A Hybrid System for MorphoSyntactic Disambiguation in Bulgarian[*]

Kiril Iv. Simov and Petya N. Osenova[†]
BulTreeBank Project
http://www.BulTreeBank.org
Linguistic Modelling Laboratory, Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A, 1113 Sofia, Bulgaria
kivs@bgcict.acad.bg, osenova@slav.uni-sofia.bg

1 Introduction

The interest in MorphoSyntactic Disambiguation Problem (MSDP) is supported by the hope that it is decidable with a high percentage of certainty without deep syntactic analysis to be involved. Our work is an attempt to solve this problem for Bulgarian using a hybrid system comprising Simple Recurrent Neural Network (SRN) component based on (Vlasseva 1999) and a rule-based component in order to disambiguate the cases for which there are rules ensuring 100% correct results. In our case the SRN component includes four layers of neurons: input, hidden, output and context. The input layer receives an encoding of the (possibly ambiguous) morphosyntactic features of the words in the sentences and the output layer represents the predicted by the net true features.

In general, a disambiguation problem is to attach the right category to a textual element from a set of possible categories for this element. In the case of MorphoSyntactic disambiguation we have to choose the right morphosyntactic features of a word in a text from the morphosyntactic features connected with the word in the lexicon. Usually the morphosyntactic features relevant to the word-forms are represented by a mnemonic symbols called *tags*. In our work we concentrate on the morpho-syntactic disambiguation within the sentence although in some cases the right choice depends on data from the surrounding text. Such cases are described in our lexicon and they receive a special marking before the work of the neural network. The output of the system contains this special marking and information about the most probable set of features predicted by the net.

In languages with rich morphology, like Bulgarian, the tagset is likely to increase in size. The main problem with having so many tags is the well-known problem of the sparseness of a corpus, i.e. from a set of linguistic descriptions only a few are frequent in the corpus. Thus representativeness with respect to all grammatical features relies on the very large size of the corpus. This phenomenon motivated us to choose compositional tags instead of atomic ones. As each word in our corpus is connected with a bunch of grammatical features, it happens that less amount of text demonstrates more dependencies between these features.

Our improvements over (Vlasseva 1999) are in several directions: (1) we extended the range of grammatical features predicted by the system to cover almost all paradigmatic members of Bulgarian words, (2) we changed the encoding schemata for grammatical features in order to minimize the computation and to use more extensively the context layer of the network, (3) we changed the evaluation of the network output in order to minimize the side effects from evaluating cases that are not relevant in a particular instance of ambiguity.

The structure of the paper is as follows: in the first section we present the encoding schemes for the grammar features and the evaluation of the output, next section discusses the rule-based component of the system, section 4 outlines the principle according to which we chose the sentences in the training corpus, at the end we conclude with some results and possible future work.

2 Modelling of MorphoSyntactic Disambiguation in SRNs

After we chose the SRN architecture for the solution of MSDP we had to answer the following questions: how to encode the morphosyntactic features carried by the words in the sentences and how to evaluate the output of the network.

[*] This work is funded by the Volkswagen Stiftung, Federal Republic of Germany under the Programme “Cooperation with Natural and Engineering Scientists in Central and Eastern Europe” contract I/76 887. It is also partially supported by Center of Excellence BIS-21 grant ICA1-2000-70016.

[†] Also at the Bulgarian Language Division, Faculty of Slavonic languages, St. Kl. Ohridsky University, Sofia, Bulgaria

The encoding of the grammatical features has to be done in terms of number of neurons and their values. If we choose to represent grammatical features as tags where each tag is a mnemonic name for a bundle of features that can be assigned to a lexical item, then one possible encoding would be the following: when each tag is represented by one neuron and if the tag is a possible description for the word then this neuron has value 1, otherwise its value is 0. Such encoding was accepted for the case of POS disambiguation in (Vlasseva 1999). A drawback of this kind of encoding is that it requires a very large number of neurons for a larger tagset. So we decided to choose a compromise and represent bundles of grammatical features instead of atomic tags. Thus, instead a tag to be represented as one neuron in the input and output layers, it is represented as several neurons depending on the grammatical features the tag stands for.

In our experiments the morphosyntactic information for each word in sentences is encoded as 36 neurons where the first 15 neurons represent all parts-of-speech in Bulgarian with participles, verbal adverbs, personal and possessive pronouns, and cardinal numerals separated as independent parts of speech, next three neurons stand for genders (masculine, feminine, neuter), then 3 neurons for number and so on for all morphosyntactic features in Bulgarian. For instance, the ambiguous wordform 'vyzhiteno' will be encoded as the following vector:

[0,**1**,0,0,**1**,0,0,0,**1**,0,0,0,0,0,0,0,0,**1**,**1**,0,0,**1**,0,0,0,0,0,0,0,**1**,0,0,0,0,0,0]

where the value 1 in the second position encodes the fact that this wordform can be a participle, the value 1 in the fifth position is for adjective, next 1 is for adverb, then the value 1 in the eighteenth represents the fact that the wordform is neuter gender if it is a participle or a noun. The next values 1 are for singular, indefinite and past passive. Thus this vector incorporates the three possibilities of grammatical features for 'vyzhiteno':

Past participle, neuter, singular, indefinite

[0,**1**,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,**1**,**1**,0,0,**1**,0,0,0,0,0,0,0,**1**,0,0,0,0,0,0]

Adjective, neuter, singular, indefinite

[0,0,0,0,**1**,0,0,0,0,0,0,0,0,0,0,0,0,**1**,**1**,0,0,**1**,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

Adverb

[0,0,0,0,0,0,0,0,**1**,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

This encoding of the morphosyntactic features for lexical items can be considered as several neural networks that work in parallel: one for part-of-speech disambiguation, one for gender, one for number and so on. In our case these networks are incorporated in one network. One advantage of this is the following: having all features represented simultaneously allows the network to learn not only dependency between values for one grammatical category but also between the values of several grammatical categories. For instance, the network can disambiguate between a verb and a noun on the base of the presence of an appropriate gender value for the next word.

The above encoding of grammatical features determines the evaluation of the network output. We proceed in the following order. First we determine the part-of-speech predicted by the network. It means that we consider the first fifteen neurons and choose the one with the highest value. In fact we decrease the range of the neurons among which we choose to represent the possible parts of speech for this word. In the above example, we consider only the second, the fifth and ninth neurons (we choose between participle, adjective and adverb). Then depending on the result for the part of speech disambiguation we check the values for other features in turn.

The actual use of the network includes the suggested encoding and the window-slide technique for prediction of the right features for each word in the text. As to the non-ambiguous words, the system copies their morphosyntactic features to the output. Actually the neural network gets activated, but the prediction is not taken into account and thus its only purpose is the adjustment of the neuron values to the next words in the sentences.

3 Rule-based Disambiguation

In the introduction it was mentioned that our system has also a rule-based component. We applied rules to disambiguate as many ambiguities on the morphosyntactic level as possible before applying neural network disambiguator. The general idea was to minimize the input ambiguities. Hence the rules that we invented are applicable in very specific contexts which determine the disambiguation with very high level of certainty—near 100%. Sometimes when we can not ensure such perfect rules, we rely on the neural network which works after the rule-based component has finished its job. We added some rules with less certainty but in this case we only modified the encoding of some morphosyntactic features without excluding any possibilities. For instance in the case of 'vyzhiteno' if we have a rule that says to us that it is very probable for the wordform in this use to be marked for gender and number then we reduce the value of the possibility for it to be an adverb. The encoding in this case could be something like this:

[0,**1**,0,0,**1**,0,0,0,**0.6**,0,0,0,0,0,0,0,0,**1**,**1**,0,0,**1**,0,0,0,0,0,0,0,**1**,0,0,0,0,0,0]
where the value for adverb is reduced to 0.6.

Another guideline for the development of rules is for them to solve ambiguities depending on lexical items that are far in the sentence. Such cases are hard to be solved by the window based approach of the neural network because the window couldn't cover the depending lexical items simultaneously.

4 Corpus

We paid a special attention to the preparation of the corpus of training and testing sentences. One of the main problems with small corpora is that of sparseness—when the tagset is large then most of co-occurrences of tags miss in the corpus and then the dependency between them cannot be learned. One thing against this problem in our system is the encoding of grammatical features instead of tags as it was described above. The second suggestion is for the system to be able to solve the most frequent cases of morphosyntactic ambiguity and so we decided to prepare a corpus that explicates them. In this section we describe the preparation of the corpus.

First, we collected a text corpus from Bulgarian texts available on Internet. It contains more than fifteen millions word usages and there are texts from different genres—16% of the texts come from fiction, 76% from newspapers and about 8% from others. The next step was to create a frequency vocabulary of the texts. The vocabulary contains all wordforms in the texts with the number of their usages. Then this vocabulary was analyzed by the program "Slovnik" (see (Popov, Simov and Chernokozhev 1997) based on (Popov, Simov and Vidinska 1998)) and a new frequency list was created but this time we counted morphosyntactic features, not words. In parallel to this we developed a small program about automatic extraction of sentences from the text. From the whole corpus we extracted 270 000 sentences. Then we ranked the extracted sentences with respect to frequency of morphosyntactic features of the words in the sentences divided by the length of the sentence. The formula for calculation of the rank of each sentence is:

$$r(S) = \frac{\sum\limits_{w \in S} f(w)}{len(S)},$$

where $r(S)$ is the rank of the sentence S, w is an ambiguous word in S, $f(w)$ is the number of occurrences of the morphosyntactic features of w in the corpus and $len(S)$ is the length of the sentence.

The corpus during the experiments contained 2500 sentences with the largest ranks. We divided the corpus into two parts: training part (1600 sentences) and test part (900 sentences).

5 Conclusion and Future Work

The results of the systems are: 95.17% accuracy for POS disambiguation and 92.87% for all grammatical features. It is important to note that most of the errors concern functional words like prepositions, particles and thus the system can be used for applications which are not concerned with these classes of words, like information retrieval.

The main directions of future work consist of making more experiments, especially with chosen at random sentences. We also plan to add a post-processing rule-based component in order to repair some of the known errors of the system. When we were choosing which morphosyntactic features to encode in the system we made our choice on two principles—first we chose features that are ambiguous for some wordform and second we chose features that probably play role in the disambiguation of other ambiguous features. We think about reducing the number of neurons in the encoding of the words.

References

Popov, D., Simov, K. and Chernokozhev, O. (1997) *Slovnik: Morphological processor for Bulgarian,* http://www.diogenes.bg/slovnik/index.html http://www.BulTreeBank.org/Resources.html

Popov, D., Simov, K. and Vidinska, S. (1998) *A Dictionary of Writing, Pronunciation and Punctuation of Bulgarian Language*, Atlantis SD, Sofia, Bulgaria.

Vlasseva St. (1999) *Part-Of-Speech Disambiguation in Bulgarian Language via Neural Networks*, Master's Thesis. Faculty of Mathematics and Computer Science, St. Kl. Ohridsky University, Sofia, Bulgaria.