The Hybrid Approach to Part-of-Speech Disambiguation

E. P. Bruches[1], D. A. Karpenko[2], and V. A. Krayvanova[3]

[1] OnPositive, Novosibirsk, Russia,
bruches@bk.ru
[2] OnPositive, Novosibirsk, Russia,
dmitry.karpenko@onpositive.com
[3] Altai State Technical University, Barnaul, Russia,
krayvanova@yandex.ru

Abstract. In this paper a hybrid approach to the part-of-speech disambiguation for Russian is presented. It combines the rule-based and the artificial neural networks oriented approaches. The methods are applied independently, then the results are compared, and the decision about the right part of speech for the word-form is made. The main advantage of the presented algorithm is the ability to suggest several parts of speech when it is difficult to select the right one with a fair degree of confidence. It reduces the probability of error during the syntactic and semantic levels analysis. Java implementation of the proposed algorithm is published under EPL.

Keywords: NLP, part-of-speech disambiguation, artificial neural networks, morphological analysis

1 Introduction

One of the barriers to use NLP in urgent needs for practical tasks is a lack of a free high-quality parser. The final aim of our work is to create such a parser. This paper describes our solution of one of the problems we faced with, namely a Part-of-Speech (POS) ambiguity and need of a good POS-tagging.

Over last few years in addition to traditional approaches to disambiguation (statistical[1] and rule-based[2]) machine learning based algorithms were proposed. In Stanford Tagger 2.0 Maximum entropy cyclic dependency network is used[3]. Santos et al.[4] have suggested the using MLP with Neural Character Embeddings. The tendency of applying machine learning algorithms to this task is also observed for the Russian language. Malanin[5] uses heterogeneous neural networks to a POS-tagging in Russian. Antonova and Solov'ev in paper[6] present the results of a Conditional Random Fields (CRF) approach.

Approaches based on the formal description of the rules for determining parts of speech, require much manual labour. Neural networks and other statistical approaches allow to extract implicitly these rules from annotated corpora, but the question about sufficiency of samples arises. Hence, in this paper we suggest our own approach to a POS disambiguation, which combines both artificial neural networks and rule-based approach.

2 Algorithm

We use Russian sentences as an input of the algorithm. We develop the tagging module which based on morphological dictionary. The dictionary is formed from the OpenCorpora[4] and the Wiktionary[5] data. The tags, which the dictionary consists of, are part of speech, gender, number, case etc. The full list can be found on our web-site[6]. At the initial stage a tagging module assigns morphological features to all the word-forms.

Let $Tags_w$ be a set of tags, which a word-form w is assigned with. A pair $T_w = \langle w, Tags_w \rangle$ is the token of the word-form w. The sequence of tokens is an input of the POS homonymy disambiguation module. Let $SP_0 \subset Tags_w$ be a set of POS tags, which a word-form assigned with. The following situations can occur as a result of the disambiguation algorithm.

- The single tag for a word-form was detected.
- The set of POS tags was decreased.
- The set of POS tags was keep which was assigned initially.

We apply the neural networks approach and the rule-based approach independently and then compare results to make a general decision of disambiguation. Let SP_r be a set of parts of speech, which were chosen by the rule-based approach and let SP_n be a set of parts of speech, which were chosen by the neural networks. If $SP_r \cap SP_n$ is not empty, then a word-form is assigned with all the tags from set $SP_{new} = SP_r \cap SP_n$. Otherwise a word-form is assigned with all the tags from set $SP_{new} = SP_r \cup SP_n$. For some words with POS homonymy one of the parts of speech appears in vast majority of cases, while the other one is extremely rare. We have defined POS-tag for such words beforehand and apply a filtration algorithm to process these words. The list of words which filtration is used for, can be found on the project web-site[7].

Let's consider each approach by itself.

There are several possible approaches to solving the task of POS disambiguation with neural networks. In the simplest case one network with outputs corresponding to all the recognized parts of speech can be trained. However, such architecture makes possible theoretically to get a part of speech, which was absent in the dictionary for this word-form. Another possible approach is to train an individual neural network for every set of POS tags, which occur a in training set. The size of training subsets for such networks is limited, and

[4] `http://opencorpora.org/`

[5] `https://ru.wiktionary.org/`

[6] `http://176.9.34.20:8080/com.onpositive.text.webview/materials/tagsets.html`

[7] `http://176.9.34.20:8080/com.onpositive.text.webview/materials/filters.html`

thus it cannot guarantee sufficient recognition quality. Therefore we use the third approach, when for every ambiguous pair of POS tags $\langle sp_1, sp_2 \rangle$ two neural networks are trained. Neural network N_{sp_1,sp_2} estimates the correctness of tag sp_1 in the presence of the alternative tag sp_2, and neural network N_{sp_2,sp_1} estimates it vice versa. For example, "двадцать минут ехали" (*'they were driving for twenty minutes'*) – the homonymous word "минут" (*'minutes'*/*'will pass'*) can be a noun or a verb, the right choice is a noun. Hence, neural network N_{sp_1,sp_2} is a classifier with a single neuron on the output layer, which estimates likelihood that the input vector corresponds to the part of speech sp_1. Training sets for neural networks are made from a part of an annotated text corpus from Open Corpora. Main corpus of OpenCorpora does not have full POS disambiguation. The cases, which in dictionary have POS ambiguity and which were disambiguated in the corpus, are used for training. We have found words, which in dictionary have POS ambiguity, and trained each network on such cases. The obtained cases are classified by parts of speech and homonymy. For uncommon pairs $\langle sp_1, sp_2 \rangle$ we are not able to collect a sufficient training set, so these homonymy cases are not examined in this approach.

The neural network analyses not the word-form but only its morphological image $Tags_n(w) \subset Tags(w)$, which includes the following key features: part of speech, case, gender, number, transitivity and aspect. The analysed context consists of three words: the homonymous word w_i, w_{i-1} to the left of it and w_{i+1} to the right of it, where i is the word number in the sentence. At the sentence boundary the parameters of previous or following word are replaced with zeros. The case where a phrase consists of one word is not analysed.

The network input is a vector of binary values (0 or 1), which is calculated as follows. Tags $Tags_n(w)$ for word-form w are converted into binary vector B with length M, where M is the number of tags in the corresponding set. Let x_t be the binary value of tag t in some grammatical category, then $x_t = 1$ if $t \in Tags(w)$ and $x_t = 0$ otherwise. For example, if x_t corresponds to the Nominative case, then $x_t = 1$ in the case when tagging module has assigned Nominative case to the current word-form. If the word-form has another case or even does not have such category, then $x_t = 0$. If more than one tag correspond to the current word-form, i.e. homonymy, then all the binary values x_t, which correspond to all the possible tags, will be set to 1 in the binary vector B. Let's consider the example of forming the data for the POS grammatical category. For simplicity let's use a reduced list of possible parts of speech: *noun*, *adjective*, *verb*, *aderb*. In this way, the word "трактор" (*'tractor'*), which is unambiguously tagged as noun, will be associated with binary vector $\langle 1, 0, 0, 0 \rangle$. The word "белила" (*'whitewash'*), which can be a noun or a verb depending on the context, will be associated with binary vector $\langle 1, 0, 1, 0 \rangle$. As a result, the group of binary vectors $\langle B_{i-1}, B_i, B_{i+1} \rangle$ will correspond to the current word and its context. The numbering of the word-forms in the sentence begins with 0. Four bits $E = \langle e_1, e_2, e_3, e_4 \rangle$, which encode word-form position concerning sentence boundaries, are added at the end of group $\langle B_{i-1}, B_i, B_{i+1} \rangle$. If $i = 0$ then $e_1 = 1$; if $i = 1$ then $e_2 = 1$; if $i = s_n - 2$ then $e_3 = 1$; if $i = s_n - 1$ then $e_4 = 1$, where s_n is word-form quantity in the sentence. Obtained vector of binary values is given as an input to all the neural networks, designed to resolve this homonymy case. For example, if there is an ambiguity "Noun – Verb", two networks $N_{noun,verb}$ and $N_{verb,noun}$ designed for cases with noun and verb respectively will be used. If the result of the neural network is above the specified threshold value, then the verified POS-tag is supposed to be correct. The algorithm described above performs within the bounds of a sentence for all the words, which have homonymy, from left to right. According to the training rate criterion and recognition quality, we use multilayer homogeneous neural networks with a sigmoidal activation function and a resilient backpropagation algorithm for a training. The input layer has 133 neurons, each of two hidden layers has 532 neurons, and the output layer has the single neuron.

Our rule-based approach to a POS disambiguation consists of two stages: rules applying and results testing with syntactic relations. In the algorithm the word-form context is considered within the bounds of a sentence. The algorithm applies the rules to each word-form w_i, for which tagging module assigned more than one part of speech, i.e. $|SP_0| > 1$. The rules are created manually and represent the pair $C(\langle T\rangle) \to sp$, where $C(\langle T\rangle)$ is a predicate, which has the context of analysed word-form w_i as input parameters; $\langle T\rangle$ is a set of tokens; sp is a part of speech, assigned to the analysed word-form w_i, if a predicate is true on this word-form context. Each rule is aimed at the choice between two parts of speech. All the rules are divided into several groups depending on the homonymy type they deal with. They are presented by two types:

1. The rules, that require the presence of certain grammatical features in the word-form context (for example, one of the rules to disambiguate the word-form "вести" (*'news'/'to lead'*) requires the presence of a preposition to define this word-form as an noun);
2. The rules, that require the absence of certain grammatical features in the word-form context (for example, one of the rules to disambiguate the word-form "том" (*'volume'/'that'*) requires the absence of a preposition to define this word-form as a noun).

The obtained sets of rules could be found on the web-site[8]. Before analysis the tokens $T_{w_{i-1}}$, T_{w_i}, $T_{w_{i+1}}$ are divided into parts of speech and all the possible sets with two and three tokens are formed. Let $T^{sp}_{w_i}$ be POS-tags from token T_{w_i}. According to the number of analysed tokens all the rules are divided into:

- triplet $\langle T\rangle = \langle T^{sp}_{w_{i-1}}, T^{sp}_{w_i}, T^{sp}_{w_{i+1}}\rangle$;
- paired prefix $\langle T\rangle = \langle T^{sp}_{w_{i-1}}, T^{sp}_{w_i}\rangle$;
- paired postfix $\langle T\rangle = \langle T^{sp}_{w_i}, T^{sp}_{w_{i+1}}\rangle$;
- paired neutral $\langle T\rangle = \langle T^{sp}_{w_i}, T^{sp}_{w_j}\rangle$, where $T^{sp}_{w_j} = T^{sp}_{w_{i+1}}$ or $T^{sp}_{w_j} = T^{sp}_{w_{i-1}}$.

For example, if $T_{w_{i-1}}$ has two possible parts of speech, and T_{w_i} has three possible parts of speech, then 2 * 3 = 6 possible combinations will be produced for prefix

[8] http://176.9.34.20:8080/com.onpositive.text.webview/materials/rules.html

and neutral types of rules. All the possible combinations for $\langle T^{sp}_{w_{i-1}}, T^{sp}_{w_i}\rangle$ are given as an input to all the prefix and neutral rules, all possible combinations for $\langle T^{sp}_{w_i}, T^{sp}_{w_{i+1}}\rangle$ are given as an input to all the postfix and neutral rules. The POS-tags from the initial set SP_0 for the word-form w_i, which are included into the phrases satisfying at least to one of the rules, are kept. At the beginning and at the end of the sentence the rules, which have only two arguments, are used as well as for the sentences consisting of two words. We do not apply this approach to the sentences consisting of the single word. After the word-form was assigned with the set of POS SP_r, the result is tested using a syntactic rules. The testing is as follows. There is the set of syntactic relations, which can be formed by the words with the certain POS-tag. If the word with the assigned POS-tag $sp \in SP_r$ is able to form the syntactic relation in the sentence, then sp is supposed to be correct. If some of POS-tags are verified by syntactic rules, the other POS-tags are removed. The algorithm proposes all the POS-tags, assigned by the rules, if there is no verified POS-tags.

3 Evaluation and Results

The evaluation of the proposed algorithm quality was carried out on two disambiguated text corpora: OpenCorpora (the part which is not used for training) and RusCorpora[9]. In order to evaluate the implementation quality for words with POS homonymy, we carried out the precision of partial disambiguation (at least on tag from assigned is correct) with formula 1.

$$P = \frac{W_{semiAmbig}}{W_{Ambig}},$$

where $W_{semiAmbig}$ is the number of words, for which $SP_{new} \cap SP_{etalon} \neq \varnothing$, SP_{new} is POS-tag set for current ambigious word-form, which was formed as a result of the algorithm implementation, SP_{etalon} is POS-tag set, which the word-form was assigned with in the corpus; W_{Ambig} is the number of words, which were initially assigned with more than one POS-tags. Another characteristic of the implementation quality is accuracy for all the words (including words without homonymy), which was carried out with formula 2.

$$Acc = \frac{W_{correct}}{W_{total}},$$

where $W_{correct}$ is the number of words, for which $SP_{new} \cap SP_{etalon} \neq \varnothing$, W_{total} is the total number of words. We also have calculated, how the size of the set of assigned POS-tags decreased after processing. Corpora sizes and evaluation results are shown in table 1. The last column shows what percent of extra POS-tags was removed. Such considerable divergence in results can be explained by the fact, that in these corpora the different tag sets and, moreover, the different approaches to determination of parts of speech are used.

[9] `http://www.ruscorpora.ru/`

Corpus	Total size	Homonymy words	Precision	Accuracy	Extra tags removed
OpenCorpora	33566	5880	96.11%	99.02%	93.04%
RusCorpora	169908	61137	86.39%	93.54%	89.22%

Table 1. Results

For comparison, the Trigram model, described in the paper [1], has precision 97,18% in the Part-of-Speech Tagging task. M-WANN-Tagger, presented in the paper [7], copes with the same task for Russian with precision 97,01%. The authors in [5] managed to achieve precision 96,56% in the Part-of-Speech Tagging using heterogeneous neural networks.

4 Conclusion

The proposed approach to the part-of-speech disambiguation for Russian combines both machine learning and expert systems. The results revealed that the algorithm copes with notional parts of speech, and it is less effective with functional ones. There is one of the priority lines for our future investigation. We are also going to improve this algorithm for multi-words and unknown words and to take into account punctuation. The java implementation of the proposed algorithm is available on our web-site[10].

References

1. Sokirko, A., Toldova, C.: The comparison of two methods of lexical and morphological disambiguation for Russian. Internet-mathematics 2005 (2005)
2. Brill, E.: A Simple Rule-Based Part of Speech Tagger. In: Proceedings of ANLC'92. (1992) 152 – 155
3. Manning, C.D.: Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? . In: Computational Linguistics and Intelligent Text Processing, 12th International Conference, Part I. Lecture Notes in Computer Science 6608. (2011) 171 – 189
4. Santos, C., Zadrozny, B.: Learning character-level representations for part-of-speech tagging . In: Proceedings of the 31st International Conference on Machine Learning, JMLR: W&CP. (2014)
5. Malanin, G.: Part-of-Speech tagging with heterogeneous neural networks and a priori information. Youth scientific-technical journal (12) (2014)
6. Antonova, A., Solov'ev, A.: Conditional Random Fields in NLP-related Tasks for Russian . Information Technologies and Systems (2013)
7. Carneiro, H., França, F., Lima, P.: Multilingual part-of-speech tagging with weightless neural networks. Neural Networks (16) (2015) 11 – 21

[10] http://176.9.34.20:8080/com.onpositive.text.webview/parsing/omonimy