Similarity Measurements for Romanian Words

Rares Dan Tiago Goia, Babes-Bolyai University, Computer Science

May 18, 2025

Contents

Abstract

Measuring semantic similarity between words is a core task in Natural Language Processing (NLP), enabling downstream applications such as information retrieval, paraphrase detection, and word sense disambiguation. This work investigates the effectiveness of various static and contextual embedding models and similarity metrics for Romanian, a lower-resourced language, by benchmarking them against human judgments from the RoSimLex-999 dataset. I present a unified framework for evaluating and visualizing Romanian word similarity, and analyze correlations between model predictions and human annotations.

1 Introduction

Semantic similarity quantifies how close in meaning two words are, which is essential for a range of NLP tasks. For Romanian, existing resources and benchmarks are scarce compared to high-resource languages. This paper describes the development and evaluation of a system that computes semantic similarity between Romanian words using several types of word embeddings and similarity metrics, evaluated against the RoSimLex-999 gold standard.

2 Task Definition and Theoretical Background

Given a pair of Romanian words (w_1, w_2), our goal is to assign a similarity score $S(w_1, w_2)$ that reflects human semantic intuition. I approach this by leveraging both static and contextual embeddings and a range of similarity metrics.

2.1 Static Embeddings and Context-Independent Similarity

Static word embeddings provide a fixed vector representation for each word in the vocabulary, independent of any surrounding context. Popular models for Romanian such as FastText and CoNLL17 are trained on large text corpora and encode information about a word's distributional semantics—its typical usage patterns and relationships to other words.

Theoretical Foundation:

- During training, each word w is assigned a vector $\vec{w}$ in a high-dimensional space. The vectors are learned such that words co-occurring with similar neighbors are placed closer together in this space.
- Models like FastText extend this by representing words as compositions of character n-grams, helping with rare and morphologically complex words—a significant benefit for Romanian.
- Once trained, each word's embedding remains constant regardless of its specific usage in a sentence.

Computing Similarity:

- The similarity between two words w_1 and w_2 is typically measured by the cosine similarity of their vectors:
$$S_{static}(w_1, w_2) = \frac{\vec{w_1} \cdot \vec{w_2}}{||\vec{w_1}||\,||\vec{w_2}||}$$
- Additional similarity measures such as Jaccard, Dice, and Levenshtein are sometimes applied, operating on either the vector representation or the string form of the words.
- In this project, the user provides two Romanian words. The system retrieves their embeddings from the selected model (e.g., FastText or CoNLL17) and computes their similarity using one or more of these metrics.

Application Example:

- The words "carte" and "manual" will have a relatively high cosine similarity, as both frequently occur in similar contexts related to education and books, regardless of the sentence.

- However, the word “carte” will always have the same embedding, whether it refers to a “book” or appears in an expression like “carte de identitate” (“identity card”), limiting the ability to distinguish between senses.

Advantages and Limitations:

- **Efficiency:** Static embeddings are fast to use, as each word maps to a precomputed vector.
- **Vocabulary Coverage:** FastText’s use of character n-grams allows for reasonable representations even for rare or unseen words, which is especially important in morphologically rich languages like Romanian.
- **Context-Independence:** Static embeddings are *not* sensitive to context. This means they can miss important differences in meaning when a word is polysemous or used idiomatically.
- **Baseline for Evaluation:** Despite their limitations, static embeddings remain a strong and interpretable baseline for many tasks, and often perform well when context effects are less pronounced.

Static embeddings thus provide an efficient and effective means of measuring general semantic similarity between Romanian words, but cannot model context-specific nuances captured by contextual embeddings.

2.2 Contextual Embeddings and Contextual Similarity

Traditional static embeddings (e.g., FastText, CoNLL17) assign a single vector to each word, regardless of its context or sense. However, many Romanian words (as in English) are polysemous—their meaning shifts depending on usage (e.g., “bancă” can mean “bank” or “bench”). This limitation is addressed by **contextual embeddings**, such as those produced by BERT models.

Theoretical Foundation:

- Contextual embeddings are generated by deep neural networks (transformers) trained on massive corpora, which output word vectors *conditioned* on the sentence or textual context in which the word appears.
- For example, the Romanian BERT model (`dumitrescustefan/bert-base-romanian-cased-v1`) takes as input a full sentence and produces embeddings for each token, with each embedding reflecting that token’s meaning in its context.
- Thus, “bancă” in the sentence “Am depus bani la bancă” (“I deposited money at the bank”) and in “Stau pe o bancă în parc” (“I am sitting on a bench in the park”) will have **different** vector representations, allowing for disambiguation based on usage.

Computing Contextual Similarity:

- To measure the similarity between two words *as they appear in context*, the system extracts their contextualized vectors from BERT using the full sentences provided by the user.
- These vectors are then compared using cosine similarity, producing a context-aware similarity score:

$$S_{contextual}(w_1, w_2; C_1, C_2) = \frac{\vec{w_1}^{C_1} \cdot \vec{w_2}^{C_2}}{||\vec{w_1}^{C_1}||\,||\vec{w_2}^{C_2}||}$$

  where $\vec{w_1}^{C_1}$ is the contextual embedding of w_1 in context C_1.

- In the implemented application, the user provides each target word along with its sentence (context). The model outputs the cosine similarity between these context-specific embeddings.
- This approach captures nuances such as semantic shift, idioms, and word sense, which static embeddings cannot distinguish.

Application Example:

- If “bancă” appears in two financial contexts, their contextual embeddings will be close, yielding high similarity.
- If “bancă” is compared across “bank” and “bench” contexts, their embeddings will diverge, resulting in low similarity—even though the *word* is the same.

Advantages:

- **Sense Disambiguation:** Captures the specific meaning of a word in a given sentence.
- **Context Sensitivity:** Differentiates word similarity depending on how words are used, which is crucial for Romanian's polysemous words and idiomatic expressions.
- **Flexibility:** Enables similarity measurements not just for isolated words, but for usage in real contexts, as encountered in actual texts or applications.

The contextual approach is a significant step toward more accurate and nuanced semantic similarity modeling, especially for a language as morphologically rich and polysemous as Romanian.

2.3 Similarity Metrics

For two word vectors $\vec{w_1}, \vec{w_2}$, we compute:

- **Cosine Similarity**: $S_{cos}(w_1, w_2) = \frac{\vec{w_1} \cdot \vec{w_2}}{||\vec{w_1}||\,||\vec{w_2}||}$
- **Jaccard Similarity**: Overlap of top-k indices in embeddings.
- **Dice Coefficient**: 2× overlap over sum of set sizes.
- **Levenshtein Similarity**: String similarity, $1 - \frac{\text{Levenshtein distance}}{\max(\text{len}(w_1),\text{len}(w_2))}$
- **RO-WordNet Path Similarity**: Shortest path between noun synsets in Romanian WordNet.

3 Dataset

I used **RoSimLex-999**, a Romanian translation and adaptation of the SimLex-999 dataset. It contains 999 word pairs, each annotated with human similarity ratings on a continuous scale.

3.1 Preprocessing

Pairs not accepted by annotators or containing invalid tokens are filtered. Only (word1, word2, human_score) are kept.

4 System Architecture

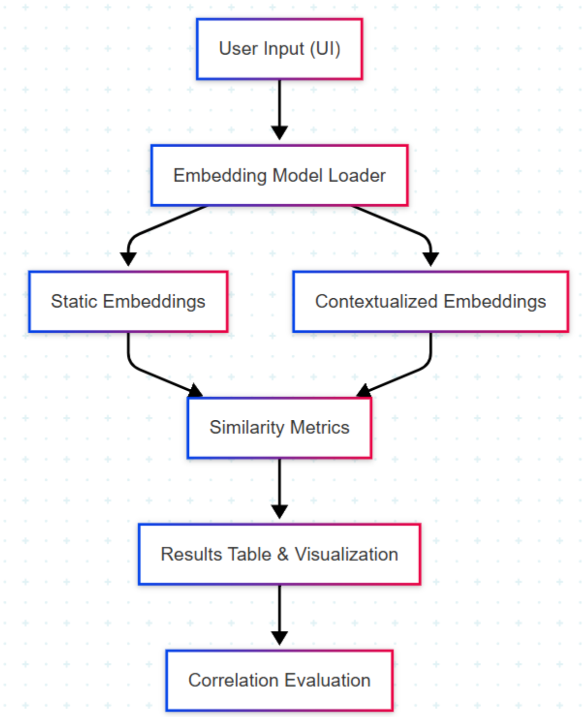

Figure 1: System overview for Romanian word similarity measurement.

5 Implementation

5.1 Libraries and Tools

- **Gensim**: FastText, CoNLL17 embedding loading.
- **HuggingFace Transformers**: Romanian BERT.
- **Scikit-learn**: Cosine similarity, t-SNE.
- **Pandas, NumPy**: Data processing.
- **Streamlit**: User interface.
- **Matplotlib, Seaborn**: Visualization.
- **python-Levenshtein**: String similarity.
- **rowordnet**: RO-WordNet path similarity.

5.2 Key Components

- `embeddings_loader.py`: Embedding/model loading and word vector lookup.
- `similarity_metrics.py`: Implements cosine, Jaccard, Dice, Levenshtein metrics.
- `rowordnet_utils.py`: WordNet path similarity.
- `evaluation.py`: Correlation computation.

- `visualizations.py`: t-SNE and scatter plots.
- `app.py`: Streamlit interface and pipeline integration.

5.3 Original Contributions

- Unified, interactive comparison of static and contextual word similarity for Romanian.
- Comprehensive metric evaluation, including lexical and data-driven approaches.
- Integration with Romanian resources (RoSimLex-999, RO-WordNet).
- Visualization tools for embedding space and metric performance.

6 Experiments and Results

For all word pairs in RoSimLex-999, I compute similarity using each metric and model, and report their Pearson and Spearman correlation with human ratings.

6.1 Overall Correlation Results

Metric	**FastText**		**CoNLL17**	
	Pearson	**Spearman**	**Pearson**	**Spearman**
Cosine	0.376	0.361	0.233	0.242
Jaccard	0.200	0.196	0.145	0.157
Dice	0.207	0.196	0.152	0.157
Levenshtein	0.004	0.011	0.004	0.011
ROWordNet Path	0.561	0.520	0.561	0.520

Table 1: Pearson and Spearman correlation between each metric/model and human similarity scores on RoSimLex-999.

6.2 Discussion

- **ROWordNet Path Similarity** achieves the highest correlation with human judgments, indicating that lexical resources remain highly valuable for Romanian word similarity.
- **Cosine similarity (FastText)** provides the best performance among embedding-based methods, but with only moderate correlation.
- **Levenshtein similarity** (string-based) is ineffective for this task, confirming that surface form similarity does not align with meaning.
- **Jaccard and Dice metrics** are less effective than cosine but provide alternative perspectives.
- Static embedding models show only moderate performance, suggesting that contextual methods or hybrid approaches may further improve results.

7 Future Work

The promising avenue for future research is the integration of etymological information into similarity computation. Inspired by recent work on etymology-aware similarity measures [2], such an approach could "boost" similarity scores when both words share a common historical origin or cognate relationship, particularly useful for languages like Romanian with Latin and Slavic roots. Incorporating etymological data alongside embeddings and lexical measures may further improve the accuracy and explainability of semantic similarity predictions.

8 Conclusion

I developed and evaluated a flexible framework for measuring semantic similarity between Romanian words, leveraging multiple embedding types and metrics. My experiments show that lexical resources such as RO-WordNet are currently the best predictors of human similarity judgments in Romanian, followed by cosine similarity with static embeddings. The provided platform enables rapid experimentation and visualization, contributing a useful tool for Romanian NLP research.

References

1. Alina Maria Ciobanu and Anca Dinu. *Alternative measures of word relatedness in distributional semantics.* In Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora, pages 80–84, 2013.

2. Alina Maria Ciobanu and Liviu P. Dinu. *An Etymological Approach to Cross-Language Orthographic Similarity: Application on Romanian.* Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1047–1058, 2014.

3. Vasile Păiș and Dan Tufiș. *More Romanian Word Embeddings from the ReTeRom Project.* Research Institute for Artificial Intelligence, Romanian Academy, 2021.