BulQA: Bulgarian–Bulgarian Question Answering at CLEF 2005

Kiril Simov and Petya Osenova
Linguistic Modelling Laboratory, Bulgarian Academy of Sciences, Bulgaria
kivs@bultreebank.org, petya@bultreebank.org

Abstract

This paper describes the architecture of a Bulgarian–Bulgarian question answering system — **BulQA**. The system relies on a partially parsed corpus for answer extraction. The questions are also analyzed partially. Then on the basis of the analysis some queries to the corpus are created. After the retrieval of the documents that potentially contain the answer, each of them is further processed with one of several additional grammars. The grammar depends on the question analysis and the type of the question. At present these grammars can be viewed as patterns for the type of questions, but our goal is to develop them further into a deeper parsing system for Bulgarian. The CLaRK System is used as an implementation platform — [5].

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 Database ManagementLanguages [Query Languages]

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Answer support, Pattern grammars

1 Introduction

This paper describes the architecture and the linguistic processing of a question answering system for Bulgarian — **BulQA**. The system has three main modules: *Question analysis module*, *Interface module*, *Answer extraction module*. The *Question analysis module* deals with the syntactic and semantic interpretation of the question. The result of this module is independent from task and domain representation of the syntactic and semantic information in the question. The *Interface module* bridges the interpretation received from the first module to the input necessary for the third module. The *Answer extraction module* is responsible for the actual detection of the answer in the corresponding corpus. This architecture has the advantage that it allows the poly-usage of the same modules in different tasks, such as Bulgarian as source language in a multilingual question answering, or Bulgarian as a target language. In fact, only *the Interface module* has to be re-implemented in order to tune the connection between Bulgarian modules and the modules for the other languages.

In CLEF 2005 we have used the *Question analysis module* for two tasks: Bulgarian-English QA and Bulgarian-Bulgarian QA. The former is very similar to our participation at the CLEF 2004 ([6]) and for that reason is remains out of this paper's scope.

However, being participants in both tasks, we had to implement two versions of the *Interface module*. For the Bulgarian-English QA task the *Answer searching module* is based on the Diogene system ([4]) implemented at the ITC-Irst, Trento, Italy. For the Bulgarian-Bulgarian task we had implemented our own *Answer searching module*. This paper describes it in more detail.

Also the paper discusses the necessary resources and processing for answer support in different contexts. In this way we delimit the future developments of the system.

The structure of the paper is as follows: in section 2 we discuss language technology adaptation for the analysis of Bulgarian questions; section 3 describes the interface module; in section 4 we present the answer extraction approach on the basis of additional grammars. Section 5 comments on the necessary language resources and processing for more complicated answer supporting; the last section reports on the results of the question answering track and concludes the paper.

2 Linguistic Processing of the Corpus and the Questions

2.1 Processing the Corpus

The processing of the corpus is done in two steps: off-line and runtime. The goal is as much as possible processing to be done prior to the actual usage in the answer searching. The off-line processing tools are as follows: tokenization, named-entity recognition, morphological analyzer, neural-network based morphosyntactic disambiguation, chunking. These are the very basic tools which were widely used in our previous systems. We consider the results of these tools as reliable. For an overview of the available language resources and tools of Bulgarian and how they were used for Bulgarian-English task at CLEF 2004 see [6]. The result of this preprocessing of the corpus is stored as a set of XML documents with some indexing for searching with XPath language, which is implemented in the CLaRK system — [5]. Although the results of the preprocessing are still not very deep, they allow us to save time during the answer searching. In future we intend to extend the processing with additional information.

The runtime processing of the corpus is based on additional partial parsing modules that are tuned to the type of the questions, the type of the answer and to the type of the content of the questions. Thus we constructed new modules, such as specific partial analyses (we developed new partial grammars for more complex NPs with a semantic categorization, such as time, location and others). The reason these new processing modules have not been included in the off-line processing is that they depend too much on the information from the questions. Thus, they are likely to produce a wrong analysis if there is no appropriate information. The runtime processing is done only for a few documents that are retrieved from the corpus on the basis of the keywords derived from the questions.

2.2 Processing the Questions

The processing of questions is similar to the off-line processing of the corpus. In fact, we have enhanced the processing from the last year. The processing is mainly connected to the use of more elaborate semantic lexicon and module for processing of time expressions (i.e. dates, periods and event marking adverbials) in order to manage questions with temporal restrictions.

Here is an example of the analysis of the question "Koj kosmicheski aparat trygva za Lunata na 25 yanuari 1994 g.?" (in English: *Which space probe started for the Moon on 25 January 1994?*):

```
<analysis group="BTB">
    <NPA>
        <Pron><w ana="Pie-os-m" bf="koj">Koj</w></Pron>
        <A><w ana="Amsi" bf="kosmicheski">kosmicheski</w></A>
        <N><w ana="Ncmsi" bf="aparat">aparat</w></N>
```

```
    </NPA>
    <V><w ana="Vpiif-o3s" bf="trygvam">trygva</w></V>
    <PP>
        <Prep><w ana="R" bf="za">za</w></Prep>
        <N><name ana="Ncfsd" sort="LocNE" bf="Luna">Lunata</name></N>
    </PP>
    <PP sort="On_Date">
        <Prep><w ana="R" bf="na">na</w></Prep>
        <NPA sort="Date">
            <M><w ana="Mc--i" bf="25">25</w></M>
            <N><w ana="Ncmsi" bf="yanuari">yanuari</w></N>
            <M><w ana="Mc--i" bf="1994">1994</w></M>
            <N><abbr ana="Ncfsi" cat="lex" sort="Time"
                type="contr" exp="godina" bf="godina">g.</abbr></N>
        </NPA>
    </PP>
    <pt>?</pt>
</analysis>
```

Here each common word is annotated within the following XML element ⟨*w ana="MSD" bf="LemmaList"*⟩*wordform*⟨/*w*⟩, where the value of attribute *ana* is the correct morpho-syntactic tag for the wordform in the given context. The value of the attribute *bf* is a list of the lemmas assigned to the wordform. Names are annotated within the following XML element ⟨*name ana="MSD" sort="Sort"*⟩*Name*⟨/*name*⟩, where the value of the attribute *ana* is the same as above. The value of the attribute *sort* determines whether this is a name of a person, a location, an organization or some other entity. The abbreviations are annotated in a similar way, and additionally they have *type* and *exp* attributes which encode the type of the abbreviation (acronym or contraction) and its extension.

The next level of analysis is the result of the chunk grammars. In the example there are two *NPA* elements (NPA stands for a noun phrase of head-adjunct type), a lexical *V* element (lexical verb) and two *PP* elements. Also, one of the noun phrases is annotated as a date expression with a sort attribute with value: *Date*. This information is percolated to the preposition phrase which is annotated with the relation label *On_Date*. This is a result of the combination of the preposition meaning and the category of the noun phrase. The noun in the other prepositional phrase is annotated as a LOCATION name.

The result of this analysis had to be translated into the format which the answer extraction module uses as an input.

3 Interface Module

Here we describe the implemented interface module which translates the result of the question analysis module into the template necessary for the system, which extracts the answers of the questions. This module is an extension of the module we have implemented for the Bulgarian-English task. The main difference is that we do not transfer the question analyses into DIOGENE's type of template with English translations of the keywords, but instead we define a set of processing steps for the Answer searching module. The processing steps are of two kinds: corpus processing and document processing. The first processing step retrieves documents from the corpus that potentially contain the relevant answers. The second one analyzes additionally the retrieved documents in order to extract the answer(s). The process includes the following steps:

- Determining the head of the question.

  The determination of the question head was performed by searching for the chunk which contains the interrogative pronoun. There were cases in which the question was expressed

with the help of imperative forms of verbs: *nazovete* (name-plural!), *kazhete* (point out-plural!; say-plural!). After the chunk selection we classify the interrogative pronoun within a hierarchy of question's heads. In this hierarchy some other elements of the chunks — mainly prepositions — play an important role as well.

- Determining the head word of the question and its semantic type.

  The chunk determined in the previous step also is used for determining the head word of the question. There are five cases. First, the chunk is an NP chunk in which the interrogative pronoun is a modifier. In this case the head noun is the head word of the question. For example, in the question: **What nation** *is the main weapons supplier to Third World countries?* the noun 'nation' is the head word of the question. In the second case the chunk is a PP chunk in which there is an NP chunk similar to the NP chunk from the previous case. Thus, again the head noun is a head word for the question. For example, in the question: **In what music genre** *does Michael Jackson excel?* the noun 'genre' is the head word of the question. Third, the interrogative pronoun is a complement of a copula verb and there is a subject NP. In this case the head word of the question is the head noun of the subject NP chunk of the copula. For example, in the question: **What** *is a basic ingredient of Japanese cuisine?* 'ingredient' is the head of the question. The fourth case covers the questions with imperative verbs. Then again the head of the question is the head noun of the complement NP chunk. For example, in the question: *Give a symptom of the Ebola virus.* the noun 'symptom' is the head of the question. The last case covers all the remaining questions. Then the head word of the question is the interrogative phrase (or word) itself. For example, in the question: **When** *was the Convention on the Rights of the Child adopted?* the head of the question is the interrogative word 'when'. The semantic type of the head word is determined by the annotation of the words with semantic classes from the semantic dictionary. When there are more than one semantic classes we add all of them. The type of the interrogative pronoun is used later for disambiguation. If no semantic class is available in the dictionary, then the class 'other' is assigned.

- Determining the type of the question.

  The type of the question is determined straightforwardly by the semantic type of the head word. For the recognition of the questions with temporal restriction we count on the pre-processing of the questions and the assigned temporal relations. As temporal restriction we consider such expressions that are not part of the head of the question.

- Determining the keywords of the question and their part of speech.

  The keywords are determined by the non-functional words in the question. Sometimes it is possible to construct multi-token keywords, such as names (Michael Jackson), terms or collocations. For the Bulgarian-Bulgarian task this is important when there are special rules for query generation for document retrieval (see next section). We also used gazetteers of abbreviated forms of the most frequent organizations in English. This was very helpful in finding the correct answers to the Definition Organization questions because in many cases these abbreviations lack Cyrillic counterparts, and thus the search is very direct even in the Bulgarian corpus. Only the extensions seem to have systematically Cyrillic counterparts, and therefore they need more complex processing sometimes.

4 Answer Extraction and Validation

The answer extraction is a two-step process: first, the documents possibly containing the answer are retrieved from the corpus; then the retrieved documents are additionally processed with special partial grammars which depend on the type of answer, the type of the question and the found keywords in the document. We can view these grammars as patterns for the different types of questions.

As it was mentioned above, for document retrieval we are using CLaRK system. The corpus is presented as a set of XML documents. The search is done via XPath language enhanced with index mechanism over the (selected) content of each document. The initial step of the answer extraction is done via translating of the keywords from the analysis of the question into an XPath expression. This expression selects the appropriate documents from the corpus. The expression itself is a disjunctive where each disjunct describes some combinations of keywords and their variants. The variants are necessary because the keywords in the question bear different degree of informativeness with respect to the answer (see the discussion below on the answer support). For example, for named entities we constructed different (potential) representations: *Michael Jackson* can be *M. Jackson* or only *Jackson*. Where possible, we convert the corresponding keyword to a canonical form (for example, dates) and we simply match the canonical forms from the corpus and the question.

Definition questions provide one key word or expression. Thus, they are easily trackable at this stage. For example ('Who is Nelson Mandela?' has a key expression 'Nelson Mandela'). However, the factoid questions are more difficult to process even at that general stage. Obviously, the reason is that the question key words are not always the best answer-pointers. This is the reason we to develop our own search engine instead of a standard one. This envisages future developments when we will maximally use the implicit lexical information and incorporate more reasoning along the lines of contemporary investigations of paraphrases, entailment and different degrees of synonymy.

When the documents are retrieved, they are additionally processed in the following way: first, the keywords (the ones from the question and its variants or synonymical expressions) are selected. Then special partial grammars (implemented as cascaded regular grammars in the CLaRK System) are run within the contexts of the keywords. These grammars use the information about the type of the answer and how it is connected to the keywords. The context of a single keyword (or phrase) can be explored by several different grammars and (potentially) several possible answers. If we found more than one answer we apply some additional constraints to select one of them as result. In case no answer was found, the NIL value is returned.

The implementation of this architecture is done in the CLaRK system. The pattern grammars are still not enough with respect to the different kinds of questions. Thus, for other types of questions the resources that we have for Bulgarian are not suffice for real question answering, and only some opportunistic patterns can be implemented. As we would like to develop the system along the lines of knowledge rich question answering systems we did not try to implement many such opportunistic patterns, but more effort was invested in classification of the contexts that support the answers. Next section is an attempt to characterize the processing that we would like to incorporate in the future developments.

5 Discourse Requirements for Answer Support

As stated in CLEF 2005 guidelines, each type of question has an abstract corresponding answer type, but when the answer is in a real context, there exists a scale with respect to the answer acceptability. And the concrete answer must be mapped against this scale. The change of the context can change the answer grade in the scale. In this section we will try to give some examples of answers supported by different contexts.

We consider the text as consisting of two types of information: (1) ontological classes and relations, and (2) world facts. The ontological part determines generally the topic and the domain of the text. We call the corresponding "minimal" part of ontology implied by the text *ontology of the text*. The world facts represent an instantiation of the ontology in the text. Both types of information are called uniformly 'semantic content of the text'. Both components of the semantic content are connected to the syntactic structure of the text. Any (partial) explication of the semantic content of a text will be called *semantic annotation of the text*[1]. The semantic content of a question includes some required, but underspecified element(s) which has(have) to be specialized

[1]Defined in this way the semantic annotation could contain also some pragmatic information and actual world knowledge.

by the answer in such a way that the specialization of the semantic content of the question has to be true with respect to the actual world.

We consider a textual element a to be an supported answer of a given question q in the text t if and only if the semantic content of the question with the addition of the semantic annotation of the textual element a is true in the world[2].

Although the above definition is quite vague it gives some ideas about the support that an answer receives from the text in which it is found. The semantic annotation of the answer comprises all the concepts applicable for the textual element of the answer and also all relations in which the element participated as an argument[3]. Of course, if we had the complete semantic annotation of the corpus and the question, it would be relatively easy to find a correct answer of the question into the corpus, if such exists. Unfortunately, such an explication of the semantic annotation of the text is not feasible with the current NLP technology. Thus we are forced to search for an answer using partial semantic annotations. In order to give an idea of the complexity necessary in some cases we would like to mention that the context which has to be explored can vary from a phrase (one NP), to a clause, a sentence, a paragraph, the whole article or even the whole issues. The required knowledge can be linguistic relations, discourse relations, world knowledge, inferences above the semantic annotation.

Here are some examples of dependencies with different contexts and a description of the properties necessary to interpret the relations:

Relations within NP. Bulgarian nominal phrase is very rich in its structure. We will consider the following models:

NP :- NP NP

This model is important for two kinds of questions: definition questions for people and questions for measurement. The first type of question is represented by the abstract question "Koj e Ime-na-chovek?" (Who is Name-of-a-Person?): Koj e Nikolaj Hajtov? (Who is Nikolaj Hajtov?). As it was discussed in [7] some of the possible patterns that can help us to find the answer to the question are: "NP Name", "Name is NP" where the Name is the name from the question and NP constitutes the answer. The first pattern is from the type we consider here. The other one and some more patterns are presented below. Although it is a very simple pattern the quality of the answer extraction depends on the quality of the grammar for nominal phrase. The first NP can be quite complicated and recursive. Here are some examples:

[NP klasikyt] [NP Nikolaj Hajtov]
(the classic Nikolaj Hajtov)
[NP golemiya bylgarski pisatel] [NP Nikolaj Hajtov]
(the big Bulgarian writer Nikolaj Hajtov)
[NP zhiviyat klasik na bylgarskata literatura] [NP Nikolaj Hajtov]
(the alive classic of the Bulgarian literature Nikolaj Hajtov)
[CoordNP predsedatel na syyuza na pisatelite i zhiv klasik na bylgarskata literatura]
[NP Nikolaj Hajtov]
(chair of the committee of the union of the writers and alive
classic of the Bulgarian literature Nikolaj Hajtov)

As it can be seen from the examples, the first NP can comprise a head noun and modifiers of different kinds: adjectives, prepositional phrases. It also can exemplify coordination. Thus, in order to process such answers, the system needs to recognize correctly the first NP. This step is hard for a base NP chunker (being nonrecursive), but when it is combined with semantic information and a named-entity module, then the task is solvable. A characteristic for the first NP is that the head noun denotes a human. If such nouns are mapped to ontological characteristics, the work of the tool is facilitated.

[2]World such as it is described by the corpus.

[3]We consider the case when the answer denotes a relation to be a concept.

Another usage of this NP recursive model concerns measurement questions, such as: "Kolko e prihodyt na "Grijnpijs" za 1999 g.?" (How much is the income of Greenpeace for 1999?). The answers to such questions have the following format: "number", "noun for number", "noun for measurement". For example, "[NP 300 miliona] [NP dolara]" (300 million dollars). The NPs are relatively easy to recognize, but their composition remains unrecognized in many cases and the systems return partial answers like '300 million' or only '300'. However, without the complete measurement information such an answer is not quite correct and is discarded.

Problems arise when there are longer names of organizations with embedded PPs or with contacting PPs which are not part of them. The systems often return some NP, but the thing is that they suggest either the dependant NP as an answer instead of the head one, or an NP, which is a part of a PP not modifying the head NP. An example for the first case is the answer to the question: What is FARC? The system answered 'Columbia' instead of answering 'Revolutionary Armed Forces of Colombia' or at least 'Revolutionary Armed Forces'. An example for the second case is the answer to the question: What is CFOR?. It was 'Bosnia' instead of 'command forces' (in Bosnija).

Another interesting case is when the first NP has the form AP NP where AP is a relational adjective connecting the noun with another noun like: italianski (Italian)− >Italy, ruski (Russian)− >Russia, etc. In this case the answer of questions like "Ot koya strana e FIAT?"(Where does FIAT come from?) or "Na koya strana e prezident Boris Yelcin?" (Of which country Boris Yelcin is the president?) is encoded within the adjective. This means that we should have lexicons, which are interrelated in order to derive the necessary information even when it is indirectly present in the text. Note that this does not hold only within NPs. For example, the answer of the question 'Who was Michael Jackson married to?' could be 'Michael Jackson's ex-wife Debby'. Of course, here the relation is more complex, because there is a relation not only between 'marry' and 'wife', but also temporal mapping between 'was married' and 'ex-wife'.

NP :- (Parenthetical NP) | (NP Parenthetical)

Such NPs are relevant for definition questions about the extensions of acronyms: Kakvo e BMW? (What is BMW?). Very often the answers are presented in the form of an NP, which is the full name of the organization and the corresponding acronym is given as a parenthetical expression in brackets, or the opposite. In this case two gazetteers: of acronyms and the corresponding organization names would be of help. Additionally, we have to rely on opportunistic methods as well, because it is not possible to have all the new occurrences in pre-compiled repositories. Then, the case with the extension as parenthesis is easier to handle than the opposite case. Recall the problems with defining the boundaries of a complex name.

NP :- NP RelClauss

Here the main relations are expressed via the following relative pronoun. It is a kind of local coreference. Let us consider the example: 'Mr Murdoch, who is the owner of several newspapers'. We can trace who is Murdoch through the relative clause. However, sometimes it might be tricky, because in complex NPs we do not know whether the relative clause modifies the head NP or the dependant one. For example, in the phrase: 'the refugee camp in the city, which is the biggest in the country', we cannot know whether the camp or the city is the biggest in the country.

Relations within a clause (sentence). In order to derive the relevant information, very often we need the availability of relations among paraphrases of the same event. This idea was discussed in [1], [2] and [3] among others. For that task, however, the corpus should be annotated with verb frames and the grammatical roles of their arguments. Additionally, lists of possible adjuncts are also needed, because they are mapped as answer types to questions for time, measure, location, manner. Thus we have to go beyond the argument structure annotation. The ideal lexical repository should include relations between semantic units, such as if something is a location, you can measure distance to it; if something is an artefact, you can measure its cost etc. Also, the

classical example with the entailment like: if you write something, then you are its author, can be derived from a rich explanatory dictionary, which is properly parsed.

Discourse relations. They are necessary, when the required information cannot be assessed locally. When some popular politician is discussed in the newspaper, it might be the case that he is addressed only by his name, not the title: 'Yaser Arafat' instead of 'the Palestinian leader Yaser Arafat'. In such cases we need to navigate through wider context and then the marked coreferential relations become a must: Yaser Arafat is mentioned in the sentence, then in the next one he is referred to as 'the Palestinian leader' and finally, as 'he'. Here we could rely on anaphora resolution tools and on some gathered encyclopedic knowledge.

World knowledge. We usually rely on our world knowledge when there is more specific information in the questions and more general in the candidate answers. For example, to the question 'Who is Diego Armando Maradona?' we found answers only about 'Diego Maradona' or 'Maradona'. For this case we could be sure that all these names belong to the same person. However, there could be trickier cases like both Bush - father and son. If the marker 'junior' or 'senior' is not there, then we have to rely on other supportive markers like temporal information or some events that are connected with the one or the other.

6 Results and Outlook

The result from our Bulgarian-Bulgarian QA track can be viewed as a preliminary test of our QA system. We got the following statistics: 37 out of the 200 extracted answers were correct, 160 were wrong and 3 inexact. The distribution of the correct answers among the question categories is as follows: 21 definition questions: 13 for organizations and 8 for persons; 16 factoid questions: 2 for locations, 2 for measure, 1 for organizations, 2 for other categories, 2 for persons, and 3 for time. For the temporal restricted questions: 2 for locations and 2 for organizations. The main problems that we encountered during the contest were as follows: (1) the lack of a complete set of relevant QA processing tools for Bulgarian, (2) for some of the questions we were not able to run the procedure because of business travels during the testing period. Thus, for about one third of the questions the answer NIL was stated without a real application of the system.

Our plans for future work are to build on our experience from CLEF 2005 participation. We plan to implement more pattern grammars and to enrich the resources for Bulgarian in two aspects: (1) qualitative – better integration of the available resources and tools, and (2) quantitative – creation of more support grammars for the off-line procedure.

References

[1] Ido Dagan and Oren Glickman. *Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability.* Learning Methods for Text Understanding and Mining Workshop. Available at: http://www.cs.biu.ac.il/ glikmao/Publications

[2] Milen Kouylekov and Bernardo Magnini.*Recognizing Textual Entailment with Tree Edit Distance Algorithms.* PASCAL Challenges Workshop. Available at: http://www.kouylekov.net/Publications.html

[3] Dekang Lin and Patrick Pantel.*Discovery of Inference Rules for Question Answering.* In: Natural Language Engineering 7(4):343-360.

[4] Negri M., Tanev H., and Magnini B.: Bridging Languages for Question Answering: DIOGENE at CLEF-2003. Proceedings of CLEF-2003, Trondheim, Norway. (2003) 321–330

[5] Simov, K., Peev, Z., Kouylekov, M., Simov, A., Dimitrov, M., Kiryakov, A.: CLaRK — an XML-based System for Corpora Development. Proceedings of the Corpus Linguistics 2001 Conference. (2001) 558–560

[6] Petya Osenova, Alexander Simov, Kiril Simov, Hristo Tanev, and Milen Kouylekov. *Bulgarian-English Question Answering: Adaptation of Language Resources.* In (Peters, Clough, Gonzalo, Jones, Kluck, and Magnini eds.) Fifth Workshop of the Cross–Language Evaluation Forum (CLEF 2004), Lecture Notes in Computer Science (LNCS), Springer, Heidelberg, Germany. (2005)

[7] Hristo Tanev. *Socrates: A Question Answering Prototype for Bulgarian.* In Proceedings of RANLP 2003, Borovetc, Bulgaria. pp. 377-386. (2003)