Bulgarian Question Answering for Machine Reading

Kiril Simov[1], Petya Osenova[1], Georgi Georgiev[2] , Valentin Zhikov[2] and Laura Toloşi[2]

[1] Linguistic Modelling Department, IICT, Bulgarian Academy of Sciences
Acad. G.Bonchev St. 25A, 1113 Sofia, Bulgaria
{petya,kivs}@bultreebank.org
[2] Ontotext AD
Polygraphia Office Center, fl. 4, 47 A Tsarigradsko Shosse, 1504, Sofia, Bulgaria
{valentin.zhikov,laura.tolosi,georgi.georgiev}@ontotext.com

Abstract. In the CLEF 2012 the BulTreeBank Group of LMD, IICT, BAS is participating for QA4MRE task for Bulgarian. The system represented in the paper exploits an NLP Pipeline for Bulgarian in order to process the questions, answers and the supporting texts. Then we represent the results of the analysis as a bag of linguistic units - lemmas, dependency relations. These bags of words are the match between the question plus answer and the sentences in the text. The answer that maximizes the overlap is selected as the correct one. Since the system is deterministic we have only one run. The score achieved by the run is 0.29. The other two runs are performed as baseline runs with randomly selected answers. Their scores are 0.20 and 0.12, respectively. Thus, the using of linguistic units in the overlapping estimation provides significant improvements over the baseline.

Keywords. Linguistic NLP Pipeline, Linguistically-enhanced Similarity, Bag of Linguistic Units

1 Introduction

Bulgarian language has been included into the set of participating languages in CLEF tasks since 2004. It is the first time when Bulgarian systems have been tuned to the new format of the CLEF main task, namely: Question Answering for Machine Reading Evaluation (QA4MRE). For the previous formats, an infrastructure was designed (Osenova and Simov 2005), which included processing NLP tools and strategies in order to better handle the task requirements. These requirements were as follows: detection of correct answers to specific categorized questions in large corpora. Thus, an adaptation was needed of our previous architecture to the new conditions of better understanding small texts in pre-selected domains. Our approach focused on the analysis of overlapping linguistic structures. In order to do this, we first converted manually each question and possible answer into a declarative sentence. For example: Защо страдащите от деменция трябва да бъдат насърчавани да рисуват? (Why do the dementia sufferers have to be encouraged to take risks?) with a possible answer: защото това укрепва паметта, вниманието и възприемането (because this would strengthen their memory, attention and perception) are combined in the sentence: Страдащите от деменция трябва да бъдат насърчавани да рисуват, защото това укрепва паметта, вниманието и възприемането (The dementia sufferers have to be encouraged to paint, because this would strengthen the memory, attention and perception). Then these sentences and the supporting texts were analyzed by our NLP pipeline for Bulgarian. This pipeline includes the following linguistic processing steps: POS tagging, lemmatization and dependency structures. For each question-and-answer pair we extracted a bag of lemmas and triples: (dependent lemma, dependency relation, head lemma). This bag is then compared to the bag for each sentence. In this way, each paired question-and-answer was ranked with respect to the overlapping parts from sentences in the texts. As a next step, the answer that provided the largest overlap has been chosen. The advantages of such an approach are: handling of the structural ambiguity, such as active/passive alternations; pro-drop subjects; modification/predication, etc. During the mapping, we also included some new triples that were derived from the possible varieties of the answer in the supporting text.

Our group provided 3 runs - one was based on the processing described above, and two were performed via a random selection in order to have a baseline for the comparison. These two runs (2 and 3 in the uploaded information) provided the baseline - 0.12 and 0.20, respectively. The result of the system based on the linguistic processing is 0.29, which shows significant improvement over the baseline case.

The paper is structured as follows: next section described the NLP Pipeline for Bulgarian, which we are suing for processing of the data within the task; Section 3 presented the answer ranking using the result from the processing via the NLP pipeline; the last section concludes the paper and outlines some future direction of development.

2 The NLP Pipeline for Bulgarian

In this section we present the linguistic processing pipeline (BTB-LPP[1]) for Bulgarian which we used for analyzing of the data. BTB-LPP comprises three main modules: a *Morphological Tagger*, a *Lemmatizer* and a *Dependency Parser*.

2.1 Morphological Tagger

The morphological tagger is constructed as a pipeline of three modules - two stat-

[1] The pipeline is developed on the basis of the language resources, created within BulTreeBank project. The prefix BTB stands for BulTreeBank.

istical taggers trained on the Morphologically Annotated Part of BulTreeBank (BulTreeBank-Morph)[2] and a rule-based module exploiting a large Bulgarian Morphological Lexicon and manually crafted disambiguation rules.

SVM Tagger

The first statistical tagger uses the SVMTool (Giménez and Márquez 2004), which is a SVM-based statistical sequential classifier. It is built on top of the SVMLight (Joachims and Schölkopf 1999) implementation of the Support Vector Machine algorithm (Vapnik 1999). Its flexibility allows it to be trained on an arbitrary language as long as it is provided with enough annotated data. The accuracy of the tagging that was achieved with the optimal training configuration ranged from 89 % to 91 % depending on the text genre. Having applied the morphological lexicon as a filter on the possible tags for each word form together with the set of disambiguation rules, the best achieved result was 94.65 % accuracy of the tagging.

Rule-based Component

The task of this component is to correct some of the erroneous analyses made by the SVM Tagger. The correction of the wrong suggestions is performed by two sources of linguistic knowledge – the morphological lexicon and the set of context based rules. In the process of repairing we used as much as possible from the information provided by the SVM tagger. The context rules are designed in such a way that they aim at achieving higher precision even at the cost of low recall. The lexicon look-up is implemented as cascaded regular grammars within the CLaRK system – (Simov et. al 2001). The lexicon is an extended version of (Popov et. al 2003) and covers more than 110 000 lemmas. Additionally, a set of gazetteers were incorporated within the regular grammars. Here is an example of a rule: If a wordform is ambiguous between a masculine count noun (*Ncmt*) and a singular short definite masculine noun (*Ncmsh*), the *Ncmt* tag should be chosen if the previous token is a numeral or a number.

Guided Learning System: GTagger

GTagger is based on the guided learning system - (Georgiev et. al 2012). The best result of the tagging is 97.98 % accuracy. It can be considered the state-of-the-art for Bulgarian. However, this result is achieved when the input to GTagger is already tagged with the list of all possible tags for each token - similarly to the morphological dataset BulTreeBank-Morph. BTB-LPP provides such an input for GTagger exploiting the SVM Tagger as well as the rule-based component that tags some tokens with a list of the best possible candidate tags according to the morphological lexicon. Additionally, the set of rules is applied in order to solve some of the ambiguities.

The combination of the three components implements the morphological tagger of BTB-LPP. The SVM Tagger plays the role of a guesser for the unknown words. The rule-based component provides an accurate annotation of the known words, leaving some unsolved cases. GTagger provides the final result. This result is used by the lemmatizer and the dependency parser.

[2] http://www.bultreebank.org/btbmorf/

2.2 Lemmatizer

The second processing module of BTB-LPP is a functional lemmatization module, based on the morphological lexicon, mentioned above. The functions are defined via two operations on word forms: remove and concatenate. The rules have the following form:

if tag = *Tag* **then** {**remove** *OldEnd*; **concatenate** *NewEnd*}

where *Tag* is the tag of the word form, *OldEnd* is the string which has to be removed from the end of the word form and *NewEnd* is the string which has to concatenated to the beginning of the word form in order to produce the lemma. Here is an example of such a rule:

if tag = *Vpitf-o1s* **then** {**remove** *ох*; **concatenate** *а*}

The application of the rule to the past simple verb form for the verb *четох* (**remove**: *ох*; **concatenate**: *а*) gives the lemma *чета* (to read). Additionally, we encode rules for unknown words in the form of guesser word forms: #*ох* and tag=*Vpitf-o1s*. In these cases the rules are ordered.

In order to facilitate the application of the rules, we attach them to the word forms in the lexicon. In this way, we gain two things: (1) we implement the lemmatization tool as a part of the regular grammar for lexicon look-up, discussed above and (2) the level of ambiguity is less than 2% for the correct tagged word forms. In case of ambiguities we produce all the lemmas. After the morphosyntactic tagging, the rules that correspond to the selected tags, are applied.

2.3 Dependency Parser

Many parsers have been trained on data from BulTreeBank. Especially successful was the MaltParser of Joakim Nivre (Nivre et. al 2006). It works with 87.6 % parsing accuracy. The following text describes the dependency relations produced by the parser.

Here is a table with the dependency tagset, related to the Dependency part of the BulTreeBank. This part has been used for training of the dependency parser:

adjunct 12009	Adjunct (optional verbal argument)
clitic 2263	Short forms of the possessive pronouns
comp 18043	Complement (arguments of non-verbal heads, non-finite verbal heads, copula, auxiliaries)
conj 6342	Conjunction in coordination
conjarg 7005	Argument (second, third, ...) of coordination
indobj 4232	Indirect Object (indirect argument of a non-auxiliary verbal head)

marked 2650	Marked (clauses, introduced by a subordinator)
mod 42706	Modifier (dependants which modify nouns, adjectives, adverbs; also the negative and interrogative particles)
obj 7248	Object (direct argument of a non-auxiliary verbal head)
subj 14064	Subject
pragadjunct 1612	Pragmatic adjunct
punct 28134	Punctuation
xadjunct 1826	Clausal adjunct
xcomp 4651	Clausal complement
xmod 2219	Clausal modifier
xprepcomp 168	Clausal complement of preposition
xsubj 504	Clausal subject

Table 1. The Dependency Tagset

In addition to the dependency tags, also the morphosyntactic tags have been attached to each word (Simov et. al 2004). For each lexical node the lemma was assigned. The number under the name of each relation indicates how many times the relation appears in the dependency version of BulTreeBank.

Here is an example of a processed sentence. The sentence is Бразилия е епицентърът на пандемията на СПИН (Brazil is the epicenter of the AIDS pandemic.) After the application of the language pipeline, the result is represented in a table form following the CoNLL shared task format. It is given in Table 2.

No	WF	Lemma	POS	POSex	Ling	Head	Rel
1	Бразилия	Бразилия	N	Np	fsi	2	subj
2	е	съм	V	Vx	itf-r3s	0	root
3	епицентърът	епицентър	N	Nc	Msf	2	comp
4	на	на	P	P	-	3	mod
5	пандемията	пандемия	N	Nc	Fsd	4	prepcomp
6	на	на	P	P	-	5	mod
7	СПИН	СПИН	N	Nc	mfi	6	prepcomp

Table 2. The analysis of the Bulgarian sentence in CoNLL format**.**

The column *WF* corresponds to the order of the word forms in the sentence. The information in *Ling* column is the suffix of the corresponding tag (according to BulTreeBank morphosyntactic tagset) after removing the prefix represented in column *POSex* (extended POS). The elements in *Head* point to number of the dependency head of the given word form. The *Rel* is the dependency relation between the two wordforms.

In the next section we present the procedure for using the pipeline for the QA4MR task for Bulgarian.

3 Answer Ranking

In the process of answer selection for each question we have performed the following steps:

1. The supporting texts were processed by the NLP pipeline described in the previous section;
2. For each question and each potential answer of the question we constructed a declarative sentence which provides evidence that the potential answer is really an answer of the question;
3. The analyses of the sentences in the texts are compared for similarity with the analysis of the declarative sentence produced in step 2. In this way we rank the answers for each question.

In the rest of the section we describe in more details each of the steps.

Each sentence in the texts was presented as a bag of linguistic units where each unit is either a lemma, either a triple from the dependency tree for the sentence - <*DepLemma*, *Rel*, *HeadLemma*>. In the triple *DepLemma* is the lemma of the dependency node in the tree, *HeadLemma* is the lemma for the head node in the tree, *Rel* is the relation between the nodes. Thus, the ranking of the answers will be done on the basis of a sentence in the text and the bag of the selected linguistic units. The first decision is motivated by the limitation of the current processing pipeline which cannot establish reliable connections between the linguistic units in more than one sentence. The second decision is motivated by the fact that the matching of dependency trees might be very complicated, although we are aware of works on edit distance comparisons, such as the one used in (Kouylekov and Magnini, 2005).

In order to ensure the mapping between the analysis of the question-and-answer pair and the analyses of the sentences in the text we had to process the pair in the same way. The initial idea was to process the question and the corresponding answers separately, but there were some problems. The dependency parser is not good on fragments of sentences. But the answers are fragments in most cases. Some possible relations between the words in the question and in the answer had to be used. On the other hand, the ideal supporting sentence in the text would be would be composed of the question (potentially rearranged in a declarative form) and the answer in an appropriate way. Thus, we decided to convert each pair of a question and a potential answer into the best supporting declarative sentence. In our case this was done manually, but in future we envisage implementing this procedure automatically. Here are two examples:

Q1: Защо страдащите от деменция трябва да бъдат насърчавани да рисуват?
(Why do the dementia sufferers have to be encouraged to take risks?)

A1: защото това укрепва паметта, вниманието и възприемането
(because this would strengthen their memory, attention and perception)

D1: Страдащите от деменция трябва да бъдат насърчавани да рисуват, защото това укрепва паметта, вниманието и възприемането.
(The dementia sufferers have to be encouraged to paint, because this would strengthen the memory, attention and perception.)

Q2: Кой е епицентърът на пандемията на СПИН?
(Where is the epicenter of the AIDS pandemic?)

A2: Бразилия
(Brazil)

D2: Бразилия е епицентърът на пандемията на СПИН.
(Brazil is the epicenter of the AIDS pandemic.)

The ranking of each answer was done by calculating, first, the size of the intersection of the bag of linguistic units for each sentence in the texts and the bag for the pair's sentence. Then we calculated the maximum of the size of the intersections. This maximum was considered a rank of the pair question-and-answer and, thus, it is the rank of the answer. Then we selected the answer with the highest rank as an answer to the question. In case there is more than one answer with the same highest rank we selected randomly one of them.

For the type of questions for which we know possible variations of the realization of the answers in the text (see Osenova and Simov 2005), we also included triples in the bag for the pair question-and-answer. In this way we approached some basic cases of paraphrases.

Three runs of the system have been performed. One is the actual application of the above procedure. We also performed two random runs in order to establish a baseline. The two baseline runs were evaluated with scores: 0.12 and 0.20. The actual run received a score 0.29. This score shows a significant improvement over the baseline scores.

The error analysis showed two main problems for the method. First, in many cases the words in the question-and-answer pair differ from the words used in the text. Second, we did not implement enough paraphrased linguistic units.

4 Conclusion and Future Work

In the paper we presented a method for QA4MR for Bulgarian which exploits linguistic analyses of both - the question-and-answer pairs and the text. The similarity metric between the question-and-answer pairs and the sentences in the text is based on bag-of-linguistic units - lemmas and dependency relations between lemmas in the sentences. Our conclusion is that for improving the results, a good synonymic lexicon is needed to cover the lexical variety as well as the usage of more other knowledge resources, such as the provided background collections, various kinds of thesauri and domain-specific dictionaries for the specialized terms. Additionally, we need to extend the paraphrases generation mechanism. In future, we will also work on the inclusion of more semantic objects in the comparison algorithm using connections to ontological knowledge. Another restriction of the current method is that the comparison of the question-and-answer derived sentence is only with one sentence in the text. It is necessary to develop a better model that broadens the observations both - in the text and in the question-and-answer unit. We expect the approach, proposed here, would perform better on technical domains, where the degree of lexical variety is minimized and literal repetitions are used instead of synonyms.

References

1. Georgiev, Georgi, V. Zhikov, P. Osenova, K. Simov, and P. Nakov. 2012. Feature-rich part-of-speech tagging for morphologically complex languages: Application to Bulgarian. In: Proceedings of EACL 2012.
2. Giménez, J. and Márquez, L. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal.
3. Joachims, T. and Schölkopf, B. 1999. Making Large-Scale SVM Learning Practical. In: Burges, C. and Smola, A. (eds.), Advances in Kernel Methods - Support Vector Learning. Cambridge, MA, USA: MIT Press.
4. Kouylekov, Milen and Bernardo Magnini. 2005. Tree Edit Distance for Recognizing Textual Entailment. In Recent Advances in Natural Language Processing (RANLP-2005), Borovetz, Bulgaria.
5. Nivre, Joakim, Johan Hall, Jens Nilsson. 2006. Malt-Parser: A data-driven parser-generator for de-pendency parsing. In Proc. of LREC-2006, pp 2216-2219.
6. Petya Osenova, Kiril Simov. Infrastructure for Bulgarian Question Answering. Implication for the Language Resources and Tools. Piperidis and Paskaleva (eds). Proc. Workshop on Language and Speech Infrastructure for Information Access in the Balkan Countries. Borovetc, Bulgaria. 2005. pp 47-52
7. Popov, Dimitar, Kiril Simov, Svetlomira Vidinska, and Petya Osenova. 2003. Spelling Dictionary of Bulgarian. Nauka i izkustvo, Sofia, Bulgaria (in Bulgarian).
8. Simov, Kiril, Z. Peev, M. Kouylekov, A. Simov, M. Dimitrov, A. Kiryakov. 2001. CLaRK - an XML-based System for Corpora Development. In: Proc. of the Corpus Linguistics 2001 Conference. Lancaster, UK.
9. Simov, Kiril, Petya Osenova and Milena Slavcheva. 2004. BTB-TR03: BulTreeBank Morphosyntactic Tagset. BulTreeBank Technical Report № 03 (http://www.bultreebank.org/TechRep/BTB-TR03.pdf).
10. Vapnik, V. N. 1999. The nature of statistical learning theory (2nd ed.). New York: Springer.