2 Research Questions

To assess the effectiveness of word embeddings enhanced with lexical resources, various evaluation metrics are utilized, including semantic similarity. Research has shown that, in specific semantic tasks, embeddings derived from semantic networks can outperform those generated from large text corpora (e.g., Word2vec). Saedi et al. (2018) underscored the importance of integrating lexical knowledge into embedding models, as it allows for a more comprehensive representation of word meanings. For the Serbian language, such experiments have not yet been conducted; therefore, in this paper, we test the research question:

**(RQ1):** *Can word vector representations learned from a corpus benefit from lexical resources?*

Although word embeddings can effectively capture language variations, they often struggle with rare words that are well-covered in traditional dictionaries. Ruzzetti et al. (2022) demonstrated that definitions from traditional dictionaries can be used to generate better word embeddings for these rare words. Therefore, we pose a research question to test whether the same holds for the Serbian language:

**(RQ2)** *Can dictionary definitions improve embeddings for rare words?*

## 2 Literature Overview

Word embeddings are typically learned through various techniques that can be broadly categorized into frequency-based and prediction-based methods. Each method has its

unique approach to generating word representations that capture semantic meaning and relationships between words. Frequency-based embeddings are derived from the statistical properties of words in a corpus, focusing on their occurrence and co-occurrence with other words. The TF-IDF (Term Frequency-Inverse Document Frequency) method evaluates the importance of a word in a document relative to a collection of documents, emphasizing words that are frequent in a specific context but rare overall. It generates vectors that reflect the significance of terms based on their distribution across documents. The Co-occurrence Matrix approach counts how often words appear together within a specified context window, capturing relationships based on proximity, and helps form a co-occurrence matrix that can be used to derive embeddings.

Prediction-based Word Embedding Techniques utilize neural network models to predict contextual relationships between words, providing richer semantic representations and are better at capturing nuances such as synonyms and polysemy. Word2vec is a pioneering model that employs a shallow neural network to predict neighboring words in a context, effectively generating embeddings that encode semantic similarity (Mikolov et al. 2013). FastText model represents words as a sum of their constituent character n-grams, enabling it to effectively generate embeddings for out-of-vocabulary words and capture morphological similarities often ignored by word-based models (Bojanowski et al. 2017). A more novel model, BERT, generates embeddings by considering the full context of words within sentences, significantly improving performance on various NLP tasks (Devlin et al. 2018). GPT is a transformer-based architecture that generates embeddings and is effective for various generative tasks (Radford et al. 2018, 2019).

Learning word embeddings on large, unlabeled corpora has proven effective for many natural language tasks, yet these representations can be further improved by incorporating external lexical resources. The main problem with Word2vec is the lack of supervision during the learning process. There have been many attempts to introduce supervision, such as the use of knowledge graphs. Unlike knowledge graphs, which demand high resources, dictionaries and lexicons are easier to employ and can fit general purposes. Dict2vec (Tissier et al. 2017) solves the problem by establishing strong and weak supervision, with data drawn from monolingual dictionaries and, in our case, thesauri as well.

Transformer models initially reached Serbian via multilingual versions, the latter trained with significant Serbian data and remaining a key encoder model, described in (Škorić 2024). While experimental generative models like procesaur/gpt2-srlat (Škorić 2023) also emerged, encoder model development continued with domain-specific versions like JelenaTosic/SRBerta for law (Bogdanović et al. 2024) and a RoBERTa-adapted QA model (Cvetanović and Tadić 2023, 2024). The larger generative models, jerteh/gpt2-orao<sup>1</sup> (800M parameters) and jerteh/gpt2-vrabac<sup>2</sup>, trained on a ~4 billion token Serbian corpus with dual-alphabet support (Škorić 2024).

More recent efforts include retraining large English models for Serbian, such as Alpaca-based models and classla/xlm-rbertic (Ljubešić et al. 2024), a retrained XLM-RoBERTa-large. Concurrently, the dataset used for the Jerteh GPT2 models was

---

<sup>1</sup> <https://huggingface.co/jerteh/gpt2-orao>

<sup>2</sup> <https://huggingface.co/jerteh/gpt2-vrabac>

leveraged to train two new encoder models from scratch: Jerteh-355<sup>3</sup> (RoBERTa-large, 355M parameters) and Jerteh-81<sup>4</sup> (RoBERTa-base, 81M parameters), designed with high-quality corpora (Škorić 2024).

The Serbian NLP landscape features numerous multilingual and around twenty monolingual models, varying in architecture, parameters, training data, and target tasks, though complete information is not always available or verifiable. This paper focused on evaluating selected static encoder models. Specifically, it analyzed the performance of the newly developed static word embedding models<sup>5</sup> on similarity tasks, comparing word embeddings that include lexical resources in the training process (as explained in the Methodology section) with others. Generative models are not a primary focus due to challenges in reliable automatic evaluation, and dedicated Serbian encoder-decoder models are still emerging.

Celli (2021) proposes an algorithm Lex2vec<sup>6</sup>, to enhance the interpretability of pre-trained word embeddings by assigning human-readable labels to their dimensions. The Lex2vec methodology takes existing word embeddings (e.g., Word2vec) and a lexical resource (like NRC) that maps words to linguistic or conceptual labels. For each word in the embedding dictionary, the algorithm checks its corresponding label(s) in the lexical resource. It then iterates through each dimension of the word's embedding vector; if a dimension's normalized value is above a certain threshold, indicating a strong association, the word's label(s) are mapped to that dimension. It was concluded that Lex2vec is suitable for explainability, acknowledging the need for strategies to filter or rank labels to balance coverage and clarity, especially with larger lexical resources.

Liang et al. (2021) introduced Anchor & Transform<sup>7</sup> (ANT), an efficient embedding algorithm designed to overcome the scalability issues of traditional methods with large vocabularies. ANT learns a small set of "anchor" embeddings and represents individual object embeddings as their sparse linear combinations, automatically determining the optimal number of anchors.

### 3 Methodology

In this paper, we are presenting static word embeddings developed within TESLA (Text Embeddings - Serbian Language Applications) project, of three architectures: Word2Vec, FastText, and Dict2vec, all freely available on HuggingFace platform<sup>8</sup> and first versions of these architectures developed on smaller datasets<sup>9</sup>.

Word2vec operates in two main architectures: Continuous Bag of Words (CBOW) and Skip-gram. The CBOW model predicts a target word based on its surrounding context words, essentially learning to fill in a blank given the words before and after it,

---

<sup>3</sup> <https://huggingface.co/jerteh/Jerteh-355>

<sup>4</sup> <https://huggingface.co/jerteh/Jerteh-81>

<sup>5</sup> <https://huggingface.co/te-sla>

<sup>6</sup> <https://github.com/takealook77/lex2vec/>

<sup>7</sup> [https://github.com/pliang279/sparse\\_discrete](https://github.com/pliang279/sparse_discrete)

<sup>8</sup> <https://huggingface.co/te-sla>

<sup>9</sup> <https://github.com/putnich/dict2vec>, <http://llo.d.jerteh.rs/putnich/>

by combining the vectors of these context words. Conversely, the Skip-gram architecture reverses this logic: given a single input word, it attempts to predict the words that are likely to appear in its surrounding context. Both approaches train a shallow neural network to map words to dense vector representations (embeddings), such that words that frequently appear in similar linguistic contexts within the training corpus are positioned closely together in the resulting high-dimensional vector space, capturing semantic relationships (Mikolov et al. 2013).

GloVe (Global Vectors for Word Representation) is a word embedding model developed to explicitly analyze and model the properties underlying the fine-grained semantic and syntactic regularities observed in vector representations (Pennington et al. 2014). Presented as a global log-bilinear regression model, GloVe aims to synthesize the advantages of both major model families: global matrix factorization and local context window methods (like Word2Vec). It efficiently leverages statistical information from the entire corpus by training only on the non-zero elements of a word-word co-occurrence matrix, directly relating word vectors through their co-occurrence probabilities. This approach allows GloVe to produce a vector space with meaningful substructure and beneficial linear properties, demonstrating strong performance, often outperforming related models, on standard tasks such as word analogy, word similarity, and named entity recognition.

Word2vec (local context) and GloVe (global co-occurrence) learn word embeddings through different objectives. Word2vec might excel slightly in capturing local semantic nuances, while GloVe is explicitly designed for capturing linear relationships based on overall word distribution. For many practical applications, both provide high-quality embeddings, and the choice might come down to available pre-trained models, computational resources, and specific project requirements rather than a definitive "better" model for all tasks.

FastText addresses the limitation of traditional word embedding models that ignore word morphology by representing each word as a bag of character n-grams (Bojanowski et al. 2017). Instead of assigning a distinct vector to each whole word, FastText learns vector representations for these n-grams, and a word's overall vector is the sum of its constituent n-gram vectors. This innovative approach enables the model to generate meaningful representations for rare or out-of-vocabulary words, train efficiently on large corpora, and achieve state-of-the-art performance on word similarity and analogy tasks across multiple languages by effectively capturing subword information.

To enhance word embeddings beyond what can be learned from unlabeled corpora alone, Dict2vec introduces a method that utilizes natural language dictionaries—one of the most extensive and refined sources for word descriptions (Tissier et al. 2017). While existing approaches often use external data that only cover a limited portion of the vocabulary, Dict2vec leverages dictionary entries to generate new word pairs. This process is designed to bring semantically related words (as defined in dictionaries) closer together in the embedding space, while negative sampling is employed to separate word pairs that are not related according to dictionary definitions. The effectiveness of Dict2vec's resulting word representations is then assessed on multiple datasets for both word similarity and text classification tasks.

Dict2vec model extends the Skip-gram model with a corpus created from Wikipedia dump. Definitions are sourced from online dictionaries, used in supervised learning

setup by generating strong and weak word pairs from definitions. Weak pairs can be promoted to strong if both words are among the  $K$  closest neighbors based on cosine similarity in pretrained word embeddings. In this way, it moves semantically related words closer in vector space.

In this paper, we are using dictionary definitions for the Serbian language and synonym data to adjust the model to learn synonyms. In Section 4, we explain Dict2vec implementation that relies on Dict2vec Project<sup>10</sup>. We also explain how we obtained dictionary definitions and how we created word pairs in data collection and processing. There, we also describe corpora used and the process of PoS tagging and lemmatizing the corpus. In Dict2vec project adaptation we describe which modifications we made to the original project. In experiments, we compared Word2vec and Dict2vec in terms of synonyms recognition by looking into the closest vectors.

Estimating semantic similarity between textual data remains a challenging and open research problem in NLP because the versatility of natural language makes it difficult to establish effective rule-based methods. We present how Dict2vec can be used to extract more synonyms from the closest neighboring words. Chandrasekaran et al. (2021) traced the evolution of various approaches developed to tackle this issue, from traditional techniques like kernel-based methods<sup>11</sup> to contemporary transformer-based models. They systematically categorized these methods—based on their underlying principles—as knowledge-based, corpus-based, deep neural network-based, and hybrid methods. By discussing the strengths and weaknesses of each, the survey provides a comprehensive overview of existing systems. In this paper, the similarity is measured as the cosine similarity between calculated vectors. The evaluation of synonym list for given word will be given in Section 5 for all developed static vectors. To assess word similarity, the standard method involves calculating Spearman's rank correlation coefficient between human-assigned similarity scores for word pairs and the cosine similarity of their corresponding word vectors; a score near 1 signifies that the embeddings closely align with human judgment. Following the protocol of Word2vec and fastText, pairs containing words not present in the embeddings were discarded; since all models were trained on the same corpora, they shared identical vocabularies and thus the same out-of-vocabulary rates.

## 4 Solution

In the context of the Serbian language, rich lexical dictionaries provide an invaluable source of semantic information that is often underutilized. In this paper, we propose a novel approach—SerbDict2vec<sup>12</sup>—that leverages Serbian dictionary definitions to construct additional word embeddings, thereby refining the semantic space. By extending the Skip-gram model's objective function with weighted dictionary-derived word pairs, our method moves semantically related words closer together, which is

---

<sup>10</sup> <https://github.com/tca19/dict2vec>

<sup>11</sup> Kernel-based methods, such as string or sequence kernels, were employed to detect patterns within text data, thereby allowing for the measurement of similarity between different pieces of text.

<sup>12</sup> <https://huggingface.co/te-sla/SerbDict2vec>

especially useful when corpus data is sparse. Definitions are sourced from several dictionaries, internal resources, and AI-generated entries (cleaned and lemmatized), but mostly from SrpWN (Krstev et al. 2004) and Systematic Dictionary<sup>13</sup>.

The corpus is POS tagged and lemmatized (Stanković et al. 2020, Škorić et al. 2023). Stop words (conjunctions, exclamations, abbreviations, prepositions, pronouns, etc) were removed from the corpus. The option to manually add additional strong and weak pairs was used. The full source code and language model are provided to foster further research in Serbian word representation. Building on the approach introduced in Dict2vec (Tissier et al. 2017), which leverages dictionary definitions to enhance word embeddings by aligning semantically related word pairs, our work adapts this framework for the Serbian language. Similar to Dict2vec, we integrate lexical information from Serbian dictionaries into the training process, using weighted word pairs to refine the semantic space.

The original corpus had 150,913,211 words, and after removal, it had 69,333,173 words, from which we obtained vocabulary that will be labelled as SerbWikiVoc with 74,315 different words.

Definitions in Serbian from 5 dictionaries were used. The main resource was the Serbian WordNet with more than 28,000 synsets, where 17,192 were mapped with SerbWikiVoc. With Wiktionary 14,975 was mapped, with online dictionary 22,158, generated with language models (Gemini and GPT4o) and partially manually checked 29,723, GPT4o 10,957 entries, Termini database and internal glossaries collected from various web sources, and translated ~20,000. The definitions were lemmatized and filtered by removing the words that were not present in the vocabulary SerbWikiVoc from the definition's text. The stop words enlisted in stopwords.txt file were also eliminated, as well as words of length one and two characters.

The next step was to generate strong pairs (if two words are in the definitions of each other, ~740,000 pairs) and weak pairs (if one word is in the definition of another, but not the opposite, ~830,000 pairs), using the base pre-trained model for words in the SerbWikiVoc vocabulary. These embeddings were then used to compute the K nearest neighbors of selected words based on cosine similarity, and the resulting pairs were stored as strong pairs.

For the base pre-trained model, we used Word2VecSr<sup>14</sup>, trained on a large Serbian language corpus of nearly ten billion words that consists of five datasets:

- Kišobran<sup>15</sup> - umbrella web corpus of Serbian and Serbo-Croatian, as the largest aggregation of web corpora so far, suitable for training Serbian large language models (Škorić and Janković 2024). For this research, only the Serbian subset (8.7 billion words) was used.
- ZNANJE<sup>16</sup> - South Slavic Scientific Research publications, highly curated, high-quality, and diverse scientific publications with over 4.2 billion words

---

<sup>13</sup> Sistematski rečnik srpskohrvatskog jezika (1938), Ranko Jovanović [https://sr.wikisource.org/sr-el/Систематски\\_речник\\_српскохрватског\\_језика\\_\(1936\)](https://sr.wikisource.org/sr-el/Систематски_речник_српскохрватског_језика_(1936)), TEI-Lex version <https://github.com/putnich/sr-sh-nlp>

<sup>14</sup> <https://huggingface.co/te-sla/Word2VecSr>

<sup>15</sup> <https://huggingface.co/datasets/procesaur/kisobran>

<sup>16</sup> <https://huggingface.co/datasets/procesaur/ZNANJE>

(Škorić and Janković 2024). For this research, only the Serbian subset (700 million words) was used.

- Vikipedija<sup>17</sup> - South Slavic corpus based on Wikipedia dumps. For this research, only the Serbian subset (150 million words) was used.
- Vikizvornik<sup>18</sup> - South Slavic corpus based on Wikisource dumps. For this research, only the Serbian subset (13 million words) was used.
- SrpELTeC<sup>19</sup> corpus of Serbian novels published for the first time in the period 1840-1920, digitized within COST ACTION CO16204: Distant Reading for European Literary History<sup>20</sup>, with more than 5.3 million words (Stanković et al. 2022).

Once all required resources were prepared and the strong and weak pairs were generated, we proceeded with training the SerbDict2vec model. The values of hyperparameters we used for training were set to those found in the literature. We use 5 negative samples, 5 epochs, a window size of 5 words, and a vector size of 100. The coefficients  $\beta_s$ ,  $\beta_w$ , ns, and nw, which control the contribution of strong and weak pairs, were set to 0.8, 0.45, 4, and 5, respectively. The number of threads used was set to 8. The real training time for the model was 10 minutes and 51 seconds. All experiments were run on a Consumer-level quad-core processor.

## 5 Evaluation and Discussion

Automatic synonym extraction is vital for many natural language processing systems. While word embeddings capture semantic relatedness, they often fail to distinguish true synonymy from other semantic relationships. We choose synonym extraction for the evaluation, since it is a fundamental research, which is helpful to text mining and information retrieval. We were inspired by Al-Matham and Al-Khalifa (2021) development of SynoExtractor, a pipeline for Arabic synonym extraction using Word2vec. They used filtering of similar word embeddings based on specific linguistic features (lemma, PoS, collocation filters) to precisely identify synonyms.

We evaluated SerbDict2vec against different variations of the base model, Word2VecSr, including a CBOW model named *TeslaW2V*, SkipGram model named *TeslaSG*, and two additional variants trained on lemmatized text: *TeslaW2Vleme* and *TeslaSGleme*. For the evaluation, aside from these variants, we also test the GloVe model *GloVeSr*<sup>21</sup> and FastText model *FastTextSr*<sup>22</sup>, which were trained on the same corpus, and an older Word2vec model for Serbian, *word2vec-sh-wiki*, which was trained on a Wikipedia dump only.

The analysis results shown in Figure 1 evaluate the presence of synonyms within the top n similar words retrieved by different models, with n ranging from 5 to 50. A carefully chosen set of words and their synonyms was used, each word having been

<sup>17</sup> <https://huggingface.co/datasets/procesaur/Vikipedija>

<sup>18</sup> <https://huggingface.co/datasets/procesaur/Vikizvornik>

<sup>19</sup> <https://huggingface.co/datasets/jerteh/SrpELTeC>

<sup>20</sup> <https://www.distant-reading.net/>

<sup>21</sup> <https://huggingface.co/te-sla/GloVeSr>

<sup>22</sup> <https://huggingface.co/te-sla/FastTextSr>

confirmed by at least two Serbian synonym lexicons and possessing at least 3 synonyms. We calculated the percentage of synonyms found in the retrieved list of similar words for  $n$  values of 5, 10, 15, 20, 25, 30, 40, and 50. The most successful architectures were the Word2vec model trained on lemmatized text (TeslaW2VLeme) and Dict2vec. Results show that Dict2vec is superior for  $n=5$  and  $n=10$ , performance is almost identical at  $n=15$ , but for  $n$  values of 20, 25, and 30, the lemmatized Word2vec model performs best. Finally, it should be noted that it was not verified whether all synonyms for the selected words were present in the model dictionaries.

Model	% for top 5	% for top 10	% for top 15	% for top 20	% for top 25	% for top 30	% for top 40	% for top 50
word2vec-sh-wiki	14,0	17,1	19,8	21,71	23,62	24,94	26,34	28,89
TeslaSG	15,1	20,8	25,5	26,91	28,51	30,35	32,59	34,78
TeslaW2V	17,8	22,8	26,1	28,66	30,93	33	34,94	37,38
TeslaSGleme	11,0	14,9	16,4	17,6	18,93	19,64	20,62	22,55
TeslaW2VLeme	20,8	28,4	31,3	35,01	38,36	39,82	43,15	45,62
SerbDic2vec	22,8	29,2	31,9	33,99	35,05	36,41	37,3	38,16
GloVeSr	9,7	12,7	15,3	17,06	18,17	19,86	22,44	24,58

Figure 1. The percentage of synonyms retrieved by different models for similar words in the range 5 to 50

Figure 2 graphically presents the synonym retrieval percentages, grouped by model. This figure also clearly shows the dominance of the SerbDict2vec model for a smaller number of retrieved similar words, while the TeslaW2VLeme model was the best option for a larger number of retrievals.

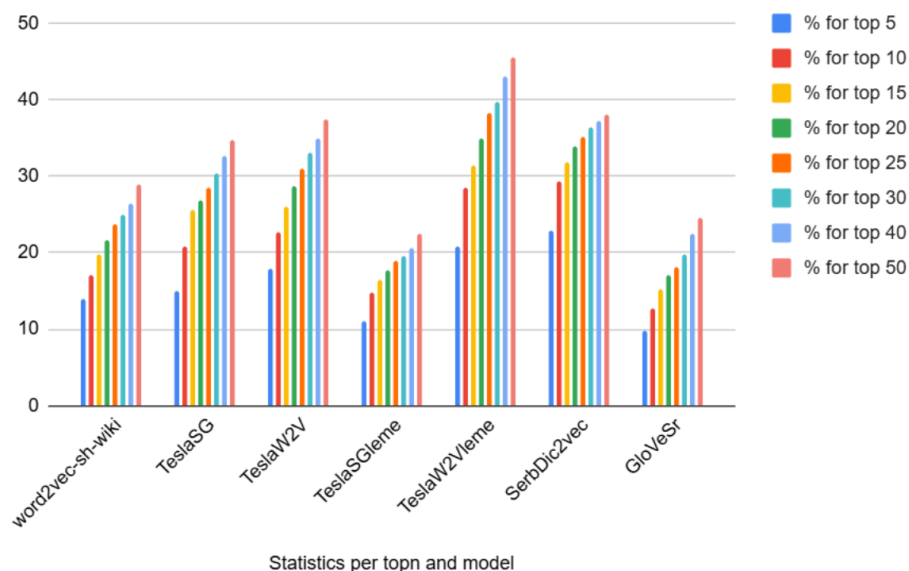


Figure 2. The histogram of synonyms for models and the retrieved similar words

The FastText model was deemed unsuitable for inclusion in this analysis. Its observed limitations included a relatively slow processing speed and generating representations for out-of-vocabulary terms, often producing similar words to correct ones, but that contained incorrect character combinations. Consequently, FastText results were excluded from the subsequent presentation and detailed analysis.

It is pertinent to highlight the fundamental distinction in how GloVe and Word2vec models learn word representations. Word2vec models operate predictively, learning word vectors by forecasting words based on their local context within a predefined window (Skip-gram) or predicting a target word from its surrounding context (CBOW); their focus is on capturing the probabilistic relationships of words appearing nearby. Conversely, GloVe is a count-based model designed to directly leverage global word-word co-occurrence statistics derived from the entire corpus, training vectors such that their dot products are proportional to the logarithm of their co-occurrence probabilities, thereby emphasizing semantic relationships derived from overall distributional patterns.

While both models produce dense word vectors where similar words are close in the vector space, the exact structure of that space and the relationships captured can differ subtly. The Word2vec, particularly Skip-gram with negative sampling, is often argued to capture more fine-grained semantic relationships and perform slightly better on tasks requiring distinguishing between subtly different word usages based on local context. GloVe is explicitly designed to capture linear relationships (like king - man + woman = queen) very well because it models the ratio of co-occurrence probabilities, which relates directly to these types of linear transformations in the vector space. Neither standard Word2vec nor standard GloVe handles out-of-vocabulary (OOV) words or morphological variations particularly well, as they learn a single vector per unique word string. FastText addresses this with subword information, but for our task, that was not important.

In practice, for many downstream NLP tasks (such as text classification, named entity recognition when used as features), the performance difference between using GloVe and Word2vec vectors as features is often not dramatic. The quality of the training corpus, the vector size, and other hyperparameters can have a bigger impact than the choice between these two specific algorithms. The Word2vec for capturing local context information and fine-grained semantic relationships based on proximal word occurrences; for preferring the simplicity of the predictive training approach if one is implementing from scratch or understanding the basics; Skip-gram is often favored if analogy tasks are particularly important for your evaluation. The GloVe is used for very large, static corpora to efficiently incorporate global co-occurrence statistics. The pre-computation of the co-occurrence matrix can be faster than iterating over the entire corpus multiple times, as Word2vec does.

Additional evaluation was originally intended to be conducted on a set of 200 synonym pairs derived from a newly created set of synsets prepared for integration into SrpWN (Krstev et al., 2004), with cosine similarity as the evaluation metric. However, due to vocabulary coverage limitations, the effective evaluation set was reduced. A box plot analysis of synonym similarities is presented for all tested models, except for the

word2vec-sh-wiki model. This model was excluded because 63 out of 200 synonym pairs contain at least one word that is not present in its vocabulary. The absence of certain specifically Serbian words can likely be attributed to the fact that this model was trained on a Serbo-Croatian corpus. Including this model would significantly reduce the evaluation set, thereby affecting the comparability of results. As for the remaining models, the number of synonym pairs in which at least one word is missing from a model's vocabulary is as follows: 5 for TeslaSG, 5 for TeslaW2V, 14 for TeslaSGleme, 3 for TeslaW2Vleme, and none for Dict2vec and GloVe. To ensure a fair comparison, the intersection of synonym pairs available across all remaining models was used, resulting in 182 pairs.

Figure 3 shows the box plot distributions of cosine similarities for synonym pairs across the evaluated models. The Dict2vec model demonstrates the highest median similarity values, which suggests it captures synonym relationships more effectively than the other models.

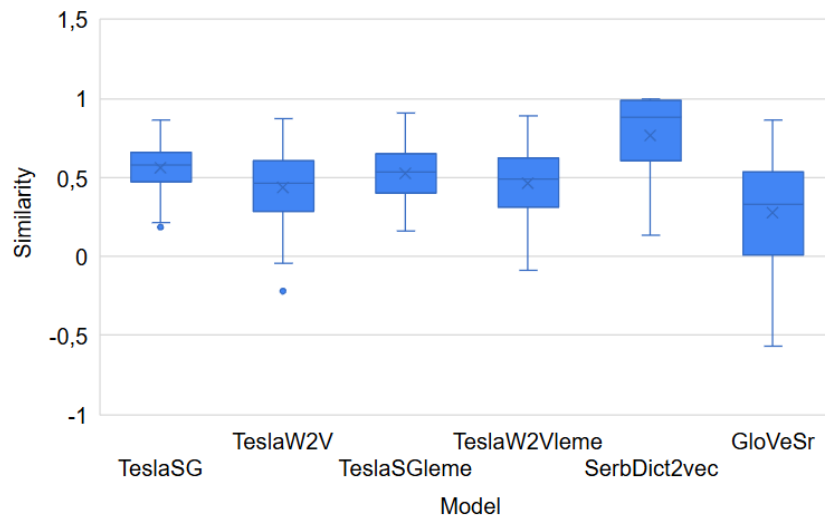


Figure 3. Distribution of Synonym Similarities for Different Models

It is worth noting that, during the development of our model, various forms of similar architectures were tested in order to assess how architectural and training data variations affect embedding quality. For this purpose, evaluation was conducted on the entire set of 200 synonym pairs referenced earlier. The evaluation metric used was cosine similarity. In this stage 3 models are compared. All models were trained using variations in the architecture and training data. The w2v model represents the Word2vec which uses the Skip-gram architecture. The d2v is the Dict2vec model which extends w2v and uses pretrained word embeddings from w2v in the training process. As mentioned, the proposed SerbDict2vec model is a Serbian-specific variant of Dict2vec, which uses pretrained word embeddings from Word2VecSr.

Figure 4 presents the distribution of synonym similarities for the three models. As can be seen from the box plots, the SerbDict2vec model demonstrates the highest average similarity and the most compact distribution, implying that the integration of linguistic resources and specialized Serbian embeddings (Word2VecSr) helps produce more accurate and consistent representations of synonyms.

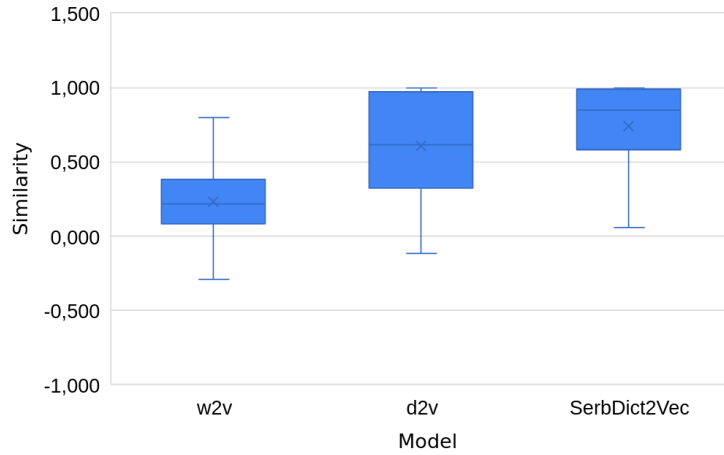


Figure 4. Distribution of Synonym Similarities for Different Models

Figure 5 illustrates the number of word pairs (out of 200) with negative similarity scores across the three models. The w2v model produced 31 such cases, while d2v yielded only 6. SerbDict2vec returned no negative similarity values, indicating a more consistent semantic space. These results further confirm that SerbDict2vec captures semantic relatedness more effectively than the other models.

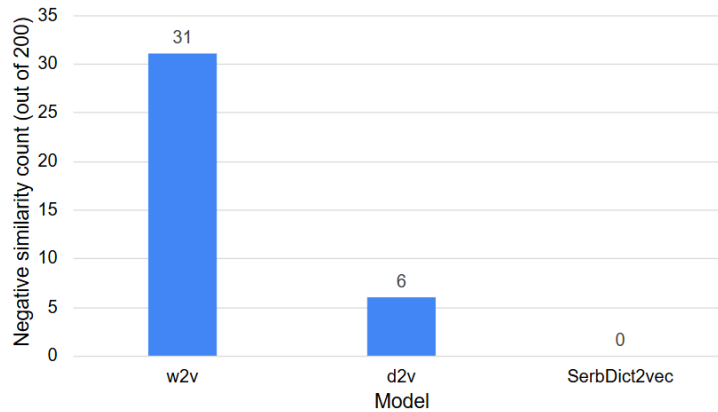


Figure 5. Negative similarity

The first practical use of the presented models is the retrieval of more synonyms from the closest neighboring words and assigning similarity measures to existing synonym pairs. This will be used by lexicographers in the compilation of a new, corpus-based Dictionary of the Contemporary Serbian Language (Stijović et al. 2024).

## 6 Conclusion and Future Research

This study described the development and evaluation of the performance of several static word embedding models developed within the TESLA project. It was shown that *SerbDict2vec* and *TeslaW2Vleme* models demonstrated superior performance on word similarity tasks compared to traditional corpus-only embeddings and other resource-enhanced techniques. This finding supports the notion that integrating dictionary definitions can effectively be used to build improved word embeddings, particularly benefiting the representations of rare words (for *SerbDict2vec*). As expected, the lemmatization (based on lexicons) was also an important step for better results for this task.

Currently, efforts are focused on conducting a more detailed quantitative evaluation, including performance on text classification tasks. The comparison for a small set of samples will be done with Kontekst<sup>23</sup>, which was produced only from web corpora. Building upon the initial lower-case approach, we are also developing a case-sensitive version of the model. For future research directions, we plan to explore advanced neural architectures for definition integration. This includes adapting approaches like Definition Neural Network (DefiNNet) and Define BERT (DefBERT) (Ruzzetti et al., 2022), which have shown promise in effectively handling unknown words by incorporating definitional information.

As highlighted by previous work (Roy and Shimei, 2021), integrating external knowledge into word embeddings presents challenges, notably concerning its impact on preserving the linear relationships and compositionality inherent to distributional semantic models. Future work should delve into these issues using theory-guided methods, novel visualization techniques, and simulations to gain clearer insights into the effects of knowledge augmentation on the embedding space. These advancements aim to further enhance the utility of dictionary-enhanced embeddings for various Serbian NLP applications, including lexicography.

### Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, #7276, Text Embeddings – Serbian Language Applications – TESLA.

---

<sup>23</sup> <https://www.kontekst.io/srpski>

## References

1. Al-Matham, R.N., & Al-Khalifa, H.S. (2021). SynoExtractor: A Novel Pipeline for Arabic Synonym Extraction Using Word2Vec Word Embeddings. *Complex.*, 2021, 6627434:1-6627434:13.
2. Bogdanović, M., Kocić, J., and Stoimenov, L. (2024) SRBerta-A Transformer Language Model for Serbian Cyrillic Legal Texts. *Information* 15 (2). issn: 2078-2489. <https://doi.org/10.3390/info15020074>.
3. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135-146.
4. Celli, F. (2021). Lex2vec: making Explainable Word Embeddings via Lexical Resources. *arXiv preprint arXiv:2103.02269*.
5. Chandrasekaran, D., & Mago, V. (2021). Evolution of Semantic Similarity—A Survey. *ACM Computing Surveys*, 54(2), 1-37.
6. Cvetanović, A., and Tadić, P. 2023. “Synthetic Dataset Creation and Fine-Tuning of Transformer Models for Question Answering in Serbian.” In 2023 31st Telecommunications Forum (TELFOR), 1–4. IEEE.
7. Cvetanović, A., & Tadić, P. (2024). Synthetic dataset creation and fine-tuning of transformer models for question answering in Serbian. *arXiv preprint arXiv:2404.08617*. <https://arxiv.org/html/2404.08617v1>, <https://paperswithcode.com/paper/synthetic-dataset-creation-and-fine-tuning-of>
8. Devlin, J., Chang, MW., Lee, K. and Toutanova, K. (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv preprint arXiv:1810.04805*.
9. He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv preprint arXiv:2006.03654*.
10. Krstev, C., Pavlović-Lažetić, G., Vitas, D., Obradović, I. (2004) Using Textual and Lexical Resources in Developing Serbian Wordnet, in *Romanian Journal of Information Science and Technology*, vol. 7, No. 1-2, pp. 147-161.
11. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the ACL*, 7871–7880.
12. Liang, P. P., Zaheer, M., Wang, Y., Ahmed, A. (2021) Anchor & Transform: Learning Sparse Embeddings for Large Vocabularies. In: *Int. Conference on Learning Representations (ICLR)*, 1-31. <https://openreview.net/forum?id=Vd7lCMvtLqg>
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.
14. Ljubešić, N., Suhomel, V., Rupnik, P., Kuzman, T., & van Noord, R. (2024). Language Models on a Diet: Cost-Efficient Development of Encoders for Closely-Related Languages via Additional Pretraining. *LREC-COLING 2024*, 189.
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013) Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 26, pp. 3111–3119.
16. Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018) Improving language understanding by generative pre-training. *Open-AI*. [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf)
17. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.

18. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140), 1-67.
19. Rodriguez, P. L., Spirling, A. (2022) Word embeddings: What works, what doesn't, and how to tell the difference for applied research. *J. Polit.* **84**(1), 101–115.
20. Roy, A., Pan, and S. (2021) Incorporating extra knowledge to enhance word embedding. In: *Proc. 29th Int. Joint Conf. on Artificial Intelligence (IJCAI 2021)*, pp. 4929–4935.
21. Ruzzetti, E. S., Ranaldi, L., Mastromattei, M., Fallucchi, F., Scarpato, N., Zanzotto, F.M. (2022) Lacking the embedding of a word? Look it up into a traditional dictionary. In: *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2651–2662. Association for Computational Linguistics, Dublin.
22. Saedi, C., Branco, A., Rodrigues, J. A., Silva, J. (2018) WordNet Embeddings. In: *Proc. 3rd Workshop on Representation Learning for NLP*, pp. 122–131. Association for Computational Linguistics, Melbourne.
23. Stanković, R., Krstev, C., Todorović, B. Š., Vitas, D., Škorić, M., & Nešić, M. I. (2022, June). Distant reading in digital humanities: Case study on the serbian part of the eltec collection. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 3337-3345).
24. Stanković, R., Sandrih, B., Krstev, C., Utvić, M., & Škorić, M. (2020). Machine learning and deep neural network-based lemmatization and morphosyntactic tagging for serbian. In *LREC 2020-12th International Conference on Language Resources and Evaluation, Conference Proceedings* (pp. 3954-3962). European Language Resources Association (ELRA).
25. Stijević, R., Stanković, R., Škorić, M., (2024) Dictionary of Modern Serbian Language: RSSJ, South Slavic Languages in the Digital Environment JuDig, 32.
26. Škorić, M. (2023) *Композитне псеудограматике засноване на паралелним језичким моделима српског језика*. Докторска дисертација. PhD diss., Универзитет у Београду.
27. Škorić, M., Utvić, M., & Stanković, R. (2023). Transformer-Based Composite Language Models for Text Evaluation and Classification. *Mathematics*, 11(22), 4660. <https://doi.org/10.3390/math11224660>
28. Škoric, M. (2025). New Language Models for Serbian. *Infotheca - Journal For Digital Humanities*, 24(1), 7-28. doi:10.18485/2024.24.1.1
29. Srinivasan, A., Kamarthi, H., Ganesan, D., Chakraborti, S. (2019) Integrating lexical knowledge in word embeddings using sprinkling and retrofitting. In: *Proc. 16th Int. Conf. on Natural Language Processing*, pp. 115–123. NLP Association of India, Hyderabad.
30. Tissier, J., Gravier, C., Habrard, A. Dict2vec (2017) Learning word embeddings using lexical dictionaries. In: *Conf. Empirical Methods in Natural Language Processing (EMNLP 2017)*, pp. 254–263.
31. Vaswani, A., Shazeer, N. Parmar, N., Uszkoreit, J. Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017) Attention is all you need. *Advances in neural information processing systems* 30, 1-11.
32. Wang, B., Wang, A., Chen, F., Wang, Y., Kuo, C.-C. J. Evaluating word embedding models: methods and experimental results. *APSIPA Trans. Signal Inf. Process.* 8(1), e19 (2019). <https://doi.org/10.1017/ATSIP.2019.12>