Developing a Question Answering System for the Slovene Language

INES ČEH, MILAN OJSTERŠEK
Laboratory for Heterogeneous Computer Systems
Faculty of electrical engineering and computer science
Smetanova 17, 2000 Maribor
SLOVENIA
ines.ceh@uni-mb.si, ojstersek@uni-mb.si

Abstract: - In today's world the majority of information is sought after on the internet. A common method is the use of search engines. However since the result of a query to the search engine is a ranked list of results, this is not the final step. It is up to the user to review the results and determine which of the results provides the information needed. Often this process is time consuming and does not provide the sought after information. Besides the number of returned results the limiting factor is often the lack of ability of the users to form the correct query. The solution for this can be found in the form of question answering systems, where the user proposes a question in the natural language, similarly as talking to another person. The answer is the exact answer instead of a list of possible results. This paper presents the design of a question answering system in natural slovene language. The system searches for the answers for our target domain (Faculty of Electrical Engineering and Computer Science) with the use of a local database, databases of the faculty's information system, MS Excel files and through web service calls. We have developed two separate applications: one for users and the other for the administrators of the system. With the help of the latter application the administrators supervise the functioning and use of entire system. The former application is actually the system that answers the questions.

Key-Words: question answering, Slovenian language, morphological dictionary of Slovenian language, Question Classification, machine learning, question templates, personalization

1 Introduction

The basic idea of question answering systems is to be able to provide answers to questions written in natural language. The answers can be retrieved from different sources, e.g. web pages, plain texts, knowledge bases, web services etc. Unlike the information retrieval applications like web search engines that flood their users with documents or best-matching passages, the goal of question answering systems is to find a specific answer.

There are many ways of looking at question answering, and they depend on the approaches towards various dimensions [1]. The aforementioned dimensions are: the question, the answer, the technique, the information source, the domain, and the evaluation. The dimensions and their mutual relationship as well as the connections are represented on Fig. 1.

Currently the most extensive source of information, which is also used by question answering systems, is the World Wide Web. The World Wide Web was not designed for mere communication purposes but to contain information, therefore the idea that computers should also be capable of collaboration and automatic task management arose.

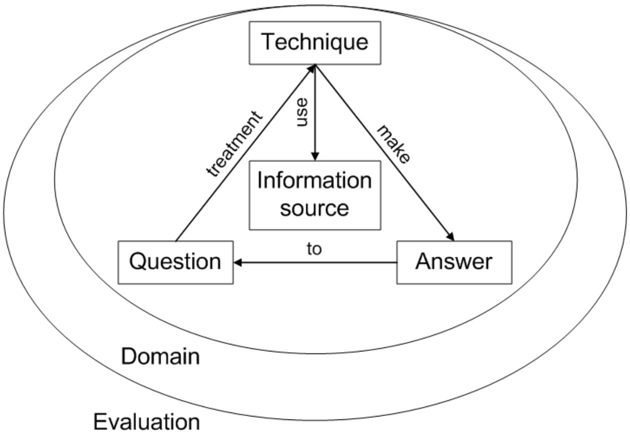


Fig. 1: Dimensions of the system and connections between them

However, the majority of information available on the web is suitable only for human use. Even the template based documents, e.g. the documents in different data bases, of which structure and meaning are defined, do not alleviate the work for programme agents [2]. There was a need for a change, and the idea of the Semantic Web was born [3], [4].

The Semantic Web is an extension of the World Wide Web. In the Semantic Web information is defined with unambiguous computer-understandable metadata, thus enabling computers and people to work in cooperation. One of the most important components of the semantic web are ontologies which can significantly enhance the functioning of the web in many ways [3], [5].

The next generation of natural language question answering systems will use the structure and elements of the Semantic Web in order to improve the web searches [6].

Students at our faculty (Faculty of Electrical Engineering and Computer Science, FERI) are faced with various questions during their studies. In their search for answers, they browse the web portal, or answers are provided by the employees of the faculty via e-mail, forum, or telephone. In order to improve the retrieval of information for students and alleviate the burden on the employees, a closed-domain question answering system had to be developed.

Since no solutions for our target domain existed, our system was the first step towards the introduction of question answering systems in education. It was one of the first question answering systems in the Slovene language.

The article is segmented into five chapters. The following chapter describes the analysis which was done before the system was implemented. The third chapter describes the definition of the approaches towards the dimensions of the system. The fourth chapter describes the implementation of the system. Chapter five concludes with the summary and some suggestions for our future work.

2 Problem analysis

Before the system was implemented, we thoroughly examined the expectations of users and their relationship to the system. More than 500 questions were chosen from the faculty's discussion board (forum) in order to develop and test the system. The questions were acquired from the forum because they were written in natural language, and we were in the process of developing a system that should answer similar questions. Of course we did not expect the identical behaviour of users in the system and the forum. We presumed that the questions that were asked on the forum would carry a fair amount of information. Because users are aware of the fact that they will have to wait for the answer, they form their questions carefully. In comparison to the questions that were asked on the forum, we expected that the degree of spontaneity in the questions that would be proposed to our question answering system will increase, as well as there will be an increase in their quantity, thus they will carry less information. During the analysis, three types of questions have been identified:

- Short and concise questions. Only one question without any additional information or the description of the problem situation is asked.
- Several questions at the same time. Several questions on the same topic with potential additional information are asked.
- Extended questions. The extensive description of the problem is followed by a question.

We decided that our system will offer direct support to the first and the third questions types. The second, however, is supported only indirectly, since users can ask multiple questions, one at a time.

During the analysis, we also observed the value of information in the questions. It was established that:

- Some questions are incomplete.
- Some questions carry redundant information.

An example of an incomplete question would be a user asking a question that refers to a certain person, that person is usually the same user that is asking the question, but there is no mentioning of the user's identity. We decided to solve the problem of incomplete questions with the help of personification and a dialog between the system and the user. If the question is incomplete, the system will try to sort out the missing information from the database containing user related information, or it will request additional information directly from the user. In contrast to users who are generally asking incomplete questions, there are also users who ask questions with redundant information that exerts no influence on the answer. In order to implement our system successfully, we decided to treat such questions equally to other types of questions. Consequently, all information in such a question was considered as equally important.

During the analysis, we were also interested in the language skills of users in general. Based on information in [7] and our own observations, we noticed the following behaviour of users:

- use of colloquial words (their writing style is similar to spoken language),
- deviation from the orthographic norms,
- use of foreign words (mostly English),

- extensive use of punctuation marks and emoticons,
- omission of Slovenian letters/sounds č,ž,š,
- omission of punctuation marks,
- omission of capitalization (proper nouns, beginning of sentence),
- typing errors and a
- combination of the above deficiencies.

In order to answer the questions written in colloquial language, the colloquial words would have to be selected and replaced with their standard counterparts manually which would substantially extend the development of the system. Therefore we decided to develop a system that at present answers only to questions proposed in standard Slovene. The only exceptions are the punctuation marks and their omissions, emoticons, and omissions of capitalization that are not taken into consideration. This decision was taken because these elements of language do not influence the quality of the answers at this point of the development of the system.
We also observed the increased use of:

- acronyms and
- abbreviations.

We decided that our system will support acronyms. Abbreviations, on the other hand, would not be supported due to the same reason as mentioned before.
During the analysis, we also established that, as expected, there is no aggressive behaviour on the forum. Nevertheless, we had expected the aggressive behaviour in the users of our system, because we gave our system a female name »Sara«. Such a name should suggest that the users can communicate with the system in their natural language. In such a way the communication with the system was stimulated. Studies on the relationship of users towards the chatting programs show a high grade of aggressive behaviour; therefore it was justified to expect the aggressive speech of users [8], [9].

3 Definition of the approaches towards the dimension of the system

The approach towards the dimensions of the system was determined based on the analysis of the questions, which we presented in chapter 2. The dimensions are the following:

- Evaluation: The system will be evaluated according to the percentage of accurate answers.
- Domain: It will be a closed-domain system. The system's structure will be general, so it will be employable also in other domains.
- Information source: The system will retrieve answers from the following sources: local database, databases of the faculty's information system, MS Excel files, and web services.
- Technique: The system's search for answers will be based on the principle of full matching or partial matching (at least in the minimum number of required keywords) of a question.
- Answer: According to the research described in [10] and our own judgment, we decided that the answers will not be exact (a word or a word phrase), but they will be in the form of a sentence or a paragraph. The answer in the form of a sentence will not contain any additional information that could be of interest to the user in his/her future questions, while the answer in the form of a paragraph will. If required, the answers will be accompanied by hyperlinks to web pages or documents containing additional information.
- Question: The system will provide answers to the following types of questions: insulting questions, ex. "You are stupid!", special questions and phrases, ex. "What is your name?", questions referring to the employees of the faculty, ex. "Where is professor Johnson's office?", template based questions which consist of the data stored in local databases and MS Excel files, ex. "What is the amount of the description fee?", and other questions, ex. "When are the office hours of the student office?".

According to the analysis, the system should also have the following properties:

- The system will be capable of forming a simple dialog with the user in order to retrieve information that could improve its answers.
- The system will save user-specific information and information on the activity of users in order to deliver a user-specific answer.
- The system will provide answers to the questions which are written in standard Slovene.
- The system will carry a female name *Sara* in order to stimulate the use of the natural language.

We developed two applications: one for users and one for the administrators of the system. With the help of the latter application the administrators expand the database of questions and answers, import the data from local databases and MS Excel files, integrate web services into the system, and supervise the functioning and use of entire system. The former application is actually the *system* that answers the questions.

As depicted on Fig. 2, which presents the architecture of our system, the user application makes use of a personalization server, which provides information needed for the personalized method of answer retrieval. The application is also linked to the search engine on the faculty portal, which provides search results for questions that cannot be answered.

Both applications are using the morphological dictionary of the Slovene language and a domain specific dictionary.

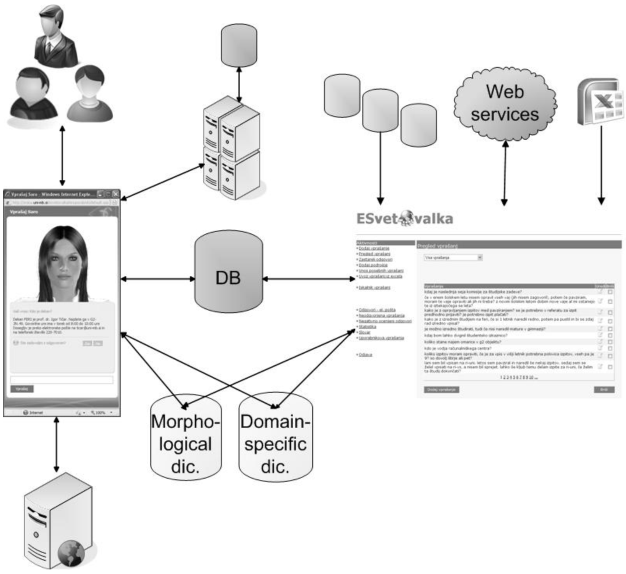


Fig. 2: Arhitecture of our system

4 Implementation of the system

4.1 Application for administrators

As previously mentioned, one of the searching techniques is based on the principle of partial matching. The question in the database and the user's question are similar when they contain the highest possible number of the same keywords. A keyword is not necessarily just one word, but it can also be a series of words. Keywords must be in their lemmatised form in order to be successfully compared.

In order to find lexical forms, words must undergo the lemmatization procedure which is highly complicated when dealing with the inflectional languages like Slovene [11], [12]. The morphological dictionary of the Slovene language, which contains approximately 8000000 declination and conjugation forms of words and 300000 lemmas, was our source of the lemmatised forms of words. Despite the large number of words, the morphological dictionary does not contain all words that occur in the domain of the faculty, therefore it was decided that a closed-domain dictionary has to be built.

The domain-specific dictionary that was built does not contain only the words that are missing in the morphological dictionary, but it contains all words that occur in the above mentioned domain. All words in the domain-specific dictionary were indicated as domain-specific words that manifest themselves as keywords in questions. Words that are not in the morphological dictionary can be added to the domain-specific dictionary in whatever form with the purpose to of expanding it. Regardless of their form, series of words can be added to the dictionary. The connection between the morphological and the closed-domain dictionary enables the retrieval of all forms of series of words. This is possible for those series of words of which words occur in the morphological dictionary. The domain-specific dictionary also contains the information on the meaning of words or series of words as well as the information on their relationship. The meaning of words tells the difference between different persons, organizations, objects, etc. The synonyms, e.g. degree dissertation – diploma thesis, and acronyms, e.g. ECTS - European Credit Transfer System are very frequent. The relationships between words enable the transformation of questions, e.g. “What is ECTS?” into “What is the European Credit Transfer System?” and vice versa.

The keywords of questions that are in the database or have been imported from the columns of the tables in local databases or MS Excel files provide the source of words for the domain-specific dictionary.

4.2 Application for users

Prior to the implementation of the application for users, we had to determine the most important properties of the system. The most important properties are:

- Reliability. The system must provide an answer to the exact question. In order to provide the correct answer, it has to be capable of assessing whether or not there is enough information available to answer a particular question. If there is not enough information, the system must react in the most appropriate way. For example, the system is obliged to inform the user about the lack of information, or that it is not capable of providing an answer.

- Response time. If the system is able to provide the answer faster than a user can find it by clicking on the web portal or with the help of a search engine, its existence is justifiable. The response time, however, does not depend merely on the system itself but also on the user. It is of the utmost importance that a user is capable of requesting certain information in the appropriate way.

The goal of the project was to develop a fast and reliable system. Unfortunately, these two properties are contradictory. In order to improve the accuracy of the answers, a more complicated search algorithm which prolongs the response time of the system is required. On the other hand, in order to shorten the response time of the system, the search algorithm must be simpler, thus leading to the inaccuracy of answers. Higher priority was given to the accuracy of answers. The response time, however, should still be as short as possible; therefore we sought to achieve the optimal combination of both properties.

As already mentioned, the system responds to certain types of questions: insulting questions, special questions or phrases, questions referring to the employees of the faculty, template based questions which consist of the data stored in local databases and MS Excel files, and other questions.

Such classification was chosen due to the information sources and approaches that are essential for the processing of questions.

In order to reach the optimal accuracy and speed of the system, the algorithm processes questions according to the above sequence of question types, starting with the insulting questions. This enables extremely short response times in case of any occurrence of insulting questions or phrases.

Two tasks run in parallel in the system: the question classification and the answer retrieval itself.

The question classification procedure is needed to classify the questions into one or more semantic categories (classes). The classification of questions is similar to the more known are of content classification [13]. Although the problems are similar, significant differences exist. While content classification can rule out the “empy” words (stop words) such as “what” or “is” they are very important to the question classification process.

As mentioned question classification classifies a question to one or more semantic classes. The number of classes varies among different systems. Some approaches [14], [15] use a two-level taxonomy where the first level (coarse grained) is used to classify the general area and the second (fine grained) to provide a more specific classification. Each first level class uses a unique set of second level classes. Because of the manual effort needed for a complex taxonomy, most QA system limit themselves to less than 20 first level classes.

In our system the first level classes consist of six different classes: entity, location, human, numeric value, description and abbreviation. These classes are further separated with level two classes as represented in Table 1.

Table 1: Coarse and fine grained classes

Coarse	Fine
ENTITY	subject, form, application pettition, scholarship, literature, doctor, event, food, sport, vehicle, computer equipment, class
LOCATION	city, country, lecture room, building
HUMAN	description, group, individual, title
NUMERIC VALUE	count, date, money, order,percent, period, mark
DESCRIPTION	definition, description, reason
ABBREVIATION	abbreviation, expansion

Although heuristic rules can be produced manually, the process would require a significant amount of time and effort. Therefore a decision was taken to use machine learning approaches to develop a highly capable program for classification. A beneficiary factor for the use of machine learning was also the fact, that the resulting program is flexible to changes; therefore it can be easily adapted to a new domain.

Training and testing set were prepared, consisting of questions that were gathered during the phase of problem analysis and all of the questions that were proposed to the system.

Probably the most important task of every classifier is the extraction of appropriate features (feature selection) [16], because better features ensure a more accurate question representation and lead to better results. Selected features are used for the construction of a feature vector, which is the basis for learning. Most commonly used is a small set of syntactic and semantic features. The use of syntactic features in question classification has been thoroughly studied. The semantic features have, with the exception of WordNet and named entity recognition, yet to be thoroughly researched. On top of the primitive features (words, named entities, part-of-speech tags) a variety of operators can be used and together they provide more complex features (relation features [17]).

We have chosen these features to be a bag of words consisting of all the words in the question, k-grams (sorted sequence of k-words), the question main pronouns (what, where), length of the question, POS tags (part-of-speech tags) and the question headword.

Question classification with machine learning is not well researched. Majority of authors have used these methods for classification; supervised learning in the form of: decision trees and rules, Bayes classifiers, nearest neighbor classifiers, discriminatory functions (Support Vector Machine; SVM), artificial neuron nets and hybrid approaches. We have decided on SVM, since this approach has proven to be the most appropriate [15],[18],[19].

As previously mentioned, the process of answer retrieval is conducted in parallel to the question classification. One part of the question processing procedure is the same for all questions. First of all, all scripts and HTML elements are eliminated in order to disable any security breaches via input form of the application for users [20]. Punctuation marks, special characters and emoticons are eliminated as well, since these elements of language do not influence the quality of answers at this point of the development of the system.

Before the system starts to examine if the question that was asked belongs to a certain type of questions written above, it determines if the question has already been asked in the past and if the correct answer was provided. If the system retrieves the answer in the described way, the further search is terminated, otherwise it continues.

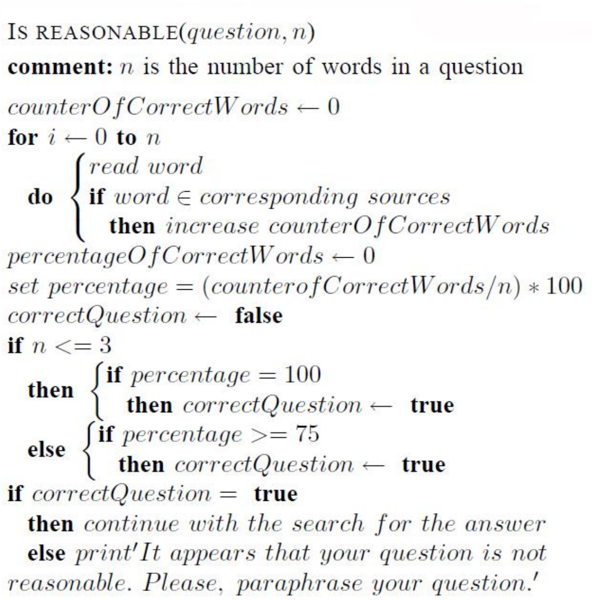
First of all, the system determines whether the question is reasonable or not; therefore a simple algorithm is required. The pseudo code of the algorithm can be seen in Fig. 3.

If the question is reasonable, the system determines to which category of questions it belongs.

The system starts with the category of insulting questions. Every word in a question is examined and compared to the list of the insulting words which was built in order to be used by the application. As soon as an insulting word is found, the system terminates the processing of the question and informs the user of its inappropriateness.

If the system determines that a question does not belong to the category of insulting questions, it tries to determine if it belongs to the category of special questions. Answers to special questions are retrieved with the help of full matching (the question matches all characters). If a question meets the criteria of a special question, it must be determined whether there are more possible answers to that question. If there are, the system chooses one of the possible answers and finishes the search. The variety of possible answers should increase the interest of users and improve their confidence in the system.

If a question does not belong to the category of special questions, the system continues to search for an answer. In the next step, the system determines which words in the question are keywords. A special algorithm is required in order to determine them. Before the description of the algorithm one must bear in mind that a keyword is not necessarily one word, but it can also be a series of words.



```
IS REASONABLE($question, n$)
comment: $n$ is the number of words in a question
$counterOfCorrectWords \leftarrow 0$
for $i \leftarrow 0$ to $n$
    do  read word
        if $word \in corresponding\ sources$
            then increase $counterOfCorrectWords$
$percentageOfCorrectWords \leftarrow 0$
set $percentage = (counterofCorrectWords/n) * 100$
$correctQuestion \leftarrow$ false
if $n <= 3$
    then  if $percentage = 100$
              then $correctQuestion \leftarrow$ true
    else  if $percentage >= 75$
              then $correctQuestion \leftarrow$ true
if $correctQuestion =$ true
    then continue with the search for the answer
    else print'It appears that your question is not
reasonable. Please, paraphrase your question.'
```

Fig. 3: Algorithm

The representation of the algorithm is based on the question “When is the exam period?” The words in the question are: *when*, *is*, *the*, *exam*, *period*. In the beginning, the system processes the whole series of words, as it would be one keyword. If the series of words represent a keyword, the system memorizes it and continues with the search. In the next step, the system eliminates the last word in the sequence and determines whether the new series of words represent a keyword. If it does, the system memorizes it and continues with the search. The system repeats the procedure until there are no more words left in the series. In the described example all the keywords are written in bold.

1. complete series: when is the exam period
    1. keyword: when is the exam period
    2. keyword: when is the exam
    3. keyword: when is the
    4. keyword: when is
    5. keyword: when

Because none of the above words can be found in the domain-specific dictionary, none is written in bold.

As the process continues, the first word in the original series of words is eliminated. The new series of words is then processed in the same way as the original.

2. complete series without the first word: is the exam period
    1. keyword: is the exam period
    2. keyword: is the exam
    3. keyword: is the
    4. keyword: is
3. complete series without first two words: the exam period
    1. keyword: the exam period
    2. keyword: the exam
    3. keyword: the
4. complete series without first three words: exam period
    1. keyword: **exam period**
    2. keyword: **exam**
5. complete series without first four words: period
    1. keyword: **period**

During the search, the following keywords were found: *exam*, *period*, *exam period*.

In order to increase the number of keywords and improve the variety of the search, the system continues searching for the keywords even if they have already been found.

When the system stops searching, it determines if any of them stands for a name of a certain employee of the faculty. In the database there is a table with the information on the employees of the faculty. The first and last names of the employees are in their lemmatised form, but in questions they can also be declined.

For that particular reason it is of the utmost importance that the database contains all possible declination forms. Since the morphological dictionary does not contain all declination forms of names, we developed a tool that is capable of declining the 5000 most common Slovene proper names.

Because Slovene proper names that end in the same way are declined according to the same declination pattern, they could be divided into three categories: *feminine names*, *masculine names*, and *last names*. Later, the declination rules for individual groups of names were determined and implemented into the system.

The declination of female names can be formally represented with just two rules. On the contrary the male names are a significantly larger problem. The base rules, which are defined with the ending character of the name or surname, are divided to another level of rules based on the two last characters. This rule division process goes up to four levels deep and is additionally supplemented with exceptions to the rules, which are used separately.

All possible declination forms of the names of the faculty's employees are currently stored in two tables. The first one contains the standard (lemmatized), the second all the declination forms of names. The number of employees, however, can change on a daily basis, therefore all three aforementioned tables are daily updated.

An example will illustrate the functioning of the system. If a question contains the name of a faculty employee and this name is not one of the names *Rok*, *Samo*, *Rado* and *Avgust*, the system starts searching for an additional keyword in order to determine whether the user asked for specific information on a certain employee.

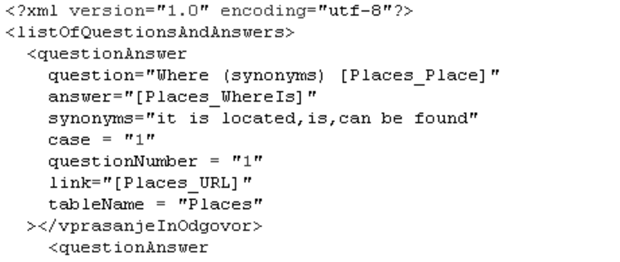
If such a keyword is found, the specific information on a certain employee is presented to the user in the first sentence of the answer, and the following sentences contain general information on the employee.

If such a keyword cannot be found, the answer contains only general information on the employee which is represented to the user in the default sequence.

Slovene proper names Rok, Samo, Avgust, and Rado must be processed separately. The aforementioned Slovene words can carry two meanings. They can be understood either as proper names, or they can carry a general meaning. As words that carry a general meaning they are also a part of the morphological dictionary. Consequently, the system has to determine whether a certain question refers to a person or not. The following example serves as an illustration. The word *avgust* in Slovene can refer either to a person (a proper name) or to a month. The questions "Where can I find **Avgust?**" and "When are the office hours of the student office in August?" (Kdaj so **avgusta** uradne ure referata?) contain the same word which in the first question refers to a person and in the second to a month. Based on the context people are able to distinguish between the two referring objects instantly. For computers, however, such deduction is a complicated task which they have to learn. Because the system in its current phase of development was not able to deduct in such a way the necessary mechanism was not planned even in the beginning a temporary solution had to be found. If one takes a closer look at the above examples, one might think that the solution lies in the capital letter. If the word *avgust* refers to a person, it must capitalized, however, it is not realistic to expect that users will use the capital letter according to the rules of grammar, therefore it is not an appropriate solution. An appropriate solution is to require additional information. For example, if a user asks "Where is Avgust?" or "Where can I find Avgust?", the system inquires whose Avgust the user might have in mind. Otherwise, the system considers the word to be a general expression (it refers to a month) and continues to search for an answer among the general questions. The same procedure applies for the other above mentioned proper names.

The following section of the article describes the retrieval of answers which use data from local databases including the databases of which content was transferred from MS Excel files. The questions are described with the help of question templates that cover the conceptual model of the database and describe the concepts, their attributes, and the relationships [21].

A question template is actually a question with empty slots for data instances that represent the main concepts of the question. For example, "Where is <place>?" is a question template where <place> represents an empty slot that belongs to the concept of place. If this slot is filled up with data instances that belong to the concept, an ordinary question, e.g. "Where is the Gama lecture hall?", is formed. The templates are written in the form of an XML file [22]. An example can be seen in Fig. 4.



```
<?xml version="1.0" encoding="utf-8"?>
<listOfQuestionsAndAnswers>
  <questionAnswer
    question="Where (synonyms) [Places_Place]"
    answer="[Places_WhereIs]"
    synonyms="it is located,is,can be found"
    case = "1"
    questionNumber = "1"
    link="[Places_URL]"
    tableName = "Places"
  ></vprasanjeInOdgovor>
    <questionAnswer
    question="What (synonyms) [PriceList_Item]"
    answer="[PriceList_Item] costs [PriceList_Currency]"
    synonyms="is the price,are the costs of"
    case = "1"
    questionNumber = "2"
    link=""
    tableName = "PriceList"
  ></questionAnswer>
</listOfQuestionsAndAnswers>
```

Fig. 4: An example of the XML template for questions that can be answered

Question template in the XML file is written as an attribute of the element *questionAnswer*. Empty slots that can be filled up with the examples of concepts are marked with square brackets "[]". In order to use the same template also for questions that refer to the same specific concept but have a different form, round brackets were introduced "()". Round brackets represent an empty slot that can be filled up with the attribute values of a *synonym*. In this way the capability of a template is increased. Each template can have more empty slots for concepts and attribute values of synonyms. In this case the values of individual synonym slots are separated with the semi colon ";", and the capability of a template is increased even further.

The number of questions that can be formed on the basis of one template is calculated with the help of the following equation:

$$N = \prod_{k=1}^{n} k * \prod_{j=1}^{m} s \tag{1}$$

in which k is the number of all instances of an individual concept, and s is the number of all instances that can fill up individual slots of synonyms.

In order to form an answer, the system also uses templates that are written in the attribute *answer* in the above XML file. In order to retrieve an answer, the system forms all possible questions based on the individual descriptions from the XML. The original question is then compared to the series of potential questions. Because Slovene is an inflectional language, questions must be lemmatised before the comparison. If two lemmatised questions match, the answer is formed on the basis of the description in the XML and later presented to the user.

To enable finding the answer, the element *questionAnswer* was extended with a *questionNumber* element. The element provides the template identifier. It is added to every automatically generated question in order to link them to their corresponding answer template. Also the *questionAnswer* element was additionally extended with the *tableName* element. Also a part of the answer can be in the form of a HTTP link, which is stored in the *link* attribute.

The formation of an answer is again a difficult task, especially due to the inflectional Slovene language. Empty slots cannot be simply filled up with the data (series of words) from the database, but they must be in the correct form. The correct form of the series which is based on the value of the attribute "case" is provided by the morphological dictionary of the

Slovene language. It has to be taken into consideration that not all forms of words in the series are changed. It is difficult to determine which word in the series changes. Answers that refer to people must also be written in the appropriate form considering the gender of the person, to which they refer. This means that the endings of certain words in a sentence have to change, and it is difficult to automatically determine which ones. The above mentioned difficulties represent a challenge for our future work. If the system finds the answer in such a way, the further search is terminated, otherwise it is continued.

In the last step, the system processes the remaining questions. The main idea of question processing is to gradually decrease the number of potential questions questions of which answers could be the answer to the user's question in order to find a question that is most similar to the user's question.

In our case those are the questions that contain at least one of the keywords in the user's question. The system memorizes their unique identifiers. In order to speed up the system, the keywords of questions in the database are prepared in advance.

During the process, it is determined how many keywords are necessary in order to define a particular question from the database as a potential question. If the user's question contains one or two keywords, a potential question must also contain one or two keywords. The questions must be a 100% match considering the keywords.

The reason the questions must be of the same number of keywords will be presented in the following example. In the question "When does the exam start?" there are two keywords: *exam* and *start*. Because the word *exam* is very frequent in the domain of the faculty, the system will find 150 potential questions containing the keyword *exam* and 45 questions containing the keyword *start*. It will have to process 195 questions or 194 questions if it is presumed that there is a question which contains both keywords. If it is also taken into consideration that all keywords in questions with a small number of keywords are of greater importance than the keywords in questions with a large number of keywords, both keywords are of the utmost importance. If in the case of two keywords we demand that a potential question must also contain both keywords, there is only one question left in the database for the system to process. In such a way, the number of potential questions and the search time are considerably diminished.

The importance of keywords diminishes with their number; therefore the matching percentage of keywords in questions that contain three or more keywords was lowered to 75%. For example, if the original question has four keywords, the potential questions from the database must have at least three matching keywords. In this case three keywords are enough to determine that two questions are similar. The one keyword that differs allows the user to form a question in different ways or include redundant information. Despite that fact, the system still answers correctly. From all potential questions the system selects only those that are at least a 75% match of the original question considering the keywords. The system then calculates the matching percentage for each appropriate question and determines the highest. All questions with the highest matching percentage are later processed by the system.

For each potential question, the difference between the number of keywords in a potential question and the user's question is also calculated, and the minimum is determined. Questions that are below the minimum are eliminated. If there is only one potential question, then this is the only candidate question and the answer to it will be presented to the user.

If there are more potential questions, the question forms of the potential question and the user's question are compared. For example, the questions "Who is the dean?" and "Where is the dean?" differ only in their question forms. If the question form of a potential question corresponds to the question form of the users question, the answer to the potential question in the database is delivered to the user. If the question forms do not match, the system selects another potential question and compares the question forms anew. This procedure is repeated until the last question but one. If none of the question forms match, the answer to the last potential question will be presented to the user. In this procedure we eliminate all but one candidate answer.

Before it is presented to the user we check the question type. If for example the question type is determined as "amount" we need to verify that the answer refers to amount. This is done with the determination of the headword of the answer. If the answer is in the correct type it is presented to the user.

The answers can contain also one or more hyperlinks to web pages or documents with additional information. The first hyperlink opens in a parent window. The parent window is the window in which the application for users was started; the user application itself runs as a popup window.

If the users are authenticated (login procedure) the system also uses available information from the user profiles. This results in personalized answers to questions like “What exams are today?”. The answer given refers to the exams the are available for the user and not all of the exams on campus today. The information is gathered during registration procedures (first name, last name, date of birth), obtained from other faculty systems or direct dialog with the user.

The system keeps track of every answer given to every user, including unknown users (not authenticated). These answers are gathered with the intention of improving the system performance and a feedback information to the administrators of the system on the areas of interest to the users that are not included in the system.

If an answer cannot be found, or if the question does not contain enough keywords, the system asks the user to paraphrase it and provide additional information; otherwise the system forwards the keywords to the search engine on the web portal which then tries to find the documents containing these keywords. The search engine is presented in greater detail in [23].

5 Conclusion

The article describes the Slovene language question answering system. The system retrieves the answers from structured and unstructured information sources. A small portion of answers is automatically formed on the basis of templates. The system is capable of forming a simple dialog with the user for purposes of information retrieval that enables better answers.

The system can also provide answers to the questions referring to the functioning of the system and respond to aggressive behavior of users. The system is in use on the web portal of the faculty. The development of this system has enabled faster access to information for students and alleviated the burden on the employees that were answering students’ question in the past. A new method of information retrieval, question answering, has been presented in our target domain. The use of the Slovene language in the system is highly innovative. The fact that the Slovene language is an inflectional language is reflected in the difficulty of its processing. During the development of the system, some solutions that are necessary for the processing of Slovene language have been developed. Some solutions, however, were only foreseen due to their extensiveness and represent a challenge for future work.

A challenge for the future work is the transition from a domain specific to a general (open domain) question answering system and the introduction of semantic web features [24], [25].

References:

[1] L. Hirschman and R. Gaizauskas, Natural language question answering: The view from here, *Natural Language Engineering*, Vol. 7, No. 4, 2001, pp. 275-300.

[2] J. Hendler, Agents and the Semantic Web, *IEEE Intelligent System*, Vol. 16, No. 2, 2001, pp. 30-27.

[3] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American*, 2001, pp. 35-43.

[4] N. Shadbolt, W. Hall and T. Berners-Lee, The Semantic Web Revisited, *IEEE Intelligent Systems*, Vol. 21, No. 3, 2006, pp. 96-101.

[5] A. Gómez-Pérez, M. Fernández-López and O. Corcho, *Ontological Engineering*, Springer-Verlag New York, 2003.

[6] D. L. McGuinness, Question Answering in the Semnatic Web, *IEEE Intelligent Systems*, Vol. 19, No. 1, 200, pp. 82-85.

[7] Š. Arhar and M. Romih, Klepec: Programirani sogovornik za slovenščino, *Proceedings of 5th Slovenian and 1st international Language Technologies Conference 2006, 2006.*

[8] A. De Angeli, R. Carpenter, Stupid computer! Abuse and social indentities, *Jabberwacky AI News*, September 2005.

[9] L. N. Foner, Entertaining Agents: A Sociological Case Study, *Proceedings of the first international conference on Autonomous agents: Marina del Ray*, 1997, pp. 122-129.

[10] J. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz and D. Karger, What makes a good answer? The role of context in question answering, *Proceddings of the Ninth IFIP TC13 International Confernce on Human-Computer Interaction (INTERACT 2003)*, 2003, pp. 25-32.

[11] D. Mladenic, Automatic Word Lemmatization, *Proceedings B of the 5th International Multi-Conference Language Technologies*, 2002, pp. 153-159.

[12] T. Erjavec, S. Džeroski, Machine learning of morphosyntactic structure: Lemmatizing unknown slovene words, *Applied Artificial Intelligence*, Vol. 18, No. 1, 2004, pp. 17-41.

[13] L. N. Foner, Entertaining Agents: Text Classification Using Machine Learning Techniques, *WSEAS Transactions on Computers*, Issue 8, Vol. 4, 2005, pp. 966-974.

[14] S. X. Li, D. Roth, Learning Question Classifier, *Proceedings of the 19th International*

Conference on Computational Linguistic, 2002, pp. 1-7.

[15] D. Zhang, W. S. Lee: Question Classification using Support Vector Machines, *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 26-32.

[16] D. Metzler, W. B. Croft, Analysis of Statistical Question Classification for Fact-based Questions, *Information Retrieval*, Vol. 8, No. 3, 2005, pp. 481-504.

[17] X. Li, D. Roth: Learning Question Classifiers, *Proceedings of the 19th International Conference on Computational Linguistic*, 2002, pp. 1-7.

[18] S. Cruchet, A. Gaudinat, C. Boyer: Supervised approach to recognize question type in a QA system for Health, *Studies in health technology and informatics (Stud Health Technol Inform)*, Vol. 136, 2008, pp. 407-412.

[19] H. Yu, C. Sable, H. R. Zhu: Classifying Medical Questions based on an Evidence Taxonomy, *American Association for Artificial Intelligence (AAAI'05) Workshop on Question Answering in Restricted Domains*, 2005.

[20] J. D. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla and A. Murukan, *Improving web application security*, Microsoft, 2003.

[21] E. Sneiders, Automated Question Answering Using Question Templates That Cover the Conceptual Model of the Database, *Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers*, 2002, pp. 235-239.

[22] S. Holzner, *Inside XML*, New Riders, Indianapolis, 2001.

[23] S. Pohorec, M. Verlič, M. Zorman, Domain specific information retrieval system, *Proceedings of the 13th WSEAS international conference on computers (part of the 13th WSEAS CSCC multiconference)*, July 2009, pp. 502-508.

[24] M. Stanojević, S. Vraneš: Semantic Approach to Knowledge Processing, *WSEAS Transactions on Information Science and Applications*, Vol. 5, Issue 6, 2008.

[25] S. Yang, C. Hsu, D. Lee, L. Deng: FAQ-master: An ontological Multi-Agent System for Web FAQ Services, *WSEAS Transactions on Information Science and Applications*, Issue 3, Vol. 5, 2008.