

Generation of Interpreted Vector Representations of Words Based on Supersenses

M. M. Tikhomirov^{a,*} and N. V. Loukachevitch^{a,**}

^a *Lomonosov Moscow State University, Moscow, 119991 Russian Federation*

^{*}*e-mail: tikhomirov.mm@gmail.com*

^{**}*e-mail: louk_nat@mail.ru*

Abstract—This work presents an approach to creating interpreted vector representations of words in which each component of the vector corresponds to a certain interpreted semantic category. To obtain such categories, a lexical and semantic resource in the form of the semantic network RuWordNet is used, as well as a representative corpus of Russian-language texts for generating vector representations. The resulting interpreted vector representations were tested for the ability to display different models in the same vector space.

Keywords: vector representations of words, semantic categories, thesaurus, semantic similarity

DOI: 10.1134/S1054661823030446

INTRODUCTION

Currently, the main form of representation of words and other language units are representations in the form of vectors of low dimension (embeddings) obtained by processing large text corpora [6, 15]. In this case, the dimensions of vectors themselves are not interpretable—it is impossible to say what this or that dimension in the vector means. Vector representations are formed when analyzing the contexts in the corpus, while in the process of learning there is an element of randomness, primarily when initializing the model. Recalculation of vector representations on the same corpus and vector representations calculated on another corpus cannot be compared with each other; they require special procedures for constructing comparisons (transformations) between vector representations.

The problem of comparing different vector representations calculated on different corpora of the same language or different languages arises in such problems as cross-language model transfers [4, 14], model transfers between different domains (domain transfer) [21], and comparison of vector representations trained on text corpora related to different time intervals (for example, for study of changes in word values).

In experiments by different authors, it was shown that vector spaces contain so-called hubs, which are vectors close to a large number of other vectors in space [18]. This problem manifests itself in vector spaces of words in the form of words that have a high cosine resemblance to a large number of other words [19].

In this study, we look at the problems of hubs, on the other hand—whether it is possible to distinguish semantically interpreted hubs in vector space as some essential semantic categories in a collection of texts, and further expand the vector space taking into account these hubs, thus making measurements of vector space interpretable.

We explore the approach of obtaining semantically interpreted categories for vector representations based on the attraction of lexical and semantic resources in the form of semantic networks such as WordNet [7]. In this study, we conduct experiments for the Russian language, and as such hubs we use semantic classes (supersenses) from the semantic resource of the Russian language RuWordNet [13]. The selection of supersenses is made on the basis of a representative text corpus, which allows one to create an adaptive set of semantic categories (supersenses) for a task and/or subject area.

1. CLOSE WORKS

In theoretical and computer semantics, there is the concept of a semantic class, that is, a set of words that have similar semantic features, for example, living beings, objects, actions, etc. Different semantic theories and models offer their own sets of semantic classes [3, 9], which are used, for example, in syntactic-semantic analysis, to construct the semantic structure of a sentence, etc.

When using resources in the form of semantic networks in automatic text analysis, for example, WordNet [7], such semantic classes are formed naturally—by means of concepts (synsets) at the top of the hierarchy of the semantic network (supersenses). In WordNet, this list of supersenses includes 26 syn-

sets of nouns, for example, ANIMAL, PERSON, and FEELING, and 15 synsets of verbs, for example, COMMUNICATION, MOTION, and COGNITION.

In [8], the authors create vector representations for WordNet supersenses. To do this, they use the corpus of English Wikipedia pages, automatically marked by the values of the BabelNet resource (which includes as an integral part WordNet) [16]—the accuracy of automatic markup is estimated as 77.8% [20]. The marked corpus is used as follows: ambiguous words receive an additional mark of the supersense to which they relate. On the basis of the received corpus with the markup of supersenses, the authors teach vector representations of words, words with a resolved ambiguity regarding the supersense, and the supersenses themselves in one vector space word2vec [15]. As a result, embeddings for words trained in conjunction with supersenses allowed for higher quality predictions of semantic similarity of words [1, 12, 17] on four datasets out of five. In addition, embeddings with supersenses have improved the results in the sentiment analysis, classification of subjective sentences, and the identification of metaphors.

The paper [2] investigates the generation of contextualized embeddings for WordNet supersenses based on the neural network model of BiLSTM. The authors use six generalized supersenses to represent nouns: Animate Entity, Natural Object, Artifact (created Object), Dynamic situation (e.g., Action), Static situation (e.g., state). For each supersense, the authors select 200 relevant and also randomly additionally selected negative examples that are not related to this supersense. The resulting lists of examples are used to mark the corpus, in which words from the lists are replaced with positive or negative tags of each supersense, and a pseudo-annotated case is obtained.

On the received corpus, a classifier is trained, which, receiving the word and context, predicts the probability of referring the word to a specific supersense. The BiLSTM neural model is used for classification. As a result, each word receives a certain value in the vector of dimension 6 of supersenses. Thus, for each word, an interpreted vector is obtained. The approach is tested in the problem of predicting a supersense for a word on a marked French corpus and compared with the neural network language model FlauBERT [11], the French version of the model BERT [6]. It is shown that, on small volumes of training data, the proposed approach is comparable with FlauBERT. In addition, the method is much faster and less expensive on resources than FlauBERT.

As can be seen, in the described works, supersenses (abstract semantic categories) are fixed in advance. In our work, we explore the approach to flexible (adaptive) allocation of supersenses for a task or subject area based on the corresponding text corpus and the structure of the corresponding vector representations.

2. THE IDEA OF ADAPTIVE SUPERSENSES

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

- the supersense should have several hyponyms (specific concepts), i.e., really be a generalizing concept for more specific concepts and be a node of close words on the thesaurus;
- the concept as a node of words close in meaning must be confirmed on the basis of some representative corpus—namely, words from supersenses and related concepts must form a clique—a graph in which all vertices are connected with each other. The connections between words in the clique are considered on the basis of a cosine measure of the similarity of embeddings for a given corpus exceeding a given threshold. For example, a synset *city* linked in RuWordNet to a large number of specific cities corresponds to a clique of more than 200 city names whose vector representations are close to each other with a cosine measure above 0.4 calculated on the basis of the Araneum corpus.

Thus, supersenses are semantic categories allocated by people in semantic resources, additionally confirmed by the high semantic similarity of vectors, associated with other words, which makes it possible to adapt a set of supersenses to a specific task or a specific subject area; i.e., it is assumed that, for different tasks and related corpora of data, the set of supersenses should differ, reflecting the most significant categories in this subject area.

3. THE ALGORITHM OF GENERATION OF SUPERSENSES

The algorithm for the formation of supersenses includes the following steps.

1. The selection of synsets from RuWordNet that have at least several specific relations (hyponyms) with other synsets are candidates for supersenses.
2. The generation of a set of words (semantic hub) corresponding to the supersense candidate is all the words and expressions from the synset of the concept in RuWordNet, as well as words and expressions from the synsets directly related to this concept by the relations of hyponym-hyperonym (generic-specific relations).
3. Creating a concept clique for a supersense based on the cosine similarity of hub synset embeddings 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 link weight of words from different synsets of the pair. The weight of the word's links is the value of the cosine similarity calculated by some corpus of the distribution model (word2vec, glove). The average weight must be greater than the set threshold. In order to be

selected as a candidate for a supersense, a minimum clique must arise—three concepts.

4. For the selected supersense, it is necessary to create a representative vector representation, since multivalued and/or rare words can occur in the semantic hub associated with the supersense (synonyms, hyponyms, hyperonyms). To do this, from the words of the hub, a network of distributive relations with the scales of cosine similarity is created, the PageRank [5] of each word is calculated on the network, the five most weighty words are selected, and their average vector representation is considered as a vector representation of a supersense. Examples of selected supersenses based on the vector model of the Araneum corpus [10] are presented in Table 1.

5. The next stage is the selection of supersenses from all selected candidates. Selection begins at the top level of the hierarchy of the semantic RuWordNet network, since we are interested in the most generalized, abstract concepts. If another supersense is similar in a vector model to one already selected and being its hyperonym, then this concept is rejected, because too many similar supersenses lead to an intricate system of semantic categories.

A simplified fragment of the supersense selection log is as follows:

```
{'106509-N', 'name': 'component', 'status': 'Added'}
{'106646-N', 'name': 'relation between entities', 'status': 'Added'}
{'106846-N', 'name': 'variant, variety', 'status': 'Added'}
{'132964-N', 'name': 'innovation', 'status': 'skipped', 'condition': 'clique_size < 3'}
...
{'112766-N', 'name': 'back', 'status': 'skipped', 'condition': 'close to (106509-N:0.59)'}
{'110735-N', 'name': 'particle (small part)', 'status': 'skipped', 'condition': 'close to (106509-N:0.50)'}
```

The log shows that some of the supersenses are not approved (skipped) because of the lack of a corresponding clique (“clique_size < 3”), and some of the supersenses are rejected because of too high a similarity to the already selected supersenses (“close to”).

Despite the selection of supersenses, there are still supersenses close in meaning, which are usually subtypes of a broader concept (sister concepts), for example, the concepts of *the humanities* and *social sciences*. In this case, it is natural to generalize sister concepts to a broader concept. Let us give examples of generalizations of sister concepts similar by the vector model to the generic concept:

- *humanities, social sciences* → *science*;
- *scientific institution, higher educational institution* → *organization that performs research and development*;
- *a foreign state, a European country, a developed state, a republic (state), a socialist state* → *a state*;

Table 1. Examples of highlighted supersenses and their most representative words

RuWordNet synset ID	Name of the RuWordNet synset	Five most representative words corresponding to the supersense
106509-N	Component part	<i>slice, strip, bottom, strip, side</i>
106501-N	Occupation, activity	<i>fun, occupation, entertainment, passion, industry</i>
106768-N	Property, characteristic	<i>abstract, unsophisticated, stylish, abstract, isolated</i>
106646-N	Relationship between entities	<i>difference, correlation, relationship, dissimilarity, divergence</i>
106505-N	Product of work	<i>original, product, production, goods, issue</i>

• *flowering plant, herbaceous plant, shrub, evergreen plant* → *plant*;

• *military rank, noble title* → *rank*.

Depending on the corpus on which the vector model is calculated, the described procedure gives 100–300 supersenses that combine different numbers of elements. Supersenses (semantic classes) should not be many. Therefore, at the last step, supersenses are replenished with all the other words from the corpus (at the current stage, only nouns from RuWordNet present in the corpus), whose cosine similarity to supersenses is above a certain threshold, and a given number of supersenses with the maximum number of elements is selected.

Thus, as a result of the described procedure, supersenses (semantic categories) from the RuWordNet resource are selected, to which a certain set of words close to each other according to the corpus distributive model (clique) simultaneously corresponds. Each supersense has its own vector representation according to the original distribution model. All other words from the text corpus can be represented as a vector of similarity to these supersenses; i.e., if 100 supersenses are selected, then a vector of dimension of 100 interpreted dimensions will be obtained for each word.

4. EXPERIMENTS

In the experiment on the described algorithm, the thesaurus RuWordNet and vector models trained on a corpus of 800 000 news articles were used. On the basis of the similarity of words to the vectors of supersenses, each word from the vector model was compared to a vector from K components according to the vector nearness of the word to supersenses.

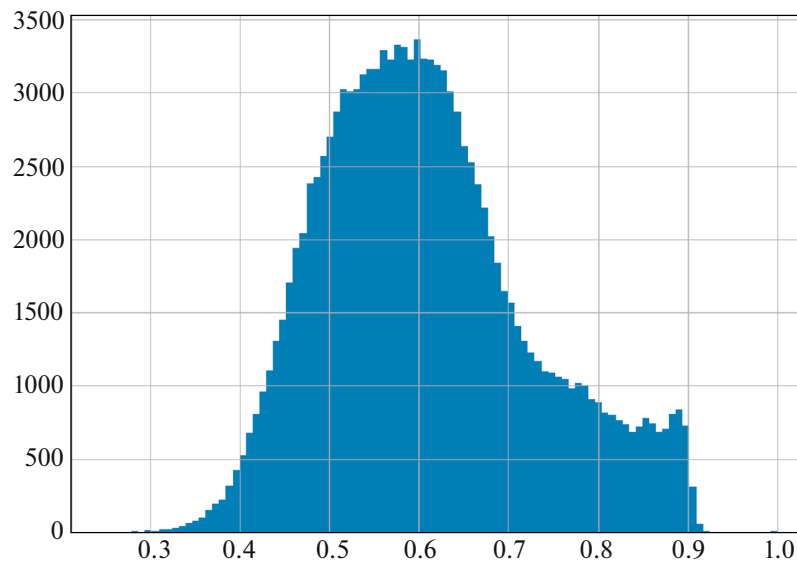


Fig. 1. Histogram of the similarity of words to themselves in different models. On the x axis is the measure of closeness; on the y axis is the number of words.

The following parameters were used in this experiment:

1. Parameters of the algorithm for constructing the list of supersenses:

(a) the minimum number of generic-specific relations in a concept for the formation of a supersense (=5),

(b) the minimum value of the similarity between concepts for the formation of an edge in the process of building a clique of a supersense (=0.4),

(c) the minimum value of the similarity between words to form the edge between words in a supersense for constructing a graph and then calculating the PageRank on it (=0.4),

(d) the minimum clique size from concepts in the supersense (=3),

(e) the minimum size of a supersense in words (=10),

(f) the number of words for calculating the supersense vector based on PageRank (=5),

(g) the minimum value of similarity between the supersense being added and its already added hyperonyms at which the supersense will be added to the final list (=0.3).

2. Parameters of the algorithm for generalization:

(a) the minimum value of similarity between a pair of co-hyponyms for generalization on a hyperonym (=0.4),

(b) the minimum depth of a supersense, which can serve as a generalization (=3),

(c) the number of words for calculating the supersense vector (=5).

We give examples of the obtained supersenses of RuWordNet on these parameters: *component part*; *product of labor*; *occupation, activity*; *to be in a state*; *image (result)*; *group, combined on a common basis*; *subject of activity*; *substance*; *scientist, etc.* (see Appendix). As one can see, it was possible to identify generalizing semantic categories. It can also be noted that narrower categories are also distinguished; for example, among the concept of the *subject of activity*, the category of *researcher is separately distinguished*, which has specific contexts. This is the supposed adaptability of the method of allocation of supersenses, which can distinguish significant, important, specific categories not necessarily of the highest level of the semantic network.

As an experiment, the ability to display different models in one interpreted vector space was tested. To do this, on the basis of the aforementioned news corpus, two word2vec vector models with the same parameters were trained (skip-gram, window size of 3, number of epochs of 10).

The first part of the experiment was to verify that two vector models trained even on the same corpus and with the same parameters would have different vector representations. For this purpose, for each word from the collection, the cosine similarity between vectors from two different models was considered. A histogram of word similarity is shown in Fig. 1. The average similarity was 0.6. This means that ordinary vector models, even trained on the same corpus, create incomparable vector representations in which the representations of even the same word may not be similar to each other.

To test the persistence of semantic similarity based on supersense vectors, we investigated how such a rep-

resentation can reveal similarities between semantically close words from independently trained vector models. To do this, a list of ~100 pairs of words was previously compiled, on the basis of the previously trained word2vec model in the same formulation, which had high similarities. To this end, for each word from the collection, a list of the closest words was obtained; these lists were ordered by the highest similarity among the received close words. For the first 100 words in the list, their closest words were selected.

Using the resulting list, in the training of vector models, in the first case, all the “right” words of pairs were removed from the corpus; in the second case, all the “left” words of pairs were removed. Thus, the final dictionaries of vector models differed according to the obtained list of pairs. For example, for the pair (**belarus**, **belorussia**), when training the first vector model, all cases of using the word **belorussia** were removed, and when training the second model, all cases of using **belarus** were removed. Thus, these models do not have the corresponding words in their dictionaries.

Further, for each of the two vector models, in accordance with the construction algorithm, vector representations based on supersenses were obtained. In order for the obtained vector models to be fully comparable, alignment was performed at several stages of the algorithm.

1. The maximum clique weight for each supersense was obtained by averaging the corresponding weights in two models.
2. The final list of supersenses was formed as an intersection of the lists of supersenses obtained in the

two models, and the position of supersenses was formed as an average position.

Thus, the obtained lists of supersenses turned out to be identical and contained 78 supersenses. Since the total number of supersenses due to the procedure of crossing the lists turned out to be less than 100, the operation to truncate the set was not carried out. In a simpler case, when work is done with only one model, usually the number of supersenses is more than 100.

As a result, each word from the model was mapped to a vector model based on supersenses, and if the word was found in both models, its representation was averaged. In the obtained vector model, the dictionary is a combination of dictionaries of two models.

To test the hypothesis that, with the help of the developed approach, it is possible to display different models in one vector space, an analysis was conducted of how close to each other are the words from previously selected pairs in relation to other words. To do this, in the obtained vector model for each word from the pair, a list of the closest words was calculated, among which a search for the second word from the pair and its position in the list was conducted. For example, the five closest words to the word **belarus** were **belorussia** 0.991, **kazakhstan** 0.988, **sudan** 0.984, **ukraine** 0.984, and **russia** 0.984; the five closest words to **belorussia** were **belarus** 0.991, **kazakhstan** 0.984, **latvia** 0.983, **hungary** 0.983, and **moldova** 0.983. Thus, the average rank for the pair (**belarus**, **belorussia**) is 0 (counting from 0).

The results showed that for 60% of pairs, the desired word was in the top 5. The final allocation of positions can be seen in Fig. 2.

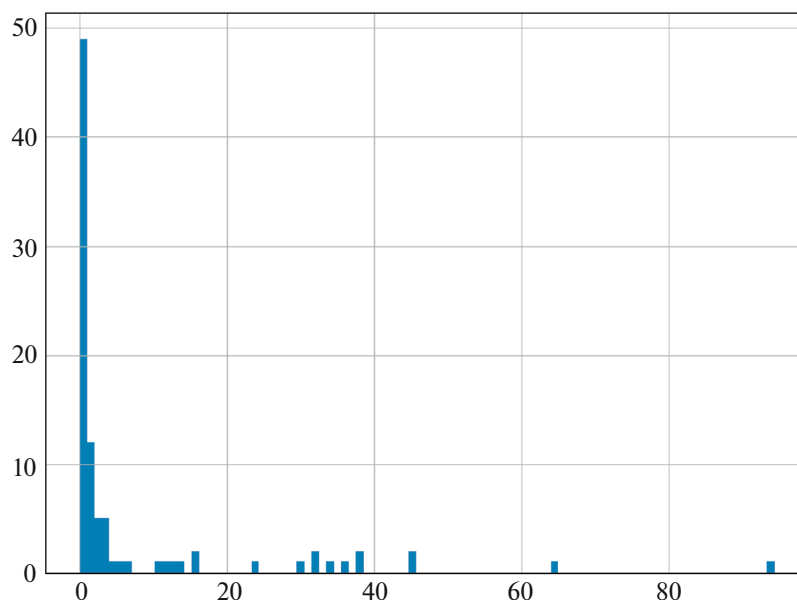


Fig. 2. Rank distribution histogram for word pairs under study. On the x axis is the rank; on the y axis is the number of pairs.

Thus, two models were independently trained on the corpus each of which did not contain some set of words close to each other, but in the representation based on supersenses, these words were close.

CONCLUSIONS

This paper presents an approach to creating interpreted vector representations of words in which each component of the vector corresponds to some interpreted semantic category. To obtain such categories, the lexical and semantic resource in the form of the semantic network of the Russian language RuWordNet is used, as well as a representative corpus of Russian-language texts for generating vector representations.

The resulting interpreted vector representations were tested for the possibility of displaying different models in one vector space so that words close in meaning that are present in one model, but are not in another, are close to each other in the final vector model.

APPENDIX

The appendix provides a list of supersenses for the model (the first 30 supersenses) which were obtained in the described experiment:

Supersense 0: component part: strip, piece, pulp, barrel, component

Supersense 1: product of work: work, production, goods, product, ware

Supersense 2: occupation, activity: occupation, industry, hobby, profession, interest

Supersense 3: to be in a state: languishing, vitality, adoration, thawing, starvation

Supersense 4: image (result): photo, drawing, image, engraving, snapshot

Supersense 5: group united by a common trait: three, thirty, ten, hundred, twenty

Supersense 6: subject of activity: debtor, opponent, buyer, benefactor, organizer

Supersense 7: substance: perfume, preservative, powder, ingredient, substance

Supersense 8: vary, change: improvement, increase, decrease, slow down, impregnation

Supersense 9: place in space: bend, bulge, speck, surface, ravine

Supersense 10: natural phenomenon: weather, sediment, cloud, wind, cyclone

Supersense 11: spend, consume: expenditure, economy, consumption, overconsumption, cost

Supersense 12: computer program: utility, application, subprogram, program, update

Supersense 13: biological essence: cell, tissue, mitochondria, organism, chromosome

Supersense 14: movement, displacement: riding, movement, inclination, travel, jump

Supersense 15: state, internal circumstances: orderliness, dryness, moisture, sputum, situation

Supersense 16: population: urban, Latino, European, Uralian, Sverdlovsk

Supersense 17: physiological process: climax, regeneration, secretion, heartbeat, menopause

Supersense 18: physical property: permeability, density, conductivity, fragility, property

Supersense 19: unit of measurement of information: Mb, gigabyte, Mbit, kbit, megabyte

Supersense 20: unit of volume: gallon, decaliter, half liter, barrel, liter

Supersense 21: change, make different: improve, lighten, increase, decrease, change

Supersense 22: unit of mass: kilogram, gram, kilo, ton, ounce

Supersense 23: ability: tact, musicality, skill, flexibility, disposition

Supersense 24: material for manufacture: foam, latex, plastic, winding, rubber

Supersense 25: unit of length: micron, meter, centimeter, nanometer, millimeter

Supersense 26: construction, structure: construction, building, structure, hut, formation

Supersense 27: object, thing: ball, bar, piece, patch, decoration

Supersense 28: age: adolescence, childhood, age, property, old age

Supersense 29: shape, appearance: roundness, bulge, trim, cut, style

Supersense 30: God: God, Lord, Christ, god-man, Jesus

FUNDING

This work was supported by a grant from the Russian Science Foundation (project no. 21-71-30003).

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and WordNet-based approaches," in *Proc. Human Language Technologies: The 2009 Annu. Conf. of the North Am. Chapter of the Assoc. for Computational Linguistics NAACL'09, Boulder, Colo., 2009* (Association for Computational Linguistics, Stroudsburg, Pa., 2009), pp. 19–27.
<https://doi.org/10.3115/1620754.1620758>

2. C. Aloui, C. Ramisch, A. Nasr, and L. Barque, "SLICE: Supersense-based lightweight interpretable contextual embeddings," in *Proc. 28th Int. Conf. on Computational Linguistics, Barcelona, 2020* (International Committee on Computational Linguistics, 2020), pp. 3357–3370.
<https://doi.org/10.18653/v1/2020.coling-main.298>
3. Yu. D. Apresyan, *Lexical Semantics: Synonymical Means of Language* (Moscow, 1974).
4. M. Artetxe, G. Labaka, and E. Agirre, "Learning principled bilingual mappings of word embeddings while preserving monolingual invariance," in *Proc. 2016 Conf. on Empirical Methods in Natural Language Processing, Austin, Texas, 2016* (Association for Computational Linguistics, 2016), pp. 2289–2294.
<https://doi.org/10.18653/v1/d16-1250>
5. S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Networks ISDN Syst.* **30**, 107–117 (1998).
[https://doi.org/10.1016/s0169-7552\(98\)00110-x](https://doi.org/10.1016/s0169-7552(98)00110-x)
6. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minn., 2019* (Association for Computational Linguistics, 2019), Vol. 1, pp. 4171–4186.
<https://doi.org/10.18653/v1/N19-1423>
7. C. Fellbaum, *WordNet: An Electronic Lexical Database* (MIT Press, Cambridge, Mass., 1998).
8. L. Flekova and I. Gurevych, "Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization," in *Proc. 54th Annu. Meeting of the Assoc. for Computational Linguistics, Berlin, 2016* (Association for Computational Linguistics, 2016), Vol. 1, pp. 2029–2041.
<https://doi.org/10.18653/v1/p16-1191>
9. I. M. Kobozeva, *Linguistic Semantics* (URSS, Moscow, 2004).
10. A. Kutuzov and M. Kunilovskaya, "Size vs. structure in training corpora for word embedding models: Araneum Russicum Maximum and Russian National Corpus," in *Analysis of Images, Social Networks and Texts. AIST 2017, Lecture Notes in Computer Science*, Vol. 10716 (Springer, Cham, 2017), pp. 47–58.
https://doi.org/10.1007/978-3-319-73013-4_5
11. H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab, "FlauBERT: Unsupervised language model pre-training for French," in *Proc. 12th Language Resources and Evaluation Conf., Marseille, 2020* (2020), pp. 2479–2490. <https://aclanthology.org/2020.lrec-1.302>
12. I. Leviant and R. Reichart, "Separated by an uncommon language: Towards judgment language informed vector space modeling," (2015).
<https://doi.org/10.3115/v1/w15-15>
13. N. V. Loukachevitch, A. A. Gerasimova, B. B. Dobrov, G. Lashevich, and V. V. Ivanov, "Creating Russian WordNet by conversion," in *Computational Linguistics and Intellectual Technologies: Proc. Int. Conf. Dialogue 2016, Moscow, 2016* (2016), pp. 22–30.
14. T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," (2013).
<https://doi.org/10.48550/arXiv.1309.4168>
15. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. on Neural Information Processing Systems, Lake Tahoe, Nevada, 2013*, Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Curran Associates, Red Hook, N.Y., 2013), pp. 3111–3119.
16. R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation, and application of a wide-coverage multilingual semantic network," *Artif. Intell.* **193**, 217–250 (2012).
<https://doi.org/10.1016/j.artint.2012.07.001>
17. A. Panchenko, D. Ustalov, N. Arefyev, D. Paperno, N. Konstantinova, N. Loukachevitch, and C. Biemann, "Human and machine judgements for Russian semantic relatedness," in *Analysis of Images, Social Networks and Texts*, Ed. by D. I. Ignatov, M. Yu. Khachay, V. G. Labunets, N. Loukachevitch, S. I. Nikolenko, A. Panchenko, A. V. Savchenko, and K. Vorontsov, *Communications in Computer and Information Science*, Vol. 661 (Springer, Cham, 2016), pp. 221–235.
https://doi.org/10.1007/978-3-319-52920-2_21
18. M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *J. Mach. Learn. Res.* **11**, 2487–2531 (2010).
19. S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual word embedding models," *J. Artif. Intell. Res.* **65**, 569–631 (2019).
<https://doi.org/10.1613/jair.1.11640>
20. F. Scozzafava, A. Raganato, A. Moro, and R. Navigli, "Automatic identification and disambiguation of concepts and named entities in the multilingual Wikipedia," in *Advances in Artificial Intelligence. AI*IA 2015*, Ed. by M. Gavanelli, E. Lamma, and F. Riguzzi, *Lecture Notes in Computer Science*, Vol. 9336 (Springer, Cham, 2015), pp. 357–366.
https://doi.org/10.1007/978-3-319-24309-2_27
21. J. Yamane, T. Takatani, H. Yamada, M. Miwa, and Yu. Sasaki, "Distributional hypernym generation by jointly learning clusters and projections," in *Proc. COLING 2016, the 26th Int. Conf. on Computational Linguistics: Technical Papers, Osaka, Japan, 2016* (The COLING 2016 Organizing Committee, 2016), pp. 1871–1879. <https://aclanthology.org/C16-1176>



Mikhail Tikhomirov (born in 1993) graduated from the Faculty of Computational Mathematics and Cybernetics of the Lomonosov Moscow State University, defended his dissertation for the degree of Candidate of Physics and Mathematics on the topic of “Methods of Automated Replenishment of Knowledge Graphs Based on Vector Representations” in 2022. Currently, he is an employee of the Research Computing Center of the Lomonosov Mos-

cow State University and is engaged in research in the field of automatic text processing. Research interests: natural language processing, neural network models, large language models, knowledge graphs.



Natalia V. Loukachevitch (born in 1964) graduated from the Faculty of Computational Mathematics and Cybernetics of the Lomonosov Moscow State University, defended her dissertation for the degree of Doctor of Engineering Sciences in 2016. Currently, she is a leading researcher at the Research Computing Center of the Lomonosov Moscow State University. Lectures on automatic text processing and information search at the Faculty of

Computational Mathematics and Cybernetics and the Faculty of Philology of the Lomonosov Moscow State University, as well as at the Bauman Moscow State Technical University. She has more than 300 publications in the field of automatic text processing, information search, and presentation of knowledge. Scientific interests: automatic text processing, information search, ontology.