Bull. Nov. Comp. Center, Comp. Science, 41 (2017), 13-25
@ 2017 NCC Publisher

A combined approach to part-of-speech homonymy resolution

E.P. Bruches, T.V. Batura

Abstract. The Russian language has an inflective structure and does not have a strict word order, which generates processing problems such as part-of-speech homonymy. The paper addresses this issue. The existing approaches to resolving the morphological homonymy problem can be divided into the following groups: rule-based approaches, statistical approaches, machine learning approaches, and combined methods. In the paper, we showed that each approach has its advantages and disadvantages; however, we can achieve a much higher precision of the algorithm by combining several approaches. The combined method based on neural networks gives better results than others (98% precision obtained). We used the following features: normalizing substitutions, grammatical and syntactic characteristics, vector representation of the word, and word forms. All the experiments were performed on the part of the National Corpus of the Russian Language with homonymy resolution. The analysis of the corpus revealed that the most frequent types of homonymy occurred between function words: a particle vs an interjection (14%), and a preposition vs an interjection (13.2%).

Keywords: text processing, part-of-speech homonymy, combined approach, machine learning, homonymy resolution.

1. Introduction

At present, computer linguistics and automatic text processing in particular is becoming increasingly important, which is attributed to the necessity of digesting the huge amount of information contained in the texts written in a natural language. Machine processing simplifies the process considerably. The volume of accumulated text information makes it impossible to grasp it without automatic text processing.

Working with Russian texts is complicated because the Russian language is inflective and does not have a strict word order, which often results in homonymy. The authors make an attempt to deal with this issue.

We consider the morphologic (grammatical) case of homonymy. Grammatical homonyms are words from different parts of speech, coinciding with one or several forms. For instance, '*ели*' (a form of the verb '*есть*') vs. '*ели*' (a form of the noun '*ель*'). Words that came from one part of speech to another are also related to the group: '*точно*' (adverb) vs. '*точно*' (comparative conjunction), '*прямо*' (adverb) vs. '*прямо*' (amplifying particle). Nouns formed from adjectives and participles fall into this category as well (for instance, '*дежурный врач*' vs. '*дежурный*').

The existing approaches to resolving the morphological homonymy problem can be divided into the following groups:

1. Rule-based approaches;
2. Statistical approaches;
3. Machine learning approaches;
4. Combined methods.

Rule-based approaches. A well-known algorithm to determine the part of speech of English words using manual rules is a tagger, which was described in paper [1]. For example, one of the transformation rules 'TO IN NEXT-TAG AT' says that if a word is assigned by the tag TO, and the following word is assigned by the tag AT, the tag TO should be changed to the tag IN, where TO is an infinitive particle '*to*', and AT and IN are prepositions. The explanation is that before a nominal group a preposition is more likely than the infinitive particle '*to*'. In [2], rules of this kind were applied to the Russian language and the precision of homonymy resolution was 89.1%.

The linguistic processor ETAP, operating with the Russian language and having the components of homonymy resolution, is described in [3]. It uses the following method. At the first stage of analysis, a set of all possible syntactic links between the words of the phrase being analyzed is built. Then, from this set, unacceptable links are deleted until the desired syntactic structure of the phrase is left.

The benefit of this method is its quite good ability to analyze texts of various fields, genres and styles. To set the rules, no labeled data is required. It is possible to make rules for special cases, which are exceptions. A drawback of the rule-based methods is high labor intensity, since establishing a system of rules requires qualified linguists. Also, it is difficult to ensure the rules consistency.

Statistical approaches. Statistical methods are based on the use of statistical data, which are calculated in advance on a marked text corpus. The algorithm based on context dictionaries is described in article [4]. It was first proposed to use normalizing substitution as a context element. The algorithm is designed to resolve morphological homonymy in words coinciding only in a few grammatical forms. For instance, '*три*' and '*трём*' are the forms of a numeral and a verb; '*стекло*', '*стих*', '*стали*' are the forms of a noun and a verb, etc.

The main goal of the algorithm is to determine the correct lemma of a given word form, i.e. make a choice between several production rules of word form transformation into potential lemmas. Word endings are the main way to express grammatical meaning in inflective languages (one of which is the Russian language); therefore, the authors have put forward a hypothesis of using rules as context elements, which allow transforming word forms into lemmas. The authors also supposed that the influence of context elements on the selection of a correct lemma is stronger or weaker depending on the location of a given word. Consequently, the algorithm takes into account the priority of neighbors' influence in numerical form based on a simple probabilistic model.

The main data structure used in homonym resolution is a context dictionary organized in the following way. The base unit of the dictionary is a triplet <*homonym, context element, lemma*>, which is an elementary and random event: when a given homonym and a given context element appear in a text, a given lemma will be chosen as the meaning of the homonym. Each elementary event is assigned a certain probability of its occurrence based on automatic processing of annotated corpora with resolved homonymy.

Examples of normalizing substitutions for ordinary words are listed in Table 1. Examples of normalizing substitutions for words with part-of-speech homonymy are listed in Table 2. Examples of context dictionary records are listed in Table 3. For each word form, one or several lemmas are defined together with sets of the main grammatical characteristics (part of speech, gender, number, case, person, etc.). To reduce the volume of the dictionary, the elements of the triplet are represented as a normalizing substitution consisting of a three-letter end of the word form, followed in brackets by how many letters and which ones need to be replaced to obtain the lemma.

Table 1. Examples of normalizing substitutions for unambiguous word forms

Substitution	Unambiguous word form
мал(1ть)	думал{думать=V,несов=прош,ед,изъяв,муж}
огу(1а)	дорогу{дорога=S,жен,неод=вин,ед}
ной(4я)	мной{я=S,ед,од=(твор,жен|твор,муж)}

Table 2. Examples of normalizing substitutions for homonymous word forms

Substitution	Homonymous word forms
чал(0о|2инать)	начал{начинать=V=прош,ед,изъяв,муж,сов|начало=S, сред,неод=род,мн}
ерь(1ить|1ять)	поверь{поверить=V,сов=ед,пов,2-л|поверять=V=ед,пов,2-л,сов}
рка(0|1|1ий)	марка{марк=S,муж,од=(род,ед|вин,ед)|марка=S,жен, неод=им,ед|маркий=A=ед,кр,жен}

Table 3. Examples of context dictionary records

Substitution	Context words	Probability
ала(1о|3инать)	[р]+2 1о	0.67
ала(1о|3инать)	[р]+2 3инать	0.33

The process of homonym resolution is as follows. First, normalizing substitutions are built for each homonym and its closest neighbors based on the results of morphological analysis. Then, a lemma and the probability of its generation by a given context element are chosen from the context dictionary for each pair <*homonym, context element*>. Further, a sum of probabilities is figured out for each lemma and multiplied by the 'power' of the context element. The lemma with the maximum weighted sum of probabilities is considered as the meaning of homonymy in the given context.

The precision of the described algorithm is 94-95%. The main advantage of the algorithm is a smaller dictionary size: rather than considering the context as word forms, the authors have suggested considering it in the form of normalizing substitutions, i.e. rules transforming word forms into lemmas. A disadvantage of the algorithm described is its inability to resolve the homonymy of the word forms whose lemmas coincide, for instance, '*когда*' as a conjunction and '*когда*' as an adverb.

Machine learning methods. The main goal of machine learning methods is to discover regularities in existing empirical data. The general statement of the problem is formulated in the following way. There is a set of objects, a set of possible answers, a certain unknown dependency between the answers and the objects, and a finite collection of precedents as pairs "object, answer" called a learning selection. On the basis of these data the dependency needs to be reconstructed, that is, an algorithm should be devised capable of giving a reasonably accurate answer to any object. In the case of a problem of part-of-speech homonymy resolution, the object is represented by certain characteristics in which regularities are discovered, and a specific part of speech is the answer.

As shown in [5] and [6], the problem of part-of-speech disambiguation can be successfully dealt with by decision trees. The author of [7] developed a system for the partial markup of texts in the Russian language based on the combination of a neural network and a priori information contained in a dictionary.

The authors of [8] described an algorithm for the English and Portuguese languages backed by a convolutional neural network. They offered a deep neural network architecture capable of considering a word on two levels: semantic and morphemic. The neural network suggested uses a convolution layer for an efficient extraction of characteristics from the words of arbitrary length. The algorithm does not require any manually extracted characteristics. The precision of the algorithm for the English language for the PennTreeBank corpus was 97.32%, and for the Portuguese language, the precision was 97.47% for the Mac-Morpho corpus. Moreover, the authors claim that the described design of the neural network can be used to resolve homonymy in any language.

The analysis of machine learning methods indicates that their application for morphological text analysis is justified. Some models do not even require the manual search and extraction of relevant characteristics because regularities are discovered on the level of separate characters. However, as in statistical methods, the results of the model using machine-learning algorithms depend heavily on the training corpus. Another disadvantage is the impossibility of interpreting the results of the model (like in the case of the ANN algorithm).

Combined methods are a combination of two or more methods in one way or another. A combined method was described in [9] for the Czech language. The authors present a hybrid system combining the merits of manually written rules with the statistical approach. The methods are applied sequentially: first, homonymy is partially resolved by means of rules, and then the results are processed by an algorithm based on the hidden Markov's models. Experiments have shown that, on average, the algorithm resolves the morphological homonymy in the Czech language with the precision of 95.38%. In the above results, it is stated that the algorithm based on the hidden Markov's models copes with the specified problem with the precision of 95.16%, while the rule-based approach gives 53.36%. Therefore, according to the authors, combining several approaches is justified.

Another work describing the combined approach is [10]. Its authors introduce the results of combining the statistical and deterministic methods on the basis of the Basque language. This is an agglutinative language, in other words, each word formant (prefix or suffix) has only one grammatical meaning. Considering these language features, the authors suggested the following algorithm. At the first stage, each word is assigned all its possible grammatical meanings with the help of a morphological analyzer. Then, the morphological ambiguity is reduced by applying rules based on the grammar of constraints. At the next stage, an algorithm based on the hidden Markov's models is used to pick one of the remaining parts of speech. The algorithm described is 97% successful in homonym resolution.

To sum up, it is evident that combining several algorithms generally allows increasing the overall precision of an algorithm.

2. Data preparation

To obtain all morphological information (grammatical characteristics of word forms, normal form of words, paradigms), a dictionary was used. It was formed on the basis of open resources, namely, OpenCorpora (http://opencorpora.org/) and Wiktionary (https://ru.wiktionary.org/). The following grammatical features were considered: part of speech, grammatical gender, number, case, view, transitivity, person, time, mood, and voice.

The system of the rules and statistical approach we implemented in Java. All neural networks described in this paper are implemented in Python version 2.7 with the Keras library [11].

We used a corpus with resolved homonymy for statistical methods and machine learning methods. The whole corpus was represented as a set of trigrams: a word for which the part of speech had to be identified, and words before and after it. If the word for which the part of speech had to be determined took the first or the last position in the sentence, the missing element of the context had a special mark. Punctuations marks were not considered as context elements.

3. Rule-based approach

Two types of the rules were constructed to resolve part-of-speech disambiguation: some of them are based on the analysis of the local context; others represent unambiguous definition of a part of speech on the basis of frequency characteristics. They are used to determine the main parts of speech and function words, respectively.

The first type of rules consists of several lists, each including rules to distinguish two parts of speech. For example, if a word is a noun and a verb, the program will choose a list with rules where one of the words must have a corresponding part of a speech conflict. The rules of this type are divided into two groups, positive and negative. In the positive rules, the existence of grammatical characteristics in the context unambiguously determines the part of speech of a word form. For example, for the word ‘*вести*’ we have the rule “*If there is a preposition before a word, then this word is a noun*.”

In the negative rules, the absence of the aforesaid grammatical characteristics in the vicinity of a word defines its part of speech. For example, for the word ‘*том*’ the following rule is applied, “*If before a word there is any part of speech except for a preposition, then the word is a noun*’.

Let W be a set of Russian words, and let there be sets corresponding to the parts of speech: Adj for adjectives, V for verbs, N for nouns, Adv for adverbs, etc. Let us denote a predicate true for a pair of consecutive words as $P(x, y)$, and predicates for each part of the speech as $Adj(w)$, $V(w)$, $N(w)$, $Adv(w)$, etc. Then some of the rules can be written in the following form:

$$\forall w \in W, x \in Adj: P(x, w) \to N(w) \vee Adj(w)$$
$$\forall w \in W, x \in V: P(x, w) \vee P(w, x) \to N(w) \vee Adv(w)$$
$$\forall w \in W, x \in Adv: P(x, w) \vee P(w, x) \to V(w).$$

The rule-based approach was not used in one-word sentences. The algorithm took into account the context of the word within a single sentence.

4. Statistical approach

We implemented the statistical approach similar to the one described in [4] with the difference that the part of speech rather than the lemma was defined for a word form. Besides, we redefined the definition of a normalizing substitution: we considered the combinations of three-letter endings of a word and its nominal form (if a word or its nominal form was less than three letters long, the entire word was considered). In this way, for example, for the word '*думал*', a normalizing substitution is '*мал-ает*', and for the word '*стекло*' a normalizing substitution is '*кло-кло/ает*'.

All trigrams were combined into elements of the following type:

$< SC, SW, POSW >$, where

SC is a normalizing substitution for the previous or following word;

SW is a normalizing substitution of a word for which its part of speech needs to be determined;

POSW is the correct part of speech of the given word.

Then we calculated the probabilities of the following condition for each word. If the substitution *SW* exists in the context of the substitution *SC*, the word belongs to the part of speech *POSW*. We select the part of speech with the maximum probability.

The advantage of this approach is that it enables us to determine, provided that there appropriate substitutions in the context, the part of speech of a word not found in the dictionary. In addition, the method proposed can reduce the size of the dictionary significantly. The drawback of the approach is the impossibility of analysis with a complete absence of context elements in precompiled statistics.

5. Machine learning approach

Determining the part of speech is a classification problem, i.e. the division of objects into a known number of classes. Artificial neural networks were chosen as the machine learning algorithm. Each neural network has one input layer, and hidden and output layers. The number of neurons of the input and hidden layers depends on output features. The number of neurons of the output layer corresponds to the overall number of parts of speech, that is, it equals to 17. As an activation function for deep neural networks on input and hidden layers, we used a rectified linear unit (ReLU):

$$\sigma(x) = \max(0, x).$$

The output layer uses the SoftMax activation function to obtain the probability distribution of belonging of the input example to one of the classes. In our case, classes are parts of speech, and this function has the following form:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \text{ где } j = 1, \dots, K \ (K \text{ is the size of the output vector}).$$

We chose as an optimizer the stochastic gradient descent with the training step size of 0.01. The input and hidden layers in the neural networks are followed by a batch-normalization layer, where data are normalized before entering each layer (except for the input one), which makes it possible to speed up learning and use quite a high size of the training step [12]. We will briefly describe each of the four features used that affect the result.

5.1 Feature: normalizing substitutions

A special layer, randomly initializing predetermined-sized vectors of integers, each denoting the position of the corresponding element in the dictionary, was used as the input layer in this neural network. Therefore, the size of the input vector was three, and in each of its positions the index of the word substitution position in the dictionary was recorded (the dictionary size was 20810):

$[W_{t-1}, W_t, W_{t+1}]$, where

W_t is the word for which part of speech needs to be determined, and

W_{t-1} and W_{t+1} are the previous and next words, respectively.

The size of the vector obtained with this layer is 100.

The substitution dictionary was made from 400,000 randomly chosen trigrams. The unknown word was assigned an index equal to the size of the dictionary plus 1.

5.2 Feature: grammatical and syntactic characteristics

As characteristics, the grammatical characteristics of the word were chosen, namely: the part of speech, case, number, gender, as well as the features of whether the word form is a numeric complex, whether it begins with a capital letter, whether there are punctuation marks surrounding it and whether the word form is at the sentence boundary. The input vector has a size of 105 (35 symbols * 3 words) and is constructed in the following way. For every given word form, all of its grammatical and syntactical features are looked up in the dictionary, and "1" is set in the position of the corresponding grammeme.

5.3 Feature: vector representation of the word

The next feature whose influence on the solution of the problem we have considered is lexical vectors. The idea of using vectors of this kind originates from distributional semantics. Distributional semantics is an area of linguistics dealing with the calculation of the degree of semantic similarity between linguistic units based on their distribution in large volumes of linguistic data (text corpora). In distributional semantics, words are usually represented as vectors in the multidimensional space of their contexts. Semantic similarity is calculated as a cosine proximity between the vectors of two words and can take values in the interval [-1 ... 1] (in practice only values above 0 are common). A value of 0 means that these words are likely to have no similar contexts, and their values are not related to each other. The value of 1, on the contrary, indicates the complete identity of their contexts and, therefore, their close meaning. Machine learning methods allow finding such vectors with a predetermined dimension. One such technology is word2Vec described in [13]. It employs two algorithms: a continuous bag of words (CBOW) and Skip-Grams. It is believed that the first algorithm is faster, while the second algorithm is better suited for rare words. It was also shown that the vectors obtained with the help of this technology can accumulate not only semantic, but also grammatical information.

We used the vector representations of words that were preliminarily calculated on a set of texts from the NCRL (National Corpus of the Russian Language, http://ruscorpora.ru) using word2Vec [14]. The size of each word vector was 300; consequently, the size of the input vector was 900 (300*3 words). Vectors for homonymic word forms were averaged.

5.4. Feature: word forms

This neural network has the same architecture as that described in Section 4.1, except that the word unit is now a word form, and not a normalizing substitution. The dictionary was formed in the way described above, and its size was 104879 word forms. The index of an unfamiliar word was equal to the size of the dictionary plus 1.

6. Combination of approaches

To achieve higher precision, we combined the methods described above, except for the rule-based approach. Since SoftMax was chosen as an activation function on the output layer of each network and its result is a probability distribution, the final solution was taken as follows. For a selected part of speech, each of the algorithms summed up its corresponding probabilities multiplied by a certain coefficient selected on the basis of several experiments. As a result, the part of the speech with the highest result was selected. 5.0 was chosen as the coefficient for the results of the neural network based on grammatical features, 4.0 for the neural network based on lexical vectors, and 1.0 for other algorithms.

7. Results

All the experiments were performed on the part of the NCRL with homonymy resolution selected specifically for testing and not intersecting with the part of the corpus used for training. The size of the corpus used for testing made up of 100 000 word forms, with approximately 33 000 out of them having part-of-speech homonymy.

The analysis of the corpus revealed that the most frequent types of homonymy occurred between function words: a particle vs an interjection (14%), and a preposition vs an interjection (13.2%). A quantitative distribution of the most frequent types of homonymy is shown in Figure 1.

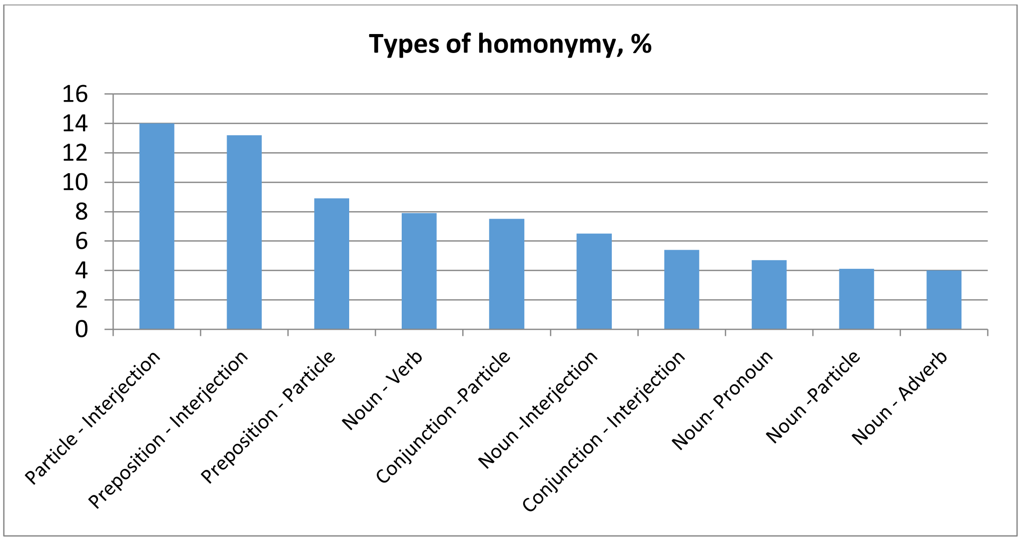

Figure 1. Quantitative distribution of homonymy types

Each algorithm was evaluated according to two characteristics, namely, defining the part of speech and homonymy resolution. They were calculated using to the following formulas.

$$WCD = WR/WT, \text{ where}$$

WR is the number of words with a correctly identified part of speech; and

WT is the total number of words.

$$HR = AWR/AWT, \text{ where}$$

AWR is the number of words having part-of-speech homonymy, with a correctly identified part of speech; and

AWT is the total number of words with part-of-speech homonymy.

The results of the experiments and their comparison with other morphological analyzers (namely, PyMorphy, FreeLing and TreeTagger, described in detail in [15]) are presented in Table 4. The given analyzers were tested on the National Corpus of the Russian Language.

Table 4. Precision of presented classification algorithms

Approach	WCD	HR
Rules	–	0.91
Statistics	0.95	0.93
NS (normalizing substitutions)	0.95	0.92
NS (grammatical and syntactic features)	0.95	0.91
NS (vector presentations)	0.83	0.78
NS (word forms)	0.92	0.91
Combined approach	**0.98**	**0.97**
PyMorphy	0.907	–
FreeLing	0.915	–
TreeTagger	0.952	–

It should be noted that rules were made up only for the part-of-speech homonymy resolution tasks; therefore, the corresponding approach was not used to evaluate part-of-speech tagging.

The results obtained by the statistical method also require explanations. As mentioned above, the drawback of this approach is that it is impossible to analyse a word form without statistics. In the experiment conducted, the statistics was only in 75826 out of 100000 text examples. Table 4 shows the results for them only.

Error analysis in combining approaches have shown that the worst result was obtained for particles and conjunctions (for example, words ‘*да*’, ‘*а*’ and so on). The frequency of such errors is 17%. Also, the algorithm has a problem with the distinction of conjunctions and adverbs, such as ‘*когда*’, ‘*как*’; the frequency of such errors is 15%. The error rate in homonymy resolution is 11% for pronouns and conjunctions in some word forms (for example, ‘*что*’, ‘*чем*’).

8. Conclusion

This paper deals with different approaches to homonymy resolution. The authors have shown that each approach has its advantages and disadvantages; however, combining several approaches can increase the precision of the algorithm significantly.

Moreover, the paper studies the influence of certain features on resolving the morphological homonymy problem. The algorithm is precise enough to be used for the tasks of the intellectual processing of texts.

Error frequency in distinguishing function words remains high. Solving this problem will be our priority in the future. We also plan to implement a system without a dictionary, in order to determine better morphological features for unknown words.

References

[1] Brill E. A simple rule-based part of speech tagger // Proc. of the Third Conference on Applied Natural Language Processing (*ANLC '92*), Stroudsburg, PA, USA. – 1992. – P. 152–155.

[2] Rysakov S.V. Resolving homonymy methods // System Administrator. – 2015. – No. 10 (155). – P. 92–95 (In Russian).

[3] Cinman L. L., Sizov V. L. Linguistic processor ETAP: descriptor matching and metaphor processing // Proc. of the International conference “Dialogue’2000”. Moscow: Nauka. 2000.: http://www.dialog-21.ru/digest/2000/articles/cinman/ (In Russian).

[4] Zelenkov Ju., Segalovich I., Titov V. Probabilistic model of morphological homonymy disambiguating based on normalizing substitutions and positions of neighboring words // Computational Linguistics and inTellectual Technologies. Moscow: Nauka. – 2005. – P. 188–197 (In Russian).

[5] Orphanos G., Kalles D., Papagelis T., Christidoulakis D. Decision tress and NLP: A case study in POS tagging//Proc. of ACAI 1999.: http://faculty.washington.edu/fxia/courses/LING572/decison_tree99.pdf

[6] Schmid H. Probabilistic part-of-speech tagging using decision trees // Proc. of the international conference on new methods in language processing. – 1994. – Vol. 12. – P. 44–49.

[7] Malanin G.P. Part-of-speech tagging using a heterogeneous neural network and a priori information // Youth Scientific and Technical Bulletin. – 2014. – No. 12.: http://docplayer.ru/42343992-Chasterechnaya-razmetka-slov-s-ispolzovaniem-geterogennoy-neyronnoy-seti-i-apriornoy-informacii.html (In Russian).

[8] Santos C., Zadrozny B. Learning character-level representations for part-of-speech tagging // Proc. of the 31st International Conference on Machine Learning, JMLR: W&CP. Beijing, China. – 2014. – Vol. 32. – P. 1818–1826.: http://proceedings.mlr.press/v32/santos14.pdf.

[9] Hajič J., Krbec P., Květoň P., Oliva K., Petkevič V. Serial combination of rules and statistics: a case study in Czech tagging // Proc. of the 39th Annual Meeting of the Association for Computational Liguistics (ACL 2001). Toulouse, France. – 2001. – P. 260–267.

[10] Ezeiza N., Alegria I., Arriola J. M., Urizar R. Combining Stochastic and rule-based methods for disambiguation in agglutinative languages // Proc. of the 17th International Conference on Computational Linguistics (COLING’98). Montreal, Canada. – 1998. – Vol. 1. – P. 380–384.

[11] Keras: The Python Deep Learning Library.: https://keras.io/.

[12] Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift // Proc. of the 32nd International Conference on Machine Learning, PMLR. – 2015. – Vol. 37. – P. 448–456

[13] Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space // Proc. of the First International Conference on Learning Representations (ICLR) . - 2013.: https://arxiv.org/pdf/1301.3781.pdf.

[14] Kutuzov A., Kuzmenko E. WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models // In: Ignatov D. et al. (eds.) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science. Springer, Cham. – 2016. – Vol. 661. – P. 155–161.

[15] Kuzmenko E. Morphological Analysis for Russian: Integration and Comparison of Taggers // In: Ignatov D. et al. (eds.) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science. Springer, Cham. – 2016. – Vol. 661. – P. 162–171.