

Method for Generating Interpretable Embeddings Based on Superconcepts

M. M. Tikhomirov^{1*} and N. V. Loukachevitch^{1**}

(Submitted by E. E. Tyrtysnikov)

¹*Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991 Russia*

Received March 1, 2023; revised March 28, 2023; accepted April 14, 2023

Abstract—This paper presents an approach to creating interpretable word embeddings, in which each component of the vector corresponds to some interpretable semantic category. To obtain such categories, a lexico-semantic resource is used in the form of the RuWordNet semantic network, as well as a representative corpus of Russian-language texts to train vector representations. The resulting interpretable embeddings were evaluated on semantic similarity tasks.

DOI: 10.1134/S199508022308053X

Keywords and phrases: *Thesaurus, Word embeddings, Lexical relations, Interpretability, Word semantics.*

1. INTRODUCTION

Currently, the main form of representation of words and other linguistic units are representations in the form of low-dimensional vectors (embeddings) obtained by processing large text corpora [1, 2]. At the same time, the dimensions of the vectors themselves are not interpretable – it's impossible to tell what any dimension in a vector means. Embeddings are trained by analyzing the contexts in the corpus. Recalculated embeddings on the same corpus or embeddings calculated on another corpus cannot be directly compared with each other, therefore the special procedure for constructing comparisons (transformations) between vector representations is required.

The problem of comparing different embeddings calculated on the corpus of different languages arises in such tasks as cross-lingual transfer [3, 4], model transfer between different domains (domain transfer) [5], comparison of vector representations trained on text corpora related to different time intervals (for example, to study the change in the meanings of words).

In the experiments by various authors, it has been shown that vector spaces contain the so-called hubs, which are vectors close to numerous other vectors in the space [6]. This problem manifests itself in word vector spaces in the form of words that have a high cosine similarity with many other words [7].

In this study, we consider the problems of hubs from the other side – is it possible to single out semantically interpretable hubs in a vector space, as some essential semantic categories in a collection of texts, and then transform embeddings, taking into account these hubs, thus making the dimensions of the vector space interpretable. We explore the approach of obtaining semantically interpretable categories for embeddings based on the use of lexico-semantic resources in the form of semantic networks such as WordNet [8]. In this study, we conduct experiments for the Russian language, and as hubs we use semantic classes (superconcepts) from the semantic resource of the Russian language RuWordNet [9]. The selection method of superconcepts is based on a representative text corpus, which makes it possible to create an adaptive set of semantic categories (superconcepts) for a task and/or domain. We have shown that using the proposed method it is possible to transform the initial word embeddings into interpretable ones without loss in quality on semantic similarity tasks.

*E-mail: tikhomirov.mm@gmail.com

**E-mail: louk_nat@mail.ru

2. RELATED WORKS

In theoretical and computer semantics there is the concept of a semantic class, i.e. set of words that have similar semantic features, such as living beings, objects, actions, etc. Different semantic theories and models offer their own sets of semantic classes, which are used, for example, in syntactic-semantic analysis, to build the semantic structure of a sentence, etc. When used in automatic text analysis, resources in the form of semantic networks, for example, WordNet [8], such semantic classes are formed naturally—at the expense of concepts (synsets) at the top of the semantic network hierarchy (superconcepts, supersenses). In WordNet, such a list of superconcepts includes 26 noun synsets, for example ANIMAL, PERSON, FEELING and 15 verb synsets—COMMUNICATION, MOTION, COGNITION.

In [10], the authors create embeddings for WordNet superconcepts. To do this, they use the corpus of English Wikipedia pages, automatically tagged according to the values of the BabelNet resource (which includes WordNet as an integral part) [11]—the accuracy of automatic tagging is estimated at 77.8% [12]. The corpus is used in the following way: ambiguous words receive an additional mark for the superconcept to which they refer. Based on the obtained corpus with superconcept markup, the authors train embeddings, words with ambiguity resolved concerning the superconcept, and superconcepts themselves in the same vector space word2vec [2]. As a result, embeddings trained together with superconcepts made it possible to achieve a higher quality of predicting semantic similarity of words [13–15] on four out of five datasets. In addition, superconcept embeddings have improved results in sentiment analysis, subjective sentence classification, and metaphor identification.

The work [16] explores the generation of contextualized embeddings for WordNet superconcepts based on the BiLSTM neural network model. The authors use six generalized superconcepts to represent nouns: Animated entity, Natural object, Artifact (created object), Dynamic situation (for example, action), and Static situation (for example, state). For each superconcept, the authors select 200 relevant ones, and also randomly select negative examples that are not related to this superconcept. The resulting lists of examples are used to create the pseudo-annotated corpus, in which the words from the lists are replaced with positive or negative tags of each superconcept.

A classifier is trained on the corpus, which, after receiving the word and context, predicts the probability of referring the word to a particular superconcept. The BiLSTM model is used for classification. As a result, each word receives a certain value in a 6-dimensional vector of superconcepts. Thus, for each word, an interpretable vector is obtained. The approach is evaluated in the word superconcept prediction task on the labeled French corpus and compared with the FlauBERT [17] neural network language model, the French version of the BERT model [1]. It is shown that on small amounts of training data, the proposed approach is comparable to FlauBERT. In addition, the method is much faster and less resource intensive than FlauBERT.

In another work [18], a method is proposed for training word embeddings in such a way that each coordinate is interpretable and associated with some concept from the thesaurus, and in such a way that positive and negative values of the coordinates are associated with different concepts. The authors achieve their goal by fixing top K concepts from the thesaurus and modifying the loss function to train embeddings. One of the advantages of the proposed approach is that there was no quality degradation on the evaluated benchmarks for the resulting embeddings.

As can be seen, in described works, superconcepts (abstract semantic categories) are fixed in advance. In our work, we explore an approach to the flexible (adaptive) selection of superconcepts for a task or domain based on the corresponding text corpus and the structure of the corresponding embeddings.

3. ADAPTIVE SUPERCONCEPTS

3.1. *RuWordNet Thesaurus*

The RuWordNet thesaurus is a lexical-semantic resource (semantic network) for the Russian language. Nodes of the RuWordNet are sets of synonyms (synsets) connected with semantic relations (hyponym—hypernym, part—whole and others). Currently, the published version of RuWordNet includes 110,000 Russian words and expressions [19].

The current version of RuWordNet includes the following types of relationships: hyponym—hypernym, antonyms, domain relationships for all parts of speech (nouns, verbs, and adjectives); part—whole relationships for nouns; and cause and entailment relationships for verbs. Synsets of different parts of speech are connected with POS-synonym relationships. Single words with the same roots are connected with derivational relationships. Relationships to component synsets are given for phrases included in RuWordNet.

3.2. Key Ideas of the Adaptive Superconcept Method

Superconcepts are synsets of RuWordNet (or other resources in the form of semantic networks) that have the following properties:

- Superconcept must have several hyponyms, i.e., actually be a generalizing concept for more specific concepts, represent a knot of similar words in the thesaurus,
- Superconcept as a knot of words that are close in meaning must also be confirmed based on some representative corpus. Words from superconcepts and related concepts must form a clique – a graph in which all vertices are interconnected. Links between words in a clique are calculated based on the cosine similarity of embeddings that exceeds a given threshold. For example, a “city” synset associated with numerous cities in RuWordNet corresponds to a clique of more than 200 city names whose vector representations are close to each other.

Thus, superconcepts are semantic categories identified by people in semantic resources, additionally confirmed by the high semantic similarity of embeddings associated with them, which makes it possible to adapt a set of superconcepts to a specific task or a specific domain. It is assumed that for different tasks and related data corpora, the set of superconcepts should differ, reflecting the most significant categories in a given domain.

4. SUPERCONCEPT SELECTION METHOD

The superconcept selection algorithm includes the following steps:

- Selection synsets from RuWordNet that have at least several species relations (hyponyms) with other synsets—these are candidates for superconcepts,
- The generation of a set of words (semantic hub) corresponding to the candidate superconcept—these are all words and expressions from the concept’s synset in RuWordNet, as well as words and expressions from synsets directly related to this concept by hyponym—hypernym relations (genus species relations),
- Creating a clique of concepts for a superconcept based on the cosine similarity of the embeddings of the hub synsets and a given threshold. Cliques are calculated using an algorithm from the NetworkX package. The link weight between two synsets is defined as the average pairwise similarity of words. The weights of word links are the value of cosine similarity calculated from some corpus by the distributive model (word2vec, glove). The average weight must exceed the specified threshold. In order to be selected as a candidate for a superconcept, a minimum clique must arise—3 concepts,
- For the selected superconcept, it is necessary to create a representative vector representation, since the semantic hub associated with the superconcept (synonyms, hyponyms, hypernyms) may contain polysemantic and/or rare words. To do this, a network of distributive relations with weights of cosine similarity is created from the words of the hub. The PageRank [20] of each word is calculated on the network, the five most weighty words are selected, and their averaged vector representation is considered as an embedding of the superconcept. Examples of selected superconcepts based on the vector model are presented in Table 1,

Table 1. Selected superconcepts example

Synset ID	Synset Name	Top-5 words
106509-N	Component	piece, stripe, bottom, ribbon, side
106501-N	Occupation, activity	fun, occupation, amusement, hobby, industry
106768-N	Property, characteristic	abstractness, artlessness, style, abstraction, isolation
106646-N	Relationship between entities	difference, correlation, relationship, dissimilarity, diversity
106505-N	Product of labor	original, production, creation, product, publication

- The last step is the selection of superconcepts from all candidates. The selection starts from the top level of the hierarchy of the RuWordNet, since we are interested in the most general, abstract concepts. If the next superconcept is similar by vector model to the one already selected and is its hyponym, then this concept is rejected, since too many similar superconcepts lead to an intricate system of semantic categories.

Despite the selection method, superconcepts that are still close in meaning are distinguished, which usually represent hyponyms of a broader concept (sister concept), for example, the concepts of the “humanitarian sciences” and “social sciences”. In this case, it is natural to generalize sister concepts to a broader concept. Here are examples of such generalizations:

- humanitarian sciences, social science -> science,
- scientific institution, institution of higher education -> research and development organization,
- foreign state, European country, developed state, republic (state), socialist state -> state,
- flowering plant, herbaceous plant, shrub, evergreen -> plant,
- military rank, title of nobility -> rank.

Thus, as a result of the described procedure, superconcepts (semantic categories) were selected from the RuWordNet, which simultaneously correspond to a certain set of words that are close to each other in terms of similarity (clique). Each superconcept has its own embedding according to the original vector model. All other words from the text corpus can be represented as a vector of similarity to these superconcepts, i.e., if 100 superconcepts are selected, then for each word a vector of dimensions of 100 interpretable dimensions will be obtained.

5. EXPERIMENTS

5.1. Datasets

The evaluation of the proposed method was carried out on the Russian language on semantic similarity datasets: WordSim-353 in two variants (two different translations) [14, 15] and SimLex-999 [21]. The metric was the Spearman correlation between scoring based on the vector model and scoring based on human ratings. It is worth noting that the quality of the RUSSE wordsim dataset [15] is higher than the other one [14] for Russian, since all translations were separately evaluated by people.

5.2. Baselines

As baselines, we considered the following approaches:

- **init_model**—Initial vector model.
- **random_words**—Random selection of K words from the vocabulary as anchors and the creation of a vector as similarity to them,
- **random_words_t5**—Random selection of K words from the vocabulary as anchors and top-5 most similar words to them and the creation of a vector as similarity to the mean vector,
- **kmeans_words**—Clustering words with k-means algorithm and choosing the words closest to the centers of the clusters as anchors,
- **kmeans_words_t5**—Similar to **kmeans_words**, but choosing top-5 words for each cluster.
- **kmeans_synsets**—Clustering synsets with k-means algorithm and choosing synsets closet to the centers of the clusters as anchors, synsets embeddings are averaged embeddings of synset words,
- **kmeans_hubs**—Clustering of superconcepts obtained at the first step of the proposed algorithm without further filtering, top 5 representative words are used as vector representations of superconcepts,
- **kmeans_synsets (taxoexpan)**—Similar to **kmeans_synsets**, but synset embeddings are formed using TaxoExpan,
- **kmeans_hubs (taxoexpan)**—Similar to **kmeans_synsets**, but superconcept embeddigs are formed using TaxoExpan (vector of supersence center),

As a k-means algorithm, 2 implementations from the scikit-learn Python package were tested: KMeans and MiniBatchKMeans. A comparison was made of the quality, std and performance.

5.3. TaxoExpan

TaxoExpan [22] is an approach in which graph vector representations (GCN [23], GAT [24]) for thesaurus concepts are trained by solving the hypernymy prediction task. The method proposed by the authors shows high quality on a number of datasets, so it was used as an alternative approach to encoding concepts into vector space.

Some previous work [25] shows, that there are problems with applying TaxoExpan implementation, provided by the authors, to RuWordNet. We investigated this problem and were able to get positive results for the investigated vector model.

- Problem 1: When using a vector model other than fasttext, not all concepts are covered by the model, which leads to abnormal values of predictions on a number of concepts,
- Problem 2: Some concepts in the thesaurus have hundreds of hyponyms, so in the case of automatic train/validation/test splitting, some concepts appear as the correct answer too many times in relation to others, which can negatively affect the learning process,
- Problem 3: In the case of prediction for new words by an existing method in infer.py, predictions are made only for a part of the nodes (without validation nodes), and test nodes are disconnected from the general graph,
- Problem 4: The normalization of vectors during training and inferring is different.

To eliminate Problem 1, the vector representation of phrases was formed by averaging the vectors of constituent words, and the remaining concepts not covered by the model were removed from the thesaurus in such a way that their hyponyms and hypernyms were connected by the corresponding relation. Problem 2 was solved by creating a partition in a special way so that the same hypernym could be present as a positive example during training no more than K times. Problems 3-4 have also been fixed.

The model was trained with LMB classifier and PGAT embeddings, with initial vectors normalization and expand factor was reduced to 10. The quality of the model was tested on the RUSSE'2020 nouns restricted dataset [25], and reached 0.30 MRR.

5.4. Evaluation

The proposed method and baselines were evaluated based on a vector model trained on a text corpus of 8 million news texts. The vector model was trained using the PPMI SVD method, since preliminary experiments on part of the data showed that this approach performs better than word2vec for this task. Embeddings were trained with **window=3** and **min word count=50**.

In this experiment, the following parameters of the superconcept selection algorithm were used (most of parameters were selected using grid search and based on wordsim):

- The minimum number of relations for a concept to form a superconcept ($=5$),
- The minimum similarity between concepts to form an edge in the process of building a superconcept clique ($=0.4$),
- The minimum similarity between words to form an edge between words in a superconcept to build a graph and then calculate pagerank on it ($=0.4$),
- The minimum size of a clique in a superconcept ($=3$),
- The minimum size of a superconcept in words ($=5$),
- Number of words to calculate superconcept vector based on PageRank ($=5$),
- The minimum similarity between the superconcept and its already added hypernyms at which the superconcept will be added to the final list ($=0.3$).

Parameters of the generalization algorithm:

- The minimum similarity between a pair of co-hyponyms to generalize to a hypernym ($=0.4$),
- The minimum depth of a superconcept that can serve as a generalization ($=3$).

Table 2 shows the results on described datasets. The results that are no worse than the initial model are underlined, and the best values are bold. The results for all models, except for the original and the proposed method, were obtained by averaging 5 runs.

The table shows that the proposed method is not inferior in quality to the `init_model` and has similar results with `model kmeans_hubs`, but is more deterministic, since it lacks an element of randomness. In addition, all the best results were achieved using a primary list of superconcepts in which representative words were formed in the manner described in the work. Should also be noted that the methods of concept vectorization using TaxoExpan performed worse in that case.

Table 2. Selected superconcepts example

method	simlex	wordsim overall	RUSSE wordsim overall
init_model	0.291	0.661	0.649
random_words	0.284	0.613	0.611
random_words_t5	0.288	0.615	0.623
kmeans_words	0.287	0.651	<u>0.649</u>
kmeans_words_t5	0.283	0.653	<u>0.655</u>
kmeans_synsets	0.271	0.648	0.630
kmeans_hubs	<u>0.298</u>	0.649	0.682
kmeans_synsets (taxoexpan)	0.278	0.632	0.627
kmeans_hubs (taxoexpan)	<u>0.297</u>	0.631	<u>0.657</u>
hub_vectors	0.303	0.643	<u>0.671</u>

Table 3. KMeans and MiniBatchKMeans

method	RUSSE wordsim overall	time (5 runs)
random_words_t5	0.615+–0.012	0s
kmeans_words_t5	0.655+–0.004	580s
mbkmeans_words_t5_64	0.642+–0.014	6s
mbkmeans_words_t5_128	0.634+–0.018	6s
mbkmeans_words_t5_512	0.612+–0.009	6s
mbkmeans_words_t5_1024	0.616+–0.011	9s
mbkmeans_words_t5_4096	0.612+–0.010	40s

5.5. Comparing KMeans and MiniBatchKMeans

We also compared the quality and performance of the kmeans_words_t5 baseline in terms of using the implementation of the k-means algorithm from scikit-learn framework¹⁾. Evaluated using an AMD Ryzen 7 5800X processor with 8 cores. We evaluated the performance of the algorithm with different batch sizes (64, 128, 512, 1024, 4096), the remaining parameters were equivalent to the regular version of the method.

As can be seen, the use of a faster implementation in this case (30 thousand vectors, 400 clusters, 10000 max iterations) significantly speeds up the algorithm, but the result itself is indistinguishable from a random selection with large batch sizes and lower than non-batched implementation in case of 64–128 batch size. It is also important to note the significantly higher standard deviation.

6. CONCLUSIONS

This paper presents an approach to creating interpretable vector representations of words, in which each component of the vector corresponds to some interpretable semantic category. To obtain such categories, a lexico-semantic resource is used in the form of a semantic network of the Russian language RuWordNet, as well as a representative corpus of Russian-language texts to train embeddings. The method allows the selection of candidates for superconcepts taking into account the trained embeddings, thus giving more attention to the domain of interest.

¹⁾<https://scikit-learn.org/>

The approach was evaluated on semantic similarity tasks. The results showed that the proposed method is able to transform the initial word embeddings into interpretable ones without significant loss of quality. Each coordinate of the resulting vector representations is interpretable, as it is associated with a specific semantic category from the thesaurus.

Also, our method is applicable to already existing embeddings and does not require retraining of the model.

FUNDING

The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University. The study was funded by a grant Russian Science Foundation (project no. 21-71-30003). The work of Mikhail Tikhomirov was supported by Non-commercial Foundation for Support of Science and Education “INTELLECT.”

REFERENCES

1. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019), Vol. 1, pp. 4171–4186.
2. T. Mikolov et al., “Distributed representations of words and phrases and their compositionality,” arXiv: 1310.4546 (2013).
3. M. Artetxe, G. Labaka, and E. Agirre, “Learning principled bilingual mappings of word embeddings while preserving monolingual invariance,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016), pp. 2289–2294.
4. T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” arXiv: 1309.4168 (2013).
5. J. Yamane et al., “Distributional hypernym generation by jointly learning clusters and projections,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (2016), pp. 1871–1879.
6. M. Radovanovic, A. Nanopoulos, and M. Ivanovic, “Hubs in space: Popular nearest neighbors in high-dimensional data,” *J. Mach. Learn. Res.* **11**, 2487–2531 (2010).
7. S. Ruder, I. Vulić, and A. Søgaard, “A survey of cross-lingual word embedding models,” *J. Artif. Intell. Res.* **65**, 569–631 (2019).
8. G. A. Miller, *WordNet: An Electronic Lexical Database* (MIT, Boston, 1998).
9. N. V. Loukachevitch et al., “Creating Russian wordnet by conversion,” in *Computational Linguistics and Intellectual Technologies: Proceedings of the Annual Conference Dialogue* (2016), pp. 405–415.
10. L. Flekova and I. Gurevych, “Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1: *Long Papers* (2016), pp. 2029–2041.
11. R. Navigli and S. P. Ponzetto, “BabelNet: Building a very large multilingual semantic network,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (2010), pp. 216–225.
12. F. Scozzafava et al., “Automatic identification and disambiguation of concepts and named entities in the multilingual wikipedia,” in *AI* IA 2015 Advances in Artificial Intelligence: Proceedings of the 14th International Conference of the Italian Association for Artificial Intelligence, Ferrara, Italy, September 23–25, 2015* (Springer Int., Switzerland, 2015), pp. 357–366.
13. E. Agirre et al., “A study on similarity and relatedness using distributional and wordnet-based approaches,” in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Anthology N09-1003 (2009).
14. I. Leviant and R. Reichart, “Separated by an un-common language: Towards judgment language informed vector space modeling,” arXiv: 1508.00106 (2015).
15. A. Panchenko et al., “Human and machine judgements for Russian semantic relatedness,” in *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts* (Springer, Cham, 2016), pp. 221–235.
16. C. Aloui et al., “Slice: Supersense-based lightweight interpretable contextual embeddings,” in *Proceedings of the 28th International Conference on Computational Linguistics COLING 2020* (2020).
17. H. Le et al., “Flaubert: Unsupervised language model pre-training for french,” arXiv: 1912.05372 (2019).
18. L. K. Şenel et al., “Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts,” *Inform. Process. Manage.* **59**, 102925 (2022).

19. N. Loukachevitch, G. Lashevich, and B. Dobrov, “Comparing two thesaurus representations for Russian,” in *Proceedings of the 9th Global WordNet Conference GWC 2018* (2018), pp. 35–44.
20. S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Comput. Networks ISDN Syst.* **30**, 107–117 (1998).
21. F. Hill, R. Reichart, and A. Korhonen, “Simlex-999: Evaluating semantic models with (genuine) similarity estimation,” *Comput. Linguist.* **41**, 665–695 (2015).
22. J. Shen et al., “TaxoExpan: Self-supervised taxonomy expansion with position-enhanced graph neural network,” in *Proceedings of The Web Conference 2020* (2020), pp. 486–497.
23. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv: 1609.02907* (2016).
24. P. Veličković et al., “Graph attention networks,” *arXiv: 1710.10903* (2017).
25. I. Nikishina et al., “Taxonomy enrichment with text and graph vector representations,” *Semantic Web* **13**, 441–475 (2022).