Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2017”

Moscow, May 31—June 3, 2017

TESTING FEATURES AND METHODS IN RUSSIAN PARAPHRASING TASK

Loukachevitch N. V. (louk_nat@mail.ru),
Shevelev A. S. (alex.shevelev@hotmail.com),
Mozharova V. A. (joinmek@rambler.ru)

Lomonosov Moscow State University, Moscow, Russia

In this paper we study several groups of features and machine learning methods in the shared task on Russian paraphrasing organized in 2016. We use four groups of features: string-based features, information-retrieval features, part-of-speech features and thesaurus-based features and compare three machine learning methods: SVM with linear and RBF kernels, Random Forest and Gradient Boosting. In our experiments, the best results were obtained with the Random Forest classifier with parameter tuning and using all groups of features. The results of Gradient Boosting with parameter tuning were slightly worse.

Keywords: paraphrasing, semantic similarity, machine learning, thesaurus

ИССЛЕДОВАНИЕ ПРИЗНАКОВ И МЕТОДОВ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ ПАРАФРАЗ ДЛЯ РУССКОГО ЯЗЫКА

Лукашевич Н. В. (louk_nat@mail.ru),
Шевелев А. С. (alex.shevelev@hotmail.com),
Можарова В. А. (joinmek@rambler.ru)

МГУ им. М. В. Ломоносова, Москва, Россия

Ключевые слова: парафразы, семантическое сходство, машинное обучение, тезаурус

1. Introduction

Accounting paraphrases and synonyms is crucially important for various natural language applications such as machine translation (Marton et al., 2009), information retrieval and question answering (Fader et al., 2013), text summarization (Nenkova, McKeown, 2012; Loukachevitch, Alekseev, 2012), document clustering (Vossen et al., 2014), plagiarism measuring (Clough et al., 2002), etc.

Data for paraphrase detection can be found in synonym dictionaries, thesauri such as WordNet, or crowdsourced resources as Wikipedia. Also specialized databases with automatically collected paraphrases have been created (Dolan et al., 2004; Pavlick et al., 2015). Large text corpora can be processed to extract information on semantic similarity between words or expressions using similarity between their contexts (Przybyla et al., 2016). In practice paraphrase detection is based on large variety of sentence features (Kozareva, Montoyo 2006).

In this paper we describe results of exploiting several groups of features to detect paraphrased sentences in Russian. We are most interested in using semantic features calculated on the basis of RuThes thesarus (Loukachevitch, Dobrov, 2014). We also study several machine learning methods in this task: SVM, Random Forest, and Gradient Boosting. The evaluation is carried out on the data of the Russian Paraphraser corpus (Pronoza, Yagunova, 2016; Pivovarova et al., 2016).

2. Related Work

Most papers on English paraphrasing have been evaluated on Microsoft Research Paraphrase Corpus (Dolan et al., 2004), which comprises 5,081 paraphrase sentence pairs. The sentences pairs have been manually annotated into two classes: paraphrases or not. The data contain 67% positive examples of paraphrases and 33% of non-paraphrases. The data have been arbitrarily split into a training set containing 4,076 examples and a test set containing 1,725 examples. Evaluation of approaches to semantic textual similarity is also organized in the framework of SemEval conference (Agirre et al., 2016).

Most approaches to paraphrase detection exploit the following groups of features and combine them with machine learning methods (Kozareva, Montoyo 2006):

- various measures of word and character similarities, including length features, longest common sequence, n-gram overlap features, edit distances, machine translation similarities (BLUE, WER, TER, ROUGE-L etc.), information-retrieval measures (tf-idf, BM25), named entity similarity (Brychcın, Svoboda 2016);
- features of lexical differences between sentences including parts of speech tags, named entities, meaningful words (Pronoza, Yagunova, 2015a);
- syntactic features based on similarity between dependency trees;
- semantic measures based on WordNet conceptual structure (Mihalcea et al. 2006; Fernando, Stevenson, 2008);
- corpus-based similarities using classical distributional vectors or distributed representations of words learned by neural networks on a large text corpus (Przybyla et al., 2016);

Last successful approaches in paraphrase detection combine neural networks, comparison of dependency trees and semantic measures based on WordNet similarity (Rychalska et al., 2016; Brychcın, Svoboda 2016).

The previous work on semantic approaches for paraphrasing in Russian includes the work by Dobrov and Pavlov (2010) who studied the contribution of synonyms described in the Socio-political thesaurus for Russian news document clustering. With this aim, they created the conceptual index where each concept contains all known synonyms for news texts. For evaluation, the collection of news documents from ROMIP (Russian Information Retrieval Seminar)[1] was used. They found that the use of the conceptual index improves the best achieved result of news clustering (if compared with clustering based on words in the text body and the header) by 5.5%.

Pronoza and Yagunova study (2015a) various factors of paraphrase detection on the Russian paraphrase corpus including shallow measures based on word or characters overlap, dictionary-based measures and distributional semantic measures based on finding context similarity between words in a text corpus. They experimented on the Russian paraphrase corpus containing 6,281 sentence pairs (1,482 precise, 3,247 loose and 2,209 non-paraphrases). Altogether more than 80 features of sentences were calculated and combined with the Gradient Boosting classifier. The similarity between synonyms in a dictionary was based on calculating the probability to meet the words in the same set of synonyms.

In 2016 the shared task on evaluation of methods for detecting Russian paraphrases has been organized (Pivovarova et al., 2016).

3. Russian Paraphrase Evaluation: Tasks, Data, Results

The main task in the evaluation was three-way classification of sentence pairs: precise, loose and non-paraphrases on the specially created Paraphraser corpus (Pivovarova et al., 2016). The task of binary classification was also considered: sentence pairs should be classified to paraphrases or non-paraphrases.

The participating teams should take a pair of sentences as an input and return the similarity class as a response. Participants could submit “standard” runs that utilize as training data only the ParaPhraser corpus and (or) manual dictionary resources, and “non-standard” runs that may use any other data. “Standard” and “non-standard” run have been evaluated separately.

The datasets were formed on the basis of news story headlines. The training collection contains about 7,000 sentence pairs. Each candidate pair was manually annotated by three native speakers with the use of crowdsourcing. The test dataset (Gold standard set) contains 1,924 sentence pairs.

[1] http://www.romip.ru/

Table 1. Russian Paraphrase Evaluation Dataset Statistics

Paraphrases	Training set	Gold standard set
Exact	1,662 (23%)	374 (19.4%)
Loose	3,644 (41%)	778 (40.4%)
Non-paraphrases	1,921 (36%)	772 (40.2%)
Total	7,227	1,924

The quality of submitted results has been assessed with Accuracy and macro F-measure. At present, the evaluation results are published only in the electronic form[2].

4. Features for Paraphrase detection

For finding paraphrases, we use four groups of features and study results for three machine-learning methods (SVM, Gradient Boosting and Random Forest) in dependence of different parameters.

Table 2. The best results achieved at Russian Paraphrase Evaluation

Task	Accuracy	F-measure
Three-class, standard	59.01	56.92
Three-class, non-standard	61.81	58.38
Two-class, standard	74.59	80.44
Two-class, non-standard	77.39	81.10

The features include the following groups: string-based features, information-retrieval features, part-of speech features, and thesaurus-based features.

String-based features include features for two and three symbol N-grams, and for word one, two and three N-grams. For each type of N-grams, the string feature group comprises the following three features:

$$feature_1 = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

$$feature_2 = \frac{|S_1 \cap S_2|}{|S_1|}$$

$$feature_3 = \frac{|S_1 \cap S_2|}{|S_2|}$$

where S_1 is a set of character of word N-grams of Sentence 1; S_2 is a set of character of word N-grams of Sentence 2, $|S|$ is the number of elements in the set S.

Information-retrieval (IR) features include BM25 and IDF features computed on the train collection (Manning et al., 2008). The BM25 feature compares similarity

[2] http://www.paraphraser.ru/contests/result/?contest_id=1

of two sentences, using formula (*). The IDF features (Formula **) are calculated for the word difference between two sentences: maxIDF is the maximal idf for words in the sentence difference, sum IDF is the sum of all idf of words from the sentence difference. Calculating IDF, we suppose that the loss of frequent words in the difference between sentences may be not very meaningful.

$$BM25(S_1, S_2) = \sum_{i=1}^{n} IDF(w_i) * \frac{TF(w_i,S_2)*(k+1)}{TF(w_i,S_2)+k*(1-b+b*\frac{|S_2|}{avg})} \quad (*)$$

$$IDF(w_i) = \log\frac{N-N(w_i)+0.5}{N(w_i)+0.5} \quad (**)$$

where $TF(w_i, S)$ is the frequency of word w_i in sentence S, N is the number of sentences in the training collection, $N(w_i)$ is the number of sentences containing word w_i, $|S|$ is the length of a sentence in words, *avg* is the average length of a sentence in the collection, k and b are parameters, their standard variants ($k = 1.2$, $b = 0.75$) are used (Manning et al., 2008).

Part-of-speech (POS) features are binary features that indicate what parts of speech are found in the difference between sentences. Five part-of-speech features show the presence of nouns, verbs, adjectives, adverbs and all other functional parts of speech in sentence difference.

Thesaurus (Thes) features are calculated on the basis the RuThes thesaurus (Loukachevitch, Dobrov, 2014). They will be described in the next section.

5. Semantic (Thesaurus) Features for Paraphrase Detection

It is useful to use semantic information about synonyms and semantically related language units to detect similarities between phrases. With this aim, we utilize RuThes thesaurus (Loukachevitch, Dobrov, 2014). The publicly available version of the RuThes thesaurus, RuThes-lite 2.0, comprises 31.5 thousand concepts, 115 thousand Russian words and expressions[3]. RuThes is a linguistic ontology, hierarchical net of concepts. It has many similarities with the Princeton Wordnet (Fellbaum, 1998) structure, therefore approaches for calculating semantic similarity proposed for wordnets can be applied to RuThes also.

We calculated several lexical similarity measures proposed for Princeton WordNet. These measures exploit paths between concepts where words under comparison were assigned. The measures include Leacock-Chodorow measure (*Lch*), Lin measure (*Lin*), and Jiang-Conrath measure (*Jcn*) (Budanitsky, Hirst 1998).

The *Lch* measure estimates the similarity of two nodes by finding the path length between them in the is-a hierarchy. It is computed as:

$$sim_{lch} = -\log\frac{N_p}{2D}$$

where *Np* is the distance between nodes and *D* is the maximum depth in the taxonomy. The distance is calculated in nodes, that is the distance between synonyms is equal 1, and the distance between a node and its hypernym is equal 2. We used two variants

[3] http://www.labinform.ru/ruthes/index.htm

of calculation of this measure: 1) using only hyponym-hypernym relations (Lch_1) and 2) using hyponym-hypernym and part-whole relations (Lch_2). In RuThes, the transitivity of part-whole relations is supported (Loukachevitch, Dobrov 2015), therefore multi-step paths of part-whole relations and their combination with hyponym-hypernym relations are also meaningful. In RuThes-lite, the maximum depth of the ontology accounting both types of relations is equal 14. The logarithm base is equal to 2*D*.

Other two measures are calculated on the basis of word probabilities and so called information content (IC). For every word the probability to meet this word in a corpus is calculated:

$$P(w) = \frac{Freq_w}{N}$$

where N is the size of a corpus in words. The probability of a concept is the sum of probabilities of all text entries assigned to this concept.

The information content of a concept is an estimate of how informative the concept is. It is supposed that frequently occurring concepts have low information content and rarely occurring concepts have high information content.

$$IC(c) = -\log(P(c))$$

In calculating information content, probabilities of all lower concepts in the hierarchy should be summed up. The *Lin* measure is calculated as follows:

$$sim_{lin} = \frac{2 \cdot IC(LCS(C_1, C_2))}{IC(C_1) + IC(C_2)}$$

where *LCS* is the least common subsume of *C*1 and *C*2.

$$sim_{jcn} = \frac{1}{IC(C_1) + IC(C_2) + 2 \cdot IC(LCS(C_1, C_2))}$$

For *Lin* and *Jcn* measures, two variants were also calculated: with and without accounting part-whole relations.

To estimate word frequencies for *IC* calculation, an additional news corpus was used. Therefore according to the evaluation rules, when we use the *Lch* measure, the run could be considered as standard. But when we use the *Lin* or *Jcn* similarity measures, these runs should be categorized as non-standard due to the use of the additional corpus.

Comparing sentences on the basis of thesaurus similarity, we use the approach proposed in (Fernando, Stevenson, 2008) that allows summing the similarity of a word in one sentence with several words from another sentence. Sentences in this approach are represented as binary vectors $\vec{a}$ and $\vec{b}$. The similarity between the sentences is calculated as follows:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} W \vec{b}}{|\vec{a}||\vec{b}|}$$

where *W* is a square matrix of the calculated similarities between words and expressions found in both sentences.

Each w_{ij} in W represents the similarity of words w_i and w_j according to some lexical similarity measure. In our case the measures are symmetric, i.e. $w_{ij} = w_{ji}$ and the matrix is also symmetric. Diagonal elements represent self similarity and have the greatest values equal to 1.

Table 3. Matrix of *Lch* similarity for the example sentences

	Деми	*Мур*	*Украсть*	*Похитить*	*Одежда*
Деми	1	0	0	0	0
Мур	0	1	0	0	0
Украсть	0	0	1	0.7941	0
Похитить	0	0	0.7941	1	0
Одежда	0	0	0	0	1

As preprocessing, before thesaurus features calculating, sentences are lemmatized, function words are removed, numbers mentioned in sentences are substituted with corresponding words. Words not found in the thesaurus but met in both sentences have maximal similarity 1.

For example, if two sentences are considered:

(s1) *У Деми Мур украли одежду. (Demi Moor's clothes were stolen)*

(s2) *У Деми Мур похитили одежду. (Demi Moor's clothes were robbed)*

The matrix according the *Lch* measure is presented in Table 3. Words "*Деми*" and "*Мур*" are absent in the thesaurus but mentioned in both sentences. The different words *украсть* and *похитить* are linked with the hyponym-hypernym relation and have high semantic similarity according to the *Lch* measure.

6. Experiments and Results

Before comparison, all sentences were lemmatized and the part-of speech information was extracted for each word. In preliminary experiments, we chose Random Forest as a basic machine learning method. We used the implementation from scikit-learn package[4].

Table 4. Best results achieved using Random Forest learning (grid parameter tuning)

	Heldout set		**Gold standard set**	
Features	**Accuracy**	**F-measure**	**Accuracy**	**F-measure**
1) String-based	63.34	61.42	60.03	57.99
2) 1)+BM25	64.59	62.76	60.55	58.67
3) 2)+ Max idf	64.59	62.67	60.96	58.99
4) 3)+POS features	**65.76**	**63.87**	61.07	59.03

[4] http://scikit-learn.org/stable/index.html

Features	Heldout set		Gold standard set	
	Accuracy	F-measure	Accuracy	F-measure
5) 4)+$Thes_{lch}$	65.35	63.56	61.48	59.33
6) 5)+$Thes_{jcn}$	65.28	63.35	**62**	**60.03**

Table 5. Results achieved with the default parameters

Methods and Parameters	Features	Accuracy	F-measure
SVM linear Default parameters (C=1, penalty=L2)	String-based	59.82	56.54
	String-based+IR	60.86	57.49
	String-based+IR+POS	60.60	57.36
	String-based+IR+POS+Thes	**61.43**	58.10
SVM rbf Default parameters C=1	String-based	58.99	56.95
	String-based+IR	59.77	57.77
	String-based+IR+POS	59.82	56.72
	String-based+IR+POS+Thes	60.49	57.62
Random Forest Default parameters N-estimators=10 Min_samples_leaf=10	String-based	54.88	52.61
	String-based+IR	57.38	54.76
	String-based+IR+POS	57.43	55.66
	String-based+IR+POS+Thes	56.65	54.60
Gradient Boosting Default parameters N_estimators = 100 min_samples_leaf = 1 max_depth = 3 learning_rate = 0.1	String-based	59.56	57.55
	String-based+IR	59.51	57.95
	String-based+IR+POS	60.91	58.89
	String-based+IR+POS+Thes	60.86	**59.11**

The parameters of the method were tuned with `GridSearchCV`[5] function. This function generates candidates from a grid of parameter values specified with the `param _ grid` parameter. All the possible combinations of parameter values are evaluated and the best combination is retained. In our case for tuning parameters, the training set was subdivided into the cross-validation dataset and the heldout set. The parameters were tuned on the cross-validation dataset with the cross-validation technique and tested on the heldout set.

Table 5 contains the achieved results on the heldout set and the gold standard set for Random Forest with parameter tuning. It can be seen that string-based features allows obtaining the significant level of the result. If to compare with the Paraphrase evaluation results (Table 2), it can be noted that the string-based features with tuned Random Forest overcome the results reported in the evaluation (Standard variant). Other groups of the proposed features gave further improvement of the results.

[5] http://scikit-learn.org/stable/modules/grid_search.html

Table 6. Results achieved with grid parameter tuning: SVM (linear, RBF), Gradient Boosting

Methods and Parameters	Features	Accuracy	F-measure
SVM linear Grid tuning, C=0.4,0.7, 0.2, 0.2 Penalty L2	1) String-based	59.92	56.71
	2) String-based+IR	60.86	57.52
	3) String-based+IR+POS	60.75	57.54
	4) String-based+IR+POS+Thes	61.64	58.52
SVM rbf Grid tuning C=1.5, 100, 70, 0.6 Gamma=0.01, 0.1	1) String-based	59.25	57.29
	2) String-based+IR	57.38	54.72
	3) String-based+IR+POS	58.00	54.85
	4) String-based+IR+POS+Thes	59.61	57.32
Gradient Boosting Grid tuning	1) String-based	60.13	58.17
	2) String-based+IR	60.55	58.65
	3) String-based+IR+POS	61.56	59.05
	4) String-based+IR+POS+Thes	**61.93**	**59.92**

We experimented with different sets of the thesaurus features. The best result (BestOfThesaurus) in combination with features of other groups was obtained using four thesaurus features: two variants of similarity based on the *Lch* measure (with and without accounting part-whole relations) and two variants of similarity based on the *Jcn* measure (Run 6 in Table 4).

For each run, the parameters of Random Forest were tuned separately. The number_of_ esimators parameter were changed from 100 till 500, and the min-samples_ leaf parameter was equal to 15 or 20.

After obtaining the results with tuned Random Forest, we compared the results of other machine learning methods working with the same feature set. We considered SVM (linear kernel and *radial basis function* kernel (RBF)) and Gradient Boosting. All methods were compared in two main regimes: with default parameters (Table 5) and with grid-tuned parameters (Table 6).

From Table 5, we can see that linear SVM achieves the performance close to the best result in Accuracy, and Gradient Boosting Method is enough close to the best result in F-measure without any tuning.

Table 6 shows the performance of the SVM and Gradient Boosting methods on the same features with tuned parameters. For Linear SVM and Gradient Boosting, the results slightly improved (if compared with default values of parameters) but were not better than for Random Forest. The parameter tuning for the rbf variant of SVM did not allow achieving better results on the Gold Standard set than with default parameters.

It also can be seen that the value of C-parameter for Linear SVM was always less than the default value (1). The C parameter for rbf SVM behaves unstable changing from 0.6 till 100.

Conclusion

In this paper we studied several groups of features and machine learning methods in the shared paraphrasing task in Russian organized in 2016. We used four groups of features: string-based features, information-retrieval features, part-of-speech features and thesaurus-based features and compared three machine learning methods: SVM with linear and RBF kernels, Random Forest and Gradient Boosting.

In our experiments, the best results were obtained with the Random Forest classifier with parameter tuning and using all groups of features. Each group of features improved the performance of paraphrase detection. The results of Gradient Boosting with parameter tuning were slightly worse.

Acknowledgments

This work is supported byRussian Science Foundation grant № 16-18-02074.

References

1. *Agirre E., Banea C., Cer D., Diab M., Gonzalez-Agirre A., Mihalcea R., Wiebe J.* (2016), Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. Proceedings of SemEval, pp. 497–511.
2. *Afzal N., Wang Y., Liu H.* (2016), MayoNLP at SemEval-2016 Task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model, Proceedings of SemEval-2016, pp. 674–679.
3. *Brychcın T., Svoboda L.* (2016), UWB at SemEval-2016 Task 1: Semantic textual similarity using lexical, syntactic, and semantic information, Proceedings of SemEval-2016, pp. 588–594.
4. *Brockett, C., Dolan, W. B.* (2005), Support vector machines for paraphrase identification and corpus construction, Proceedings of the 3rd International Workshop on Paraphrasing, pp. 1–8.
5. *Budanitsky A., Hirst G.* (2006), Evaluating wordnet-based measures of lexical semantic relatedness, Computational Linguistics, Vol 32, №. 1, pp. 13–47.
6. *Clough P., Gaizauskas R., Piao S., Wilks Y.* (2002), METER: MEasuring TExt Reuse, Proceedings of the 40th Anniversary Meeting for the Association for Computational Linguistics (ACL-02), pp. 152–159.
7. *Dobrov B., Pavlov A.* (2010), Basic line for news clusterization methods evaluation, Proceedings of the 5-th Russian Conference RCDL-2010, pp.287–295.
8. *Dolan W. B., Quirk C., Brockett C.* (2004), Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources, Proceedings of the 20th International Conference on Computational Linguistics Coling-2004, Geneva, Switzerland.
9. *Fellbaum Ch* (ed.). (1998), WordNet: An Electronic Lexical Database, The MIT Press.
10. *Fader A., Zettlemoyer L. S., Etzioni O.* (2013), Paraphrase-Driven Learning for Open Question Answering, Proceedings of ACL- 2013, pp. 1608–1618.
11. *Fernando S., Stevenson M.* (2008), A semantic similarity approach to paraphrase detection, Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, pp. 45–52.

12. *Kozareva Z., Montoyo A.* (2006), Paraphrase identification on the basis of supervised machine learning techniques, Advances in natural language processing. Springer Berlin Heidelberg, 2006. pp. 524–533.
13. *Loukachevitch N., Dobrov B.* (2014), RuThes Linguistic Ontology vs. Russian Wordnets, Proceedings of Global WordNet Conference GWC-2014, pp. 154–162.
14. *Loukachevitch N., Dobrov B.* (2015), The Sociopolitical Thesaurus as a resource for automatic document processing in Russian, Terminology. Special issue Terminology across languages and domains, Vol. 21, N 2, pp. 238–263.
15. *Loukachevitch N., Alekseev A.* (2012), Summarizing News Clusters on the Basis of Thematic Chains, Proceedings of LREC-2012, pp.1600–1607.
16. *Manning, C. D., Raghavan, P., Schütze, H.* (2008), Introduction to information retrieval. Cambridge: Cambridge University Press.
17. *Marton Y., Callison-Burch C., Resnik P.* (2009), Improved statistical machine translation using monolingually-derived paraphrases, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing EMNLP-2009, pp. 381–390.
18. *Mihalcea R., Corley C., Strapparava C.* (2006), Corpus-based and Knowledge-based Measures of Text Semantic Similarity, Proceedings of the American Association for Artificial Intelligence (AAAI 2006).
19. *Nenkova A., McKeown K.* (2012), A Survey of Text Summarization Techniques. Mining Text Data Book, Springer US, pp. 43–76.
20. *Pavlick E., Rastogi P., Ganitkevitch J., Durme B., Callison-Burch Ch.* (2015), PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification, Proceedings of ACL-2015 and the 7th International Joint Conference on Natural Language Processing, Vol. 2, pp. 425–430.
21. *Pronoza, E., Yagunova, E.* (2015a), Low-Level Features for Paraphrase Identification, Mexican International Conference on Artificial Intelligence, Springer International Publishing. pp. 59–71.
22. *Pronoza E., Yagunova E.* (2016), Construction of a Russian paraphrase corpus: unsupervised paraphrase extraction, Information Retrieval, Springer International Publishing, pp. 146–157.
23. *Pivovarova L., Pronoza E., Yagunova E.* (2016), Shared Task on Sentence Paraphrase Detection for the Russian Language, http://www.paraphraser.ru/download/get?file_id=2
24. *Przybyla, P., Nguyen, N., Shardlow, M., Georgios K. Ananiadou, S.* (2016), NaCTeM at SemEval-2016 Task 1: Inferring sentence-level semantic similarity from an ensemble of complementary lexical and sentence-level features, Proceedings of the 10th International Workshop on Semantic Evaluation SemEval-2016, pp. 614–620.
25. *Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., Andruszkiewicz, P.* (2016), Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, CA, USA.
26. *Vossen, P., Rigau, G., Serafini, L., Stouten, P., Irving, F., van Hage, W. R.* (2014), NewsReader: recording history from daily news streams, Proceedings of LREC-2014, pp. 2000–2007.