Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2019”

Moscow, May 29—June 1, 2019

AUTOMATIC VOCABULARY POSITIONING IN A THESAURUS

Likhonosov A. (andrew.likhonosov@abbyy.com),
Indenbom E. (Eugene_I@abbyy.com),
Yudina M. (maria_yu@abbyy.com)

ABBYY, Moscow, Russia

Thesauri are one of the most widely used resources in natural language processing. At the same time, many of them are built manually, which takes a lot of time and, due to human errors, can affect their quality and completeness. We propose a procedure for automatic positioning of vocabulary in the ABBYY Compreno thesaurus using large monolingual corpora, a regular bilingual dictionary and a subset of already positioned words.

Key words: vocabulary positioning, thesaurus, word embeddings, supervised bilingual dictionary induction

АВТОМАТИЧЕСКОЕ ПОЗИЦИОНИРОВАНИЕ ЛЕКСИКИ В СЛОВАРЕ ТЕЗАУРУСНОГО ТИПА

1. Introduction

Thesauri are one of the most useful resources in natural language processing. However, most of them are crafted by hand, which brings up problems of incompleteness, human errors and time costs. A perfectly complete thesaurus is inherently impossible, as language changes with time, new words appear to describe new objects and phenomena, while others disappear. Moreover, human brains are not designed for enumerating objects, in this case, new words. So, a machine might be of great help.

The machine can crunch corpora with infinite patience and precision and produce a perfect list of unknown words and even position them into a thesaurus. A human linguist would only have to supervise the process.

On the other hand, available language resources (and thesauri in particular) are unequally distributed between languages. English resources are by far more rich, complete and diverse than resources for any other language. So, the problem of knowledge transfer to languages other than English is well recognized in linguistic community, and several attempts to automate the process has been made, mainly for WordNet (Fellbaum, 1998). For example, (Farreres et al., 1998) described the process for Catalan and Spanish. (Patel et al., 2018) did it for Hindi, they used the idea of linear transformation of word embeddings between languages, as originally proposed in (Mikolov et al., 2013b). And (Niemi et al., 2012) did the English-Finnish transfer, where they built the Finnish version of WordNet mostly manually and then used bilingual resources to extend and to improve it.

In building ABBYY Compreno Semantic Hierarchy (a kind of thesaurus, which is described below) we also face the problem of knowledge transfer from languages that we already have in our system to languages that are new to it. Doing it manually takes a lot of time and resources.

In this paper we describe a method for automatic positioning of a language vocabulary in a thesaurus using knowledge transfer technique from positioning of English and Russian languages in the same thesaurus. We report results for the ABBYY Compreno Semantic Hierarchy and the German language.

1.1. ABBYY Compreno Semantic Hierarchy

The ABBYY Compreno Semantic Hierarchy is the key element of the ABBYY Compreno linguistic model. It can be thought of as a tree of universal notions called "semantic classes". In a sense, these semantic classes are like Plato's "ideas", as opposed to real world objects, "shadows", but applied to natural language. For example, there is a semantic class "BULLDOG", which, as a pure idea, resides in the world of ideas, in universal language. You can think of a BULLDOG as a dog with all the necessary breed characteristics. But for real languages we have actual words, for example, "bulldog" in English and "Bulldogge" in German. So, for every language these semantic classes are filled with actual words. Since natural languages have synonymy, sometimes there are several words in a semantic class.

Apart from regular words and classes there are also "collocations" and "idioms" under semantic classes. Both collocations and idioms are stable multiword expressions, the difference between them lies in compositionality principle of semantics. Collocations can represent stable concepts, specifying in some way the meaning of the core, adding more information to it and thus forming a new concept: "буря:STORM" -> "песчаная буря". It is not rare that in other language the same notion is expressed by whole word: "снежная буря"—"blizzard" under semantic class 'SNOWSTORM'. On the contrary, the meaning of the idiom is never composed from the meanings of its parts ("белая ворона", "вбить в голову"). In the Semantic Hierarchy collocations are positioned under the semantic class of the root (or main) word. Idioms are positioned under semantic classes, according to the meaning of the whole expression. For more detailed information on how the Semantic Hierarchy is designed see [Manicheva et al., 2012], [Petrova et al., 2018], [Goncharova et al., 2015]. In this article we treat all collocations as their roots and idioms as distinct language units.

Another important property of the Semantic Hierarchy is its hierarchical structure. All classes are organized according to the hyper-hyponym relationship. To continue with our BULLDOG example, the semantic class can be found under the following path (arrow "->" designates the hyper-hyponym relation): PHYSICAL_OBJECT->BEING->ANIMAL->CHORDDATA->WARM_BLOODED->PREDATORS->CANIDAE->CANINAE->DOG->BULLDOG. A lot of semantic properties are described at the level of semantic classes. It allows us to describe a concept once and then all the descendants will inherit its semantic (and syntactic, when applied to a particular language) properties. This structure simplifies the process of semantic description, making the positioning of new words the most challenging part. As there are almost 200k semantic classes, this task, done manually, is quite resource-intensive, and there is need for optimizations.

1.2. Word Embeddings

Distributional vector space semantic models, or word embeddings, prove to be useful in many natural language processing tasks and are de facto standard for modern deep learning researches in NLP domain [Collobert et al., 2011]. According to the distributional hypothesis [Harris, 1954], words with similar distribution tend to have similar meanings, thus, can be considered synonyms. Mikolov [Mikolov et al., 2013a] suggested a method that scales well on billion-word size corpora and allows to capture distributional properties of a huge vocabulary. Basing on Harris hypothesis, we assume that word embeddings encode word relationship information and can be used to position words into a thesaurus.

As we aim to map words (and word embeddings) of a 'new' language into an existing thesaurus with bindings to 'known' languages, the task is essentially cross-lingual knowledge transfer. A good overview of cross-lingual word embedding models is given in [Ruder et al., 2017]. Our own method is described below.

2. Proposed Method

In this paper we describe the following approach to automatic positioning of new words in the Semantic Hierarchy. We assume there is at least one (almost) fully described language in ABBYY Compreno (as of today, we consider the description of the first languages, i.e. Russian and English, to be almost full). From now on we will refer to it as 'source' language, and, similarly, to the language of interest—as 'target' language.

First, we train semantic class embeddings for the source language. We use our Compreno Syntactic and Semantic parser [Anisimovich, et al., 2012] to extract semantic classes from a big corpus. We train embeddings on these semantic classes as tokens using the SkipGram algorithm [Mikolov et al., 2013a]. During training all proper names were generalized to their hypernyms (e.g. 'Smith' => 'PERSON_BY_LASTNAME', ‘Intel’ => ‘ORGANIZATION_BY_NAME’). 82396 embedding were trained on the corpora of $3{,}5 \cdot 10^9$ words.

Second, we train lemma embeddings for the target language. Technically speaking, objects to be positioned in the Semantic Hierarchy are lexemes, but lemmas are rather good approximation for our purposes. Lexeme is a word with all its morphological forms with implied lexical grammatical characteristics, i.e. part of speech. Lemma is the text of the dictionary form of the lexeme. So, different lemmas must correspond to different lexemes, but different lexemes may share the same lemma (with, for example, different parts of speech). This can cause some problems with homonyms, for example, ‘address' could be either verb or noun. We cannot distinguish between such homonyms while training word embeddings, nor between their corresponding vectors. Since there is only one vector for “address”, it will represent some average meaning, which can be something completely different from what we expect. We decided not to deal with this problem in this paper and leave it for future research. The corpora used for lemma embedding training contained $5{,}7 \cdot 10^9$ words.

Finally, we train a binary classifier for pairs of Semantic Classes and Lemmas. The classifier, given a pair (SemanticClass, Lemma), will output a number between 0 and 1. This way, for each lemma from a target language, and a set of N hypotheses of semantic classes, we feed this classifier with pairs [(SemanticClass1, Lemma), (SemanticClass2, Lemma),…,(SemanticClassN, Lemma)], and use its output to sort the hypotheses and, finally, find the best semantic class candidates.

3. Experiments

3.1. Data preparation

As we stated earlier, our method requires at least one fully described language (which will be used as a source language), but in this paper we ended up using English and Russian as source languages. We trained semantic class embeddings on a big English corpus and we used a German-Russian bilingual dictionary. Since the word list of this dictionary was already manually positioned in the Compreno system, we use manual positioning data as reference markup for our evaluation. It would be interesting to measure how the two components—semantic class embeddings and a bilingual dictionary—affect the final quality. For example, it would be interesting to run the same experiment with semantic class embeddings trained on a Russian corpus and the same (or another) German-Russian dictionary, it is the task for the future research.

As previously mentioned, while training semantic class embeddings, we treat collocations and idioms in a special way: we extract semantic class which corresponds to the collocation (or idiom), not semantic classes of the collocation’s parts. Thus, for example, the whole idiom “to beat around the bush” will be extracted as the class “TO_BEAT_AROUND_THE_BUSH” which is located under the class “TO_EVADE”, and “BUSH” (as a plant) will not be extracted.

For the classification problem we generate a positive and a negative sample of pairs (Semantic class, Lemma). We use the ABBYY Lingvo Universal (De-Ru)[1] dictionary, which contains Russian translations for German words. An article in the Lingvo dictionary (Universal (De-Ru)) looks like this:

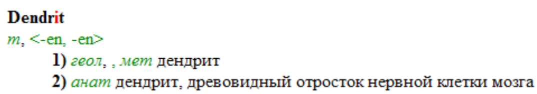
Dendrit
m, <-en, -en>
1) *геол.*, , *мет* дендрит
2) *анат* дендрит, древовидный отросток нервной клетки мозга

Figure 1

For "Dendrit" there are two classes in Semantic Hierarchy called "DENDRITE" and "DENDRITE_AS_CRYSTAL". These pairs (Dendrit, DENDRITE), (Dendrit, DENDRITE_AS_CRYSTAL) will be added to our positive sample.

For the negative sample we created a simple hypotheses generator.

3.2. Simple hypotheses generator

We propose the following procedure for generating hypotheses for German words. The ideal hypotheses generator should be simple but must produce all true and not so many negative classes, so that was what we were aiming for.

For the Russian part, we run the Compreno Syntactic and Semantic Parser and extract all possible classes for the Russian translation. For example, the first meaning from the dictionary article, "дендрит", is parsed like "DENDRITE" or "DENDRITE_AS_CRYSTAL", so these classes are added to the set of hypotheses. The second meaning, "дендрит, древовидный отросток нервной клетки мозга" is parsed like this:

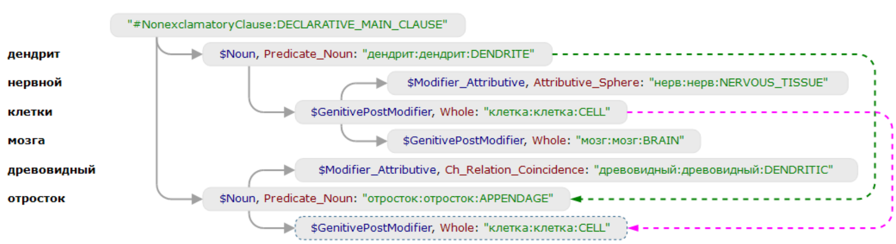

Figure 2

We don't pay any attention to the structure, but simply extract all classes for all words. This gives us another addition to the hypotheses set: {DENDRITE, NERVOUS_TISSUE, CELL, BRAIN, DENDRITIC, APPENDAGE, CELL}.

We found that adding neighbor classes (parents, or hypernyms, and children, or hyponyms) to the hypotheses set improves chances to generate all the true classes. But it expands the hypotheses set too much, which affects the final quality of the

[1] Universal (De-Ru) (for ABBYY Lingvo x6). The comprehensive German-Russian dictionary contains over 80,000 entries. © ABBYY, 2013, website: www.lingvo.ru/european/dictionary/

classifier. So, we decided to do it another way: we don't add neighbor classes to the hypotheses set, but we still consider these classes as true. Thus, we are allowing our hypotheses generator to be a little bit imprecise and be one-level wrong in the Semantic Hierarchy.

Then we parsed redirects—constructions like "см Apsis"—"see Apsis". We merge such redirects to the word where these redirects point to. This way, we added all hypotheses, as well as true classes, for "Apside" to the destination word "Apsis".

As in general there is no gold standard for morphological description, dictionary entries are not necessarily primary forms in our morphological system. Therefore if dictionary entry does not represent primary form in our morphological system we assign semantic classes extracted from dictionary article to all possible primary forms of the dictionary entry. For example, there are a lot of past participles (Partizip II) of German verbs in this Lingvo dictionary which may or may not have another translation. For example, "gezählt", being the past participle of "zählen", has also another meaning "встречающийся в самом большом [маленьком] количестве". So we take all the classes from this translation and add them to the lemma "zählen". And "gezählt" is removed completely from the list of lemmas.

Finally, for every word that begins with a capital letter (and there are a lot of them in German, since all nouns in German are capitalized) we add classes that contain named objects, like "PERSON_BY_NAME", "TOWN_BY_NAME" and so on. We needed them because the case when our hypotheses generator was not able to generate true classes for named objects was quite frequent.

This way, we have achieved a recall of 0.81, that is, our suggest generator was able to cover 81% of true classes across all the German words from the Lingvo dictionary.

3.3. Neural network architecture

For the classifier we used a simple feed-forward multilayer perceptron. It takes as input a concatenation of two embeddings: one for a target (German) word and the other for a semantic class. At the output layer, we have a single neuron with the sigmoid activation function. We interpret the value of this sigmoid as probability for a given pair (lemma, class) to be a good positioning suggestion. We use the log loss as a loss function in our neural network.

The best architecture was chosen by a randomized hyperparameter search procedure called Tree of Parzen Estimators (TPE). We used the implementation provided by a python package called Hyperopt [Bergstra, 2013]. The varying parameters included layers' neuron count, batch size, learning rate, activation function, optimizer algorithm, dropout rate and batch normalization momentum. The final architecture has 6 layers of 2048, 1024, 512, 256, 128 and 64 neurons, the learning rate of 0.01, the batch size of 128 examples, dropout rate of 0.3, leaky relu activation function with alpha parameter 0.1 and the Momentum optimizer.

3.4. Training

The whole dataset of positioning hypotheses (pairs of lemma and class, positive and negative) was split into three datasets: training, validation and test. We sort the German lemmas list by frequency and then perform the split based on lemmas' ranks.

We take first 1,000 lemmas for training. Then we split next 14,000 lemmas (from 1,000th to 15,000th) into parts: one for training, with 7,000 lemmas in it and the other part is added to the rest of lemmas. The rest of lemmas is split in half: one part for validation and the other for test dataset.

So we have 8k lemmas for training, and these lemmas are among the most frequent ones. In real world we usually start with description of some "core" words of a language, so by performing these manipulations we tried to simulate these conditions.

3.5. Results

We measure our classifier's performance on train, validation and test datasets. As we said above, our classifier evaluates pairs of lemmas and classes. Table 1 is an example output of our classifier for several German dictionary entries and a number of Semantic Classes. Suggests generator proposed from 10 to 400 Semantic Classes for each lemma. We sorted suggested Semantic Classes by score and cut the list at score 0.1 or last Semantic Class from markup, whichever comes last. Semantic Classes matching markup (the manually built Semantic Hierarchy contains the lemma in this Semantic Class) are marked 'Yes' in 'Markup' column and printed bold. Scores are given in probability scale.

Table 1

Lemma	Semantic Class	Representatives/Description	Score	Markup
Kranich	**CRANE**	'crane' (as a bird)	0.51	Yes
	CONSTELLATION_BY_NAME	'Grus'	0.18	Yes
	PERSON_BY_LASTNAME	surnames, i.e. 'Smith'	0.14	No
	PERSON_BY_FIRSTNAME	first names, i.e. 'Michelle'	0.05	No
	COMPANY_BY_NAME	company name, i.e. 'Intel'	0.02	No
Eruption	**TO_ERUPT**	'to erupt' (about a volcano)	0.71	Yes
	RASH	'rash' (spots on the skin)	0.57	Yes
	ACUTE_STAGE	'burst', 'explosion' (as top point of some process)	0.49	No
	LAVA	'lava' (as substance)	0.45	No
	TO_CUT_AS_TO_APPEAR	'eruption' (as process of appearance of the teeth)	0.42	Yes
	OUTLIER	'outlier' (as a sudden peak in a graph)	0.24	No
	FALLOUT	'fallout', 'emission' (as waste)	0.17	No
	TO_FLAKE	'to flake', 'to exfoliate'	0.14	No
	TO_THROW	'to throw', 'to heave', 'to toss'	0.13	No
	TO_POUR_SMTH_FRIABLE	'to dust', 'to strew'	0.06	No

Lemma	Semantic Class	Representatives/Description	Score	Markup
verschlucken	**TO_SWALLOW**	'to swallow', 'to gulp'	0.35	Yes
	TO_ABSORB	'to absorb', 'to ingest' (something inedible)	0.33	Yes
	TO_DEVOUR	'to guttle', 'to devour' (as eat greedily)	0.31	No
	TO_DIE_AWAY	'to muffle', 'to drown' (to diminish about sound and light)	0.30	No
	TO_TAKE	'to take' (in general meaning)	0.28	No
	TO_HIDE	'to hide', 'to conceal'	0.24	No
	ANGER	'anger' (as emotion)	0.20	No
	TO_MAKE_INVISIBLE	'to envelop', 'to haze'	0.18	No
	TO_TAKE_AWAY_BY_FORCE	'to bereave', 'to deprive'	0.15	No
	TO_SUPPRESS_FEELINGS	'to suppress', 'to swallow down'	0.10	Yes
	TO_BE_FULL_ABSORBED	'to absorb' (about work, activity)	0.10	No
	TO_CRITICIZE	'to criticize'	0.09	No
	TO_CONSUME	'to consume', 'to absorb' (about resources, i.e. fuel)	0.09	Yes
obsolet	**UP_TO_DATENESS**	'modern', 'outdated'	0.32	Yes
	TO_USE	'to use' (in general meaning)	0.27	No
	SUPERFLUOUS	'superfluous', 'excessive'	0.26	Yes
	TURN_OUT_AS_BE	'to turn out' (as to prove to be)	0.14	No
	TO_GET_RID_FROM_ DIFFICULTY	'to extricate' (to get someone out of a difficult or unpleasant situation)	0.08	No
Mysterium	**MYSTERY**	'mystery', 'secret'	0.57	Yes
	MYSTERY_AS_RELIGIOUS_ CEREMONY	'mystery', 'rite'	0.56	Yes
	SACRAMENT	'sacrament' (as rite)	0.03	No
	PERSON_BY_FIRSTNAME	first names, i.e. 'Michelle'	0.02	No
	PERSON_BY_LASTNAME	surnames, i.e. 'Smith'	0.02	No
Statistik	**STATISTICAL_DATA**	'statistics' (as data)	0.82	Yes
	STATISTICS	'statistics' (as science)	0.60	Yes
	DATA	'data'	0.50	No
	STATISTICIAN	'statistician'	0.31	No
	NATURAL_SCIENCE	sciences like 'physics'	0.28	No
	SCIENCE	'science' (in general meaning)	0.13	No
	QUALITY_PROPERTY	'characteristic', 'quality'	0.11	No

Lemma	Semantic Class	Representatives/Description	Score	Markup
Bestellung	**TO_ORDER_GOODS**	'to order' (as to buy something)	0.50	Yes
	TO_DELIVER	'to convey', 'to deliver'	0.28	No
	ORDER_AS_RESULT	'order' (as a result of making an order)	0.25	Yes
	WARRANT	'order', 'warrant' (as permission or command to do something)	0.23	No
	TO_INFORM	'to report', 'to inform'	0.18	No
	MESSAGE_COMMUNICATION	'message' (as quantum of information)	0.18	No
	TO_PROCESS_INFORMATION	'to process', 'to handle' (things like reports and claims)	0.15	No
	REPORT	'report' (as official description of something)	0.12	No
	TRANSPORT_COMMUNICATIONS	'service' (as transport communication)	0.11	No
	TO_GIVE	'to give' (in general meaning)	0.11	No
	ORDER	'order' (as a thing that is ordered or bought)	0.07	Yes

Table 2

Dataset	acc	precision	recall	f1	top1	top3	top5
Train	1.00	0.85	0.38	0.53	0.85	0.89	0.92
Dev	1.00	0.50	0.19	0.28	0.59	0.79	0.87
Test	1.00	0.47	0.22	0.30	0.61	0.80	0.88
test_with_true_class	1.00	0.51	0.20	0.29	0.59	0.78	0.86

In the Table 2 we summarized the results for our classifier on different parts of the dataset. Train, dev and test set were described earlier. The "test_with_true_class" is a filtered test dataset, containing only those lemmas for which our suggest generator was able to generate at least one true class. The accuracy, precision, recall and F1 are simple classifier metrics for classifying pairs (Lemma, Semantic class).

In the Table 3 we provide more information on the distribution of target lemmas in the test dataset and on our classifier performance on these parts of the test dataset. "Core lexis" column indicates that this part of dataset contains only core lexis, i.e. set of lemmas with frequency ranks less than 15,000. "Polysemous" indicates that words have more than one true class.

Table 3

core lexis	polysemous	lemmas	top1	top3	top5
yes	yes	1,102	0.71	0.60	0.70
yes	no	2,435	0.65	0.85	0.92
no	yes	1,475	0.57	0.57	0.70
no	no	22,244	0.56	0.78	0.86

TopN are metrics for ranking hypotheses and positioning. First, while calculating these metrics, we excluded lemmas for which either there is no true hypothesis, or the number of negative hypotheses is less than N. Thus, we leave only those German lemmas, where our method can fail. For example, if we included German lemmas with only 3 negative hypotheses and 2 positive ones, the top5 metric would always be 1, no matter how good or bad the final ranking is. TopN is the average of true hypotheses shares in the top N ranked hypotheses over all German lemmas:

$$top(N) = \frac{1}{lemmasCount}\sum_{lemma} trueHypothesesShareInTop(N, lemma), \text{ where}$$

$$trueHypothesesShareInTop(N, lemma) = \frac{numberOfTrueHypothesesInTop(N, lemma)}{\min(totalPositiveHypotheses, N)},$$

where $numberOfTrueHypothesesShareInTop(N, lemma)$ is the number of true hypotheses found in top N hypotheses in a ranked list of hypotheses.

So, for example, top1 of 0.61 on the test means that on the test dataset our classifier was able to rank hypotheses for 61% of lemmas so that a true class was on the top of the list. On average, we were able to guess 80% of classes across all lemmas in top3. And 88% of classes across all lemmas in top5. The positioning (topN) metrics on "test_with_true_class" dataset are two percent worse, which means that lemmas where our hypotheses generator produced a true class were positioned worse than when only negative hypotheses were generated and the true class was obtained from the Semantic Hierarchy. This means that we can position a significant amount of lexis totally automatically. The rest must be verified by a linguist, and for the 88% of cases the right decision lies within first 5 results and can easily be reached.

4. Conclusion

In this article we tried to apply embeddings to the problem of positioning new words in an existing thesaurus. Our method requires semantic-syntactic parsing for a large monolingual corpus of texts in the source language (English in our case), morphological parsing for a large monolingual corpus in the target language (we used German), a number of already positioned words and a regular bilingual dictionary (German-Russian) for hypotheses generation. We expect our method to position automatically about 61% of lemmas of a new language. On average, lemmas will be assigned to 88% of correct classes when using top-5 results.

We see two main directions for further research. Firstly, we plan to compare performances of our classifier on different source language data (different corpora for semantic class embeddings and different bilingual dictionaries). Secondly, we plan to run similar experiments using words alignment statistics from parallel corpora instead of manually positioned core lexis words, avoiding the necessity to position initial word set manually.

All in all, our results show that the process of positioning (and thus adding) new lexis to the thesaurus-like dictionary can be automated to a significant extent. This is crucial when we deal with a language that is completely new to the linguists involved. Otherwise it speeds up and simplifies considerably the work on the lexical description of a language that is new to the system.

References

1. *Anisimovich, K., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., & Zuev K. A.* (2012). Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. Computational linguistics and intellectual technologies: Proceedings of the International Conference "Dialog 2012", Vol. 2, pp. 91-103.
2. *Bergstra, J. Y.* (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. 30th International Conference on Machine Learning (ICML 2013).
3. *Collobert, R., Weston, J., Bottou, L., Karlen, M., & Kuksa, P.* (2011). Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research 12(12).
4. *Farreres, X., Rigau, G., & Rodríguez, H.* (1998). Using WordNet for Building WordNets. WordNet@ACL/COLING.
5. *Fellbaum, C.* (1998). A semantic network of English verbs, WordNet: An electronic lexical database., (pp. pp. 153–178.).
6. *Goncharova, M., Kozlova, E., Payukov, A., Garashchuk, R., & Selegey, V.* (2015). Model-Based WSA as Means of New Language Integration into a Multilingual Lexical-Semantic Database with Interlingua. Papers from the Annual International Conference "Dialogue", (pp. vol. 1, pp. 169-182).
7. *Harris, Z. S.* (1954). Distributional structure. Word, 10(23), 146-162.
8. *Manicheva, E., Petrova, M., Kozlova, E., & Popova, T.* (2012). The Compreno Semantic Model as an Integral Framework for a Multilingual Lexical Database. Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon, 215-230.
9. *Mikolov, T., Chen, K., Corrado, G., & Dea, J.* (2013a). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR.
10. *Mikolov, T., Le, Q. V., & Sutskever, I.* (2013b). Exploiting similarities among languages for machine translation.
11. *Niemi, J., Linden, K., & Hyvarinen, M.* (2012). Using a Bilingual Resource to Add Synonyms to a Wordnet: FinnWordNet and Wikipedia as an Example.
12. *Patel, K., Kanojia, D., & Bhattacharyya, P.* (2018). Semi-automatic WordNet Linking using Word Embeddings.
13. *Petrova, M., Druzhkina, A., Garashchuk, R., & Yudina, M.* (2018). Semi-automatic Integration of a new Language into a multilingual NLP model: the case of Japanese. Proceedings of the International Conference “Dialogue 2018”. Moscow.
14. *Ruder, S., Vulić, I., & Søgaard, A.* (2017). A Survey of Cross-lingual Word Embedding Models. The Journal of Artificial Intelligence Research.