Corpus-Based Text Retrieval and Adaptation for Learning System

Nikolay Karpov

***Abstract*—The algorithm to adapt lexical complexity in the news article which can be used as materials for learning language presented in the paper. We consider words substitution retrieval according to wordnet-based and corpus-based semantic relatedness. Two corpus-based similarity measures empirically tested: Vector Space Model and Distributional Semantic Model. This language processing algorithm has created as a client-server application. It retrieves appropriate text from Web-resource. Next it performs adaptation procedure.**

***Keywords*—distance learning, foreign language, distributional semantic model, contextual proximity.**

I. Introduction

Customized educational distance learning solutions for corporations, governments, non-profits and students alike, such as NYSEBA[1] and other[2] become popular. They provide complex of educational services in one virtual place. Such multifunctional complexes consist of a set of different software, hardware and human resources, which interact with each other in a complicated way. User interface must be simple and convenient. New information technology for electronic language tutorial, can offer many advantages over traditional textbook. It can be felt in many areas such as pronunciation training [8], translation, giving explanation and synonyms.

In this study we discuss algorithms and technology which are helpful to build interactive learning system. This system retrieves texts from news articles like in other studies [17, 19]. We automatically adapt text to a lower level of competence [10, 11] in Russian language.

This paper is organized as follows: first, below we present a common structure the technology we used for the learning system (Section II). Then, we investigate empirically how experts carry out text simplification (Section III).

In Section IV we formulate lexical adaptation algorithm. Section V and VI are devoted to contextual proximity of words using Vector Space Model and Distributional Semantic Model respectively. Section VII summarizes the study.

Nikolay Karpov
National Research University Higher School of Economics
Russia

nkarpov@hse.ru

1 http://www.nyseba.com
2 http://lms.hse.ru

II. Technology design of Learning System

As a platform for filling electronic distance course in the electronic format an open source e-learning system eFront[3] was chosen. Using this system teachers and scientists can easily design and publish educational materials in electronic form.

Structure of educational material was developed as a tree-like structure with theoretical and practical part of the textbook. The main component of a theoretical part of the book is the rules of the use of the prefix to the verb and a number of examples. These examples include materials selected by the author to illustrate the usage of words. The practical part of the tutorial we organized the ability to get the «live» examples from the news. Such «live» examples are extracted in the real-time mode from renewing Russian National Corpus[4], and therefore the content always varied, timely and actual examples of words use.

The program, which extracts «live» examples from news for the electronic tutorial of the Russian language, works on the server and complies with a CGI standard. We created such a server application, with the help of the Python language and did not use external tools.

It was divided into separate sub-tasks. The first sub-task - getting search results from the Corpus, solved by sending the specified HTTP request to the server and parsing coming response from it. To prevent finding wrong words we use metadata, which provides a limitation of grammatical features. For instance: a verb in the indicative mood. We specify the step search of a preposition from the verb (one or two words), because after the verb and before the preposition we often use a direct object (pushed HIM/BOY into the room).

Each of the section in the tutorial has its own query. The query was developed and optimized, thus, to always receive a request, the most relevant to a specified section in the tutorial. It is possible mistakenly to include in a tutorial small percentages of examples. This is a downside to use «live» search examples from Corpus. This shortcoming was minimized with the help of a multiple-page viewing search results in repeated request to the server.

At the second stage, we selected interesting examples from the server. For this, we use the settings: select a single example of one of the author and ordered, by date of creation. Also we set the number of examples on the page and requested variant from a newspaper sub-corpus.

3 http://efrontlearning.net
4 http://www.ruscorpora.ru/en/index.html

Response from Corpus is stored and transmitted to the input of a finite state machine, which implemented the second task – the obtained results analysis to create actual examples. State machine is designed to extract examples and highlight the key words and cut off unnecessary information. For the implementation of each of functional capabilities, we use a separate pair of states of a finite state machine. Alphabet of a finite state machine is HTML tags and character set of any human language in the used encoding.

A third sub-task - output of search results structured as a HTML text. It was implemented in such a way that finite state machine could extract various types of words. For each of the word type we have a method of realization in HTML. These chunks are joined and added to the template of the page. After it we create the page and transfer information to the client computer.

III. Analysis of Methods for Manual Text Adaptation

In order to build an automated system, we have investigated empirically how experts carry out such text simplification. First, we formed a set of news articles on various subjects. The set included ten texts from the RIA information agency.[5] The articles were then adapted by two independent experts to suit the B1 (Threshold) level in accordance with the official Levels of Competence in Russian as a Foreign Language.[6] Both experts worked with all ten articles, which resulted in twenty adaptations in total. The adaptation methods used by the experts were logged to a file. Upon completion a report was prepared, where the methods were systematized. Two main method types were singled out: structural and lexical.

In this paper we focus on methods for lexical adaptation, so we will examine it in more detail than structural transformations. Our experts used the following methods of manual lexical adaptation:

1. Replacing shortened and stylistically marked words (e.g. *соцсеть* sotcset` 'social network' → *социальная сеть* sotcial`naia set` 'social network');

2. Replacing (relatively) rare words with more common ones (e.g. *свыше* svy`she 'over' → *более* bolee 'over'; *глава* glava 'head (of department)' → *руководитель* rukovoditel` 'head (of department)');

3. Replacing hypernyms with hyponyms when the hyponym is a higher frequency word and when the transformation does not change the meaning dramatically. For instance, *табачные изделия* tabachny`e izdeliia 'tobacco products' can be replaced with *сигареты* sigarety` 'cigarettes' in an article about a ban on smoking. The justification for this replacement is that in Russian, the lexical unit *табачные изделия* is much less commonly used than *сигареты*. Furthermore, it is likely that there is a word that sounds similar to *сигареты* in the learner's native language.

4. Replacing hyponyms with hypernyms when the hypernym is a higher frequency word and when the use of the hyponym is not critical. For example, the more common word *врач* vrach 'doctor' can be a good alternative for the less common *врач-терапевт* vrach-terapevt 'physician' in some contexts; the same is true for *рыба* ry`ba 'fish' instead of *щука* shchuka 'pike (the fish)' in a sentence like *Putin caught a large pike*: it may be more important for the news story that President Putin's summer recreational fishing trip was successful, rather than the particular kind of fish he caught.

5. Sometimes the author of a news article may use different words to refer to the same object or person to avoid word repetition; some of these references are easier for the language learner to understand than others. For instance, the same person can be referred to in an article as *врач* vrach 'doctor' and as *собеседница агентства* sobesednitca agentstva 'the agency's interlocutor'. The frequent word *врач*, understandably enough, is more suitable for an adaptation than the rare *собеседница*. For this reason, the adapter of the text may choose to replace the latter word with the former or with a personal pronoun. However, automating this particular 'adaptational' kind of anaphoric resolution seems a very challenging task in its own right, and would require a powerful anaphora resolution module.

6. One of characteristic features of the Russian language is the use of affectionate diminutive suffixes that can be added to virtually any non-abstract noun. These morphemes often pose a challenge to learners of Russian. However, the suffixes do not form new words, but rather modify the original meanings [2, 13]. Therefore, for the purpose of adaptation it makes sense to replace the words having these suffixes with their base non-suffixal forms. For instance,magazinchik 'shop (affectionate diminutive)' → *магазин* magazin 'shop'; *машинка* mashinka 'car (affectionate diminutive)' → *машина* mashina 'car'.

Ideally, a fully automated system for text adaptation should accept an article as input, analyze and process it to produce a simplified version of the original. Any changes made to the original should concern form, not content. That is, the resulting text should contain roughly the same information as the source article, merely expressed in other words. However, the more we simplify the language, the more it might affect the meaning. This is somewhat similar to type I and type II errors in mathematical statistics, where the reduction in one error leads to an increase in the other. Similar to significance level, we chose the B1 (Threshold) level as the target adaptation level and try to minimize semantic distortion.

Any natural language processing algorithm has to be tested for precision and recall. One option is to manually mark up a test corpus and use it to verify the algorithm. In this case, it would be necessary to mark all complicators present in the text to ensure completeness. Creating such markup seems an extremely difficult task; what is worse, there is more than one way it can be done, as text adaptation is highly subjective in nature.

Verifying accuracy seems no less challenging a task for the same reason, i.e. subjectivity. We would first need to come up with clear criteria for distinguishing between adaptationally 'successful' and 'unsuccessful' text transformations. While this

5 http://www.ria.ru

6 http://en.russia.edu.ru/russian/levels

could probably be accomplished somehow, we have set ourselves a different task: building a semi-automatic system that requires human participation in the adaptation process.

IV. Wordnet-based Lexical Adaptation Algorithm

In many cases, using only lexical adaptation methods can significantly improve the readability of the text. Furthermore, such methods are relatively easier to automate in comparison with structural adaptation. Let us consider a 'difficult' word to be replaced, *w*. We have formulated the task as compiling a list of words to replace w in the text, each of which has its own weight $\mathbf{R}=\{ r_1, r_2 \dots r_{Sw}\}$. The weight of *w*, which is also added to the list, is r_0.

$$\mathbf{R}=\{ r_0, r_1, r_2 \dots r_{Sw}\} \quad (1)$$

The number of word substitutes S_w depends on *w* itself. Weights r_i should reflect both ease and semantic proximity. Some of them can be extracted from WordNet, but there is no resource like WordNet [5] in Russian language. Thus we get corresponding information from separated sources:

1. Whether the word is included in the B1 (our target level) lexical minimum [15] – r_{i1};
2. Word frequency in Russian in general and in texts of the selected genre – r_{i2};
3. Whether the word is present in the dictionary of synonyms. The dictionary contains over 300,000 words and expressions and relies on the ASIS word database [23] and the AOT morphological dictionaries [1] – r_{i3};
4. Whether the word is a hypernym or a hyponym [3] of *w* – r_{i4};
5. Contextual proximity (being used in similar contexts) of the substitute under consideration and *w* – r_{i5}.

We are to determine the weights of the word according to each of these factors, and then calculate the overall weight using the following formula:

$$r_i = r_{i1} * r_{i2} * (r_{i3} + r_{i4}) * r_{i5} \quad (2)$$

The weights of the lexical minimum dictionary r_{i1} and dictionaries r_{i3} and r_{i4} are binary and are equal to 1 if the word is on the list and 0 otherwise. For r_{i3} calculation, the list contains all synonyms of *w* from the dictionary. The list of Russian hypernyms and hyponyms for r_{i4} is compiled in the same way as described by Lukanin et al. [4, 5]. Weight r_{i2} reflects word frequency in Russian in general, and is determined using data from the Russian National Corpus of over 300 million words. To calculate the weight of *w* itself (r_0) we choose $r_{05} = \max(r_{i5})$; $i=1 \dots S_w$.

The overall weights are ranked in a descending order so that the first word on the list is the one having the greatest weight. We consider it to be the best substitute candidate. Words with zero weight are discarded. Thus, the suggested substitute list contain only words that are included in the lexical minimum and are in the synonyms and/or hypernyms/hyponyms dictionary. The word having the greatest weight replaces *w*. According to this rule, the choice between the best word substitute and *w* is made only based on use frequency.

Lexical substitution often necessitates morphological alterations to the dependent words in synthetic languages like Russian. For example, if we are to replace the rarer word *автомобиль* avtomobil` 'automobile' with the more frequent *машина* mashina 'car', we will have to take into account the fact that the former is masculine, while the latter is feminine. If the original word was used with an attribute, e.g. *дорогой* dorogoi` 'expensive', we would have to change the form of the attribute, too, from masculine into feminine (→ *дорогая* dorogaia 'expensive'). Word stemming and morphological processing is performed using the open application Pymorphy[7] which is based on OpenCorpora [24].

V. Contextual Proximity of Words

Using a large collection of texts of the same genre would allow us to investigate contextual proximity of words and word groups. These data can be used in several ways. One application is measuring r_{i5}, i.e. ranking words from the dictionary of synonyms according to their relevance to the context. Contextual proximity data could also help in the following tasks:

1. Morphological paradigm evaluation;
2. Ranking search query extensions;
3. Evaluating thematic similarity between the text and the search query;
4. Looking for new synonyms which are not in the dictionary.

Investigating contextual proximity of words, we assume that words having similar meanings can be found in similar contexts. To verify this assumption, we carry out empirical research using texts from the international news website Epochtimes[8]. The size of collection *D* is 78,000 articles, most of which are news stories.

The word we analyze is *w*. We choose the size of the context we are interested in and designate it *m*. The size of the n-gram for analysis, then, is 2m+1=n.

I	drove	to	work	this	morning.
	-2	-1	0	1	2

Figure 1. An example of a context for *work*, m=2, n=5.

Contextual proximity can be determined by comparing the context vectors of different words using Vector Space Model. We hope that it will be a measure of their semantic similarity.

To compare two words we need to get a subset of use frequencies of other words in the context of our words w_1 and w_2. It is a vector of frequencies from the context of a given width – *n*. These are to be compared and then ranked.

7 https://github.com/kmike/pymorphy2
8 http://www.epochtimes.ru/

$$\mathbf{x}_1=NC/N1;\ \mathbf{x}_2=NC/N2 \quad (3)$$

where N1 is the context of the first word, N2 is the context of the second word, NC is the frequencies vector of words which are used in the contexts of both w_1 and w_2. We call it the overlapping of the contexts of both words.

There are numerous ways of calculating the distance between the resulting vectors in Vector Space Model. We compared the overlapping of the contexts by calculating the following:

1. Euclidian distance;

$$L_{EU}=((\mathbf{x}_1-\mathbf{x}_2)\,(\mathbf{x}_1-\mathbf{x}_2)^T)^{1/2} \quad (4)$$

2. Cosine distance [22].

$$L_{COS}=1-(\mathbf{x}_1*\mathbf{x}_2)/((\mathbf{x}_1*\mathbf{x}_1^T)(\mathbf{x}_2*\mathbf{x}_2^T))^{1/2} \quad (5)$$

Such coefficients as Kullback–Laibler [18] and Jensen-Shannon divergence [9] are not suitable for use in our model, so we did not study them empirically.

Predictably, we get different results depending on the context width. In a narrow context, where m=2, there are many idioms and collocations; also, lexical data are limited. The wider the context, the more lexical data can be gathered. At the same time, the portion of general context-independent vocabulary increases. The maximum context width in our research was m=20. The product of the wide context window and a linear function penalizing remoteness allows us to flexibly filter out values that are not significant for the context.

We have removed stop words from the text, but even without them, there are still many general words in the broad context, viz. *который*, *свой*, *этот* and *один*. These words and the like are fairly common and do not seem to be indicative of any particular context. In order to reduce the significance of such words in the vector, i.e. to deprioritize them while verifying synonyms, it is important to normalize the frequencies. We use Z-scores $f'=(f-\mu)/\sigma$ and TF/IDF $f'=f/\ln(N)$ with N being the number of vectors containing the context and D – the total number of vectors.

Thus, in the algorithm, an n-dimensional hypercube contains frequencies normalized with the Z-score. The context window is selected with width m=20 words with a linear function penalizing remoteness. The frequency vectors are compared using the cosine distance and reflect contextual proximity. For comparison with w, all words having non-zero ranks R_1-R_4 are selected, then the ordered distances are converted into ranks.

For example, the word *правительство* pravitel'stvo 'government' has several synonyms in the dictionary: власть vlast' *'authority'*, администрация *administraciya 'administration'*, аппарат *apparat 'apparat'*, центр *center 'center'*. To estimate contextual proximity of the word *правительство*, we calculate the context frequencies vector in the original sentence for each element on the list of synonyms found. Here each word is normally used only once, so the vector consists of pairs of values (word number in the dictionary and its frequency) as shown in the Table 1.

TABLE 1. CONTEXT VECTOR IN THE ORIGINAL SENTENCE: WORD NUMBER IN THE DICTIONARY AND FREQUENCY.

Word Number in the Dictionary	Word Frequency
1	0
2	1/n
...	0
43	1/n
...	…
56	1/n
...	0
195	1/n
N	0

The same vectors of context frequencies are built for every synonym, but this time across the entire collection of documents. Synonyms having smallest distances to the vector in the original sentence have a higher ranking that corresponds to weight r_{i5}.

TABLE 2. WEIGHTS OF SYNONYMS OF THE WORD ПРАВИТЕЛЬСТВО PRAVITEL'STVO 'GOVERNMENT'.

Synonym	r_{i5}
власть vlast' 'authority'	4
администрация administraciya 'administration'	3
аппарат apparat 'apparat'	2
центр center 'center'	1

A disadvantage of the algorithm is a high data dimension N which results in great computational complexity. For this reason, we cannot investigate word groups in contexts. Besides, the algorithm produces a deterministic result that can only be analyzed empirically. There seem to be no other, more rigid methods of algorithm evaluation in this case.

VI. Distributional Semantic Model of Contextual Proximity

Another method for context analysis can be implemented with Distributional Semantic Model (DSM) [14, 16]. We use Latent Dirichlet Allocation [6] which is a generative model that uses latent groups to explain results of observations – data similarity in particular. For instance, if observations are words in documents, one can posit that each document is a combination of a small number of topics and that each word in the document is connected with one of the topics. Latent Dirichlet Allocation (LDA) is one of topic-modeling methods and was first introduced by its authors as a graphical model for topic detection.

The model is based on the assumption that words in a document are independent of one another (bag of words [12]) and of their order in the text. Similarly, documents in a Corpus are independent of one another and unordered. Distribution of words **w** is determined by the set of topics **z**. Each topic z_n has its own word distribution $P(w_i/z_k)$.

By training the model, we form the statistical portrait of its author. A person writing a text has a set of topics in their mind, and each document has a certain distribution of these topics. The author first selects the topic to write on; within this

topic, there is a distribution of words that may occur in any document that contains this topic. The next word in the text is generated within the distribution. Then the same procedure is repeated. On each iteration, the author either selects a new topic or continues to use the previous one, and generates the next word within the active topic [6].

After training the model on a collection of texts, we get an estimate of two discrete distribution functions. The following is distribution of probabilities of words **w** in topics **z**:

$$P(w_i / z_k);\ i=1...|\mathbf{w}|,\ k=1...|\mathbf{z}| \quad (6)$$

Distribution of probabilities of topics **z** in documents **d**:

$$P(z_k / d_n);\ n=1...|\mathbf{d}|,\ k=1...|\mathbf{z}| \quad (7)$$

To find out whether it is possible to use the LDA model in our task of synonym ranking, we must test the following hypothesis: if words w_A and w_B are synonyms of word w_0 and

$$L[P(z_k /w_0), P(z_k /w_A)] > L[P(z_k /w_0), P(z_k /w_B)];\ k=1...|\mathbf{z}| \quad (8)$$

is true, then word w_A is a better substitute for w_0 than w_B. $L[P(z_k /w_0), P(z_k /w_A)]$ and $L[P(z_k /w_0), P(z_k /w_B)]$ are distances from the distribution vector of word w_0 based on all topics to the vectors of words w_A and w_B respectively.

In order to build the LDA model for calculating $L[P(z_k /w_0), P(z_k /w_A)]$ and $L[P(z_k /w_0), P(z_k /w_B)]$, we will use the same collection of news articles as with the previous solution. The size of collection D is 78,000 articles, most of which are news stories.

Preprocessing includes the following steps:

- tokenize the text;
- lemmatize the tokens;
- index the words using the dictionary of lemmas;
- filter out the words that are too frequent (stop words) or too rare (used only once).

The processed text is then directed to LDA algorithm with a given number of topics K. At present, there are several methods for building LDA models, that is, methods of searching for parameters of all distribution functions in the model. All of the methods are iteration-based and are similar in structure to the Expectation Maximization (EM) algorithm. They are:

- Online Variational Bayes algorithm [7];
- Gibbs Sampling [20];
- Expectation Propagation [21].

We use the Online Variational Bayes algorithm as it is the most precise one (Hoffman et al., 2010). It is realized in the Gensim[9] toolkit.

Based on word probability distribution for topics $P(w_i / z_k);\ i=1...|\mathbf{w}|,\ k=1...|\mathbf{z}|$, we build a vector of probabilities that the word corresponds to each topic. The length of such vector is equal to the number of topics $|\mathbf{z}|$.

$$P(z_k /w_i);\ k=1...|\mathbf{z}| \quad (9)$$

We rank the cloud of 'similar' words from the dictionary of synonyms according to the context distance between these words and the original one. That is, we create a weighted cloud. The context distance is calculated using four different methods. Apart from the Euclidian (4) and cosine (5) distances, we use the Kullback-Leibler divergence [18]:

$$KL(P(z_k / w_A), P(z_k / w_B)) = \Sigma_{k=1}^{|\mathbf{z}|} P(z_k / w_A) log(P(z_k / w_A) / P(z_k / w_B)); \quad (10)$$

and Jensen-Shannon divergence [9]:

$$JS(P(z_k / w_A), P(z_k / w_B)) = 0.5*(KL(P(z_k / w_A), P()) + KL(P(z_k / w_B), P())); \quad (11)$$
$$P()=0.5(P(z_k / w_A), P(z_k / w_B))$$

Since in this case we compare two functions of probabilities distribution, the divergences (10) and (11) can be interpreted well. The calculation results for the synonyms set for the word *правительство* pravitel'stvo 'government' is presented in Table 3.

TABLE 3. CONTEXT DISTANCES BETWEEN THE WORD ПРАВИТЕЛЬСТВО PRAVITEL'STVO 'GOVERNMENT' AND ITS SYNONYMS.

Synonym	Euclid x0.01	Cos	KL x0.01	JS x0.01
власть vlast 'authority'	1. 5493	0. 41598	1. 73546	0. 8771
администрация administraciya 'administration'	1. 2175	0. 67216	1. 96434	1. 1365
центр center 'center'	1. 7214	0.82965	2. 52262	2. 1914
аппарат apparat 'apparat'	1. 9592	0.98475	1. 27487	1. 7923

As can be seen from the results, the word власть vlast *'authority'* has the lowest distance values for almost all metrics used. This word is considered the best substitute for *правительство* pravitel'stvo 'government', according to our experts on Russian as a foreign language. Similar results were obtained for synonyms of other words such as *крошечный* kroshechny`i` 'tiny', *свыше* svy`she 'more than' and others. Therefore, we can conclude that our hypothesis was true for the Euclidian and cosine distances and for the Jensen-Shannon divergence. This means that we can create an algorithm for synonym ranking using these metrics and the LDA model.

When we change the number of topics in the model K from 100 to 500 the context distances between objects do not change significantly; neither does the synonyms ranking. In the upper (main) part of the list there are no changes at all.

In the Table 4 we show all weights r_{ij} for the word *правительство* pravitel'stvo 'government' and r_i as a result.

TABLE 4. WEIGHTS OF SUBSTITUTIONS FOR WORD ПРАВИТЕЛЬСТВО PRAVITEL'STVO 'GOVERNMENT'

Synonym	r_{i1}	r_{i2}	r_{i3}	r_{i4}	r_{i5} x0.01 (JS div.)	r_i
власть vlast 'authority'	1	4 (20694)	1	0	4 (0,8771)	16
центр center 'center'	1	3 (7589)	1	0	2 (2,1914)	6
аппарат apparat 'apparat'	1	2 (4600)	1	0	1 (1,7923)	2
администрация	1	1 (1838)	1	0	3 (1,1365)	1

9 http://radimrehurek.com/gensim/index.html

administraciya 'administration'						

VII. Conclusion

This paper proposes a technology for text retrieval from Corpus an algorithm for lexical adaptation of news texts that can be used as materials for learning/teaching Russian as a foreign language. The algorithm relies on the wordnet-based and corpus-based contextual proximity.

We considered two methods of calculating contextual proximity. The first relies on the vector of normalized frequencies of word use in the nearest context. The second is based on the LDA model and on the vector of topic-based word frequencies distribution. We have found that in both cases contextual proximity yields useful results for synonym ranking.

The drawback of the first method that is based on Vector Space Model of texts is a high data dimension. The vector of frequencies length is equal to the size of the dictionary, which is around 20,000 words in our case. The second method, based on LDA, solves the problem of the high dimension and makes it possible to calculate and interpret contextual proximity efficiently.

With LDA it is feasible to rank clouds not only of similar words, but also of word groups for the given word. The LDA model also allows us to generate document descriptions and find clouds of similar documents.

Acknowledgment

This study comprises research findings from the «Adaptation of texts from the Russian National Corpus for the electronic textbook «Russian language as a foreign one» carried out within The National Research University Higher School of Economics' Academic Fund Program in 2013, grant No 13-05-0031.

References

[1] A. V. Sockirco. 2004. "Morphological Modules on the Site www.aot.ru" In Computational Linguistics and Intelligent Technology: Proceeding of the International Conference «Dialog–2004».

[2] A. A. Shakhmatov. 2001. Syntax of Russian Language. Editorial URSS, Moscow, Russia.

[3] A. Budanitsky, G. Hirst. 2006. "Evaluating wordnetbased measures of lexical semantic relatedness," in Computational Linguistics, 32 (1):13-47.

[4] A. Lukanin, K. Sabirova, A. Panchenko. 2013. "Extraction of Russian Hypernyms and Co-Hyponyms with Lexico-Syntactic Patterns". In RuSSIR Young Scientist Conference (RuSSIR YSC 2013) Proceedings.

[5] C. Fellbaum. 1998. WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, MA.

[6] D. M. Blei, A. Y. Ng, Michael I. Jordan. 2003. "Latent Dirichlet Allocation, " In Journal of Machine Learning Research, pages 993–1022.

[7] D. M. Blei, Matthew D. Hoffman. 2010. "Online Learning for Latent Dirichlet Allocation. Advances" in Neural Information Processing Systems, pages 856-864

[8] F. Horacio, Luciana Ferrer, and Harry Bratt. 2012. Adaptive and Discriminative Modeling for Improved Mispronunciation Detection.

[9] F. Bent, F. Topsoe. 2004. "Jensen-Shannon divergence and Hilbert space embedding, " in Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on page 31. IEEE.

[10] J. Burstein, J. Shore, J. Sabatini, Yong-Won Lee, M. Ventura. 2007. "The Automated Text Adaptation Tool," in NAACL-Demonstrations '07 Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pages 3-4.

[11] J. M. Bradley, Wilbur S. Ames & Judy N. Mitchell. 1984. "The Effects of Text Adaptation on Rated Appeal and Difficulty," in Reading Psychology, 5 (3):185-191

[12] H. M Wallach. 2006. "Topic modeling: beyond bag-of-words," in Proceedings of the 23rd international conference on Machine learning, pages 977–984.

[13] K. S. Aksakov. 2011. A Russian Grammar: Name. Librokom, Moscow, Russia.

[14] M. Baroni, and Alessandro Lenci. "Distributional memory: A general framework for corpus-based semantics," in Computational Linguistics 36(4) 2010: 673-721.

[15] N. P. Andriushina. 2011. Russian as a Foreign Language Lexical Minimum. First Certificate Level. General Proficiency. Zlatoust, Saint-Petersburg, Russia.

[16] P. Turney. 2006. "Similarity of semantic relations, " Computational Linguistics, 32(3):379–416

[17] S. Banville. 2005. "Creating ESL/EFL Lessons Based on News and Current Events, " In The Internet TESL Journal, volume XI, No. 9.

[18] S. Kullback, R. A. Leibler. 1951. "On information and sufficiency, " in The Annals of Mathematical Statistics, 22(1): 79–86.

[19] T. A. van Dijk. 1988. "News Analysis: Case Studies of International News" in the Press. Lawrence Erlbaum Associates, Publishers. Hillsdale, New Jersey, Hove and London.

[20] T. L. Griffiths, Mark Steyvers. 2004. "Finding scientific topics, " Proceedings of the National academy of Sciences of the United States of America, 101(Suppl 1): 5228–5235.

[21] T. Minka, J. Lafferty. 2002. "Expectation-propagation for the generative aspect model," in Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pages 352–359. Morgan Kaufmann Publishers Inc.

[22] T. Korenius, J. Laurikkala, & Martti Juhola. (2007). "On principal component analysis, cosine and Euclidean measures" in information retrieval. Information Sciences, 177(22): 4893–4905.

[23] V. N. Trishin. 2010. "Electronic dictionary and handbook of Russian synonyms in the ASIS system," In Vladimir Dal at happy home on the Presny str. pages. 158–165. Moscow, Russia: Academia.

[24] V. Bocharov, M. Stepanova, N. Ostapuk, S. Bichineva, D. Granovsky. 2011. "Quality assurance tools in the OpenCorpora project," in *Computational Linguistics and Intelligent Technology: Proceeding of the International Conference «Dialog–2011»,* pages 10–17