Medicine.Ask

João Pedro Videira Bastos

Instituto Superior Técnico

Abstract. There is more and more digitalized information in hospitals. The better this information is used, the better the hospital will work. This paper presents a system for information extraction, in medicine, which allows the doctor to obtain information about medicines, using natural language. It is intended that the system collect information on medicines from a database online, with techniques commonly used to extract information. This information will be stored in a knowledge base that will provide answers to questions raised by doctors. For example, the doctor may ask the system "Medicines for eye infections?". Note that the purpose of construction of this system is to assist the doctor in their function and not to replace it.

Key words: Information Extraction, Natural Language, Database, Interface

1 Introdution

In order to prescribe a medicine, either a Doctor already knows all the details about it (such as its name, posology and contra-indications) or he/she needs to spend some time searching for that information. Several books are at the Doctor's disposal (e.g., the Portuguese Therapeutic Symposium), as well as Web sites, such as the electronic Medicine Compendium, which contains information about UK licensed medicines (http://emc.medicines.org.uk/). Moreover, independent software packages with extra search capabilities can be used in the Doctor's Pocket PC or mobile. Nevertheless, despite all the available information, the provided search functionalities are usually based in keywords or class-oriented (allowing, for instance, a search by laboratory or by ATC classification.

With the goal of speeding up the prescription process, we propose a system called Medicine.Ask, a system that allows to search for information about medicines, through a (controlled) set of questions posed in Natural Language, such as "Which are the medicines for influenza that can be used during pregnancy?". Building Medicine.Ask encloses the following sub-tasks: (i) to understand the doctors' prescribing needs; (ii) to design a relational database that will store information about medicines; (iii) to extract information about medicines from the Web and load it into the database; and (iv) to implement a natural language interface (for Portuguese).

In order to accomplish sub-task (i), Health professionals were asked about relevant questions in the prescription process. These questions, as well as a de-

tailed analysis of the information available in the Infarmed site[1], an on-line Web site of information about medicines in Portuguese, led to the design database scheme addressed in sub-task (ii). The database was built in MySQL, an open source database management system. Then, information from the Infarmed site was extracted and processed to populate the database, thus accomplishing sub-task (iii). To extract information from the site, we used the Web Harvest software package. It supports querying XML nodes through techniques such as XPath, XQuery and Regular Expressions. Besides these techniques, the information extraction step also required the manual annotation of medicine information such as contraindications, indications and adverse reactions. Finally, sub-task (iv) consisted in the creation of a set of template questions that the system knows how to map into SQL queries to be posed against the database of medicine information. The technology used to create the interface was JSP.

This paper is organized as follows:

- In section 2 is described a survey about information extraction systems in medicine.
- The section 3 explains de architecture of the system implemented.
- The section 4 shows an example of the system.
- The section 5 evaluates the system.
- At least, in section 6 the conclusions are shown.

2 Survey

There are many systems built with the aim of extracting information about medicine. In this thesis some of these systems were analyzed. Some ideas are described here to a better comparison. A comparison between the systems is shown in Table 1 that shows the tasks that each system implements. In Table 1, we can conclude two things about the presented systems.

- few systems have implemented the task of pre-processing text.
- systems are all focused on the classification task.

Table 1. *The tasks that each system implements*

Systems	Pre-processing	Classification	Association	Interp. of discourse
Proteus-Bio	No	Yes	No	Yes
MedLEE	No	Yes	Yes	Yes
BioAnnotator	No	Yes	No	No
PASTA	Yes	Yes	Yes	Yes
System PVP	Yes	Yes	No	Yes
MedsynDiKate	No	Yes	Yes	Yes

[1] http://www.infarmed.pt/

A detailed description of these systems can be found in the following papers: Proteus-Bio (Grishman, Huttunen, & Yangarber, 2002), MedLEE (Friedman, Alderson, Austin, Cimino, & Jonhson, 1994), BioAnnotator (Subramaniam, Mukherjea, Kankar, & Srivastava, n.d.), PASTA (Gaizauskas, Demetriou, Artymiuk, & Willett, 2003), System PVP (Ferreira, Teixeira, & Cunha, 2008) and MedsynDiKate (Hahn, Romacker, & Schulz, 2002).

Table 2 shows the ratings of the analyzed systems, but is a subjective comparison, since each system belongs to a different project, with a different corpus and with different developing levels, so it can't be compared through measures that normally were used to evaluate information extraction systems.

Table 2. *Evaluation of the systems presented*

Systems	Recall	Precision	F-measure	Scope
Proteus-Bio	41%	79%	n.d.	diseases outbreaks
MedLEE	70%	87%	n.d.	diseases description
BioAnnotator	60,3%	68,6%	64%	biologic terms
PASTA	82%	84%	n.d.	amino-acids
Sistema PVP	99%	100%	99%	vaccination
MedsynDiKate	85%	81%	n.d.	diseases description

3 Systems architecture

Medicine.Ask is composed by two distinct modules. The first one is responsible for creating the knowledge base as well as the entire phase of information extraction; the second is responsible for the interface in Natural Language. Figure 1 shows the system architecture.

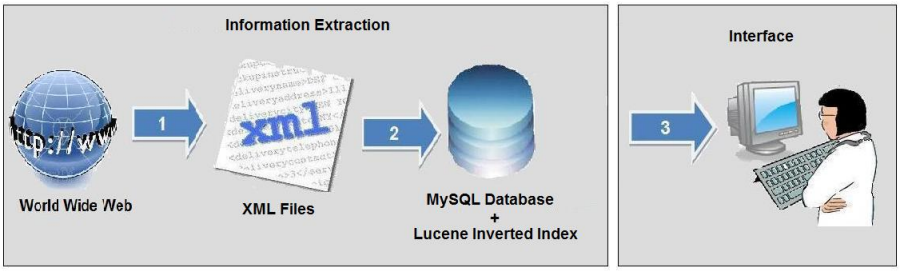

Fig. 1. *Medicine.Ask architecture*

In the beginning of the project questionnaires were given to health professionals, in order to get questions templates that they would ask to the system. The

first module includes the phase information extraction that extracts the information from web and segments it, producing two types of information: structured and unstructured. The structured information is the information that is not necessary to process after the extraction and is therefore ready to fill the database. The unstructured data represents information that needs to be treated before insertion into the database.

The data source chosen to extract the information was Infarmed. The information extraction was done with the use of: technology XPath to navigate in the html code and to extract information, regular expressions to eliminate code layout and XQuery to produce XML documents containing the extracted information to be processed later. The assessment of this module was done manually and all the xml files were created with information from the page Infarmed.

Finishing the first module we built up a knowledge base. This step was divided into two phases: first we built up a database with structured information and then we created inverted indexes or to index the unstructured information. The structured information is the information of medicines, substances that have medicines and the chapters that have the substances. This information was processed directly from the xml files created in previous step for the database. The evaluation of this phase resulted in 100% of accuracy as all the information was passed to the database.

The unstructured information was indexed in two ways: with inverted indexes and by using machine learning. A lot of machine learning techniques were tested, to index all the information of the test substances, in particular, indications, adverse reactions, interactions, precautions, and dosage. This information is different because it is structured in Natural Language text while the structure information is in tables. As so, it is necessary to index this information so that when looking for some information, for example, a symptom, it is easier to find it. In the case of dosage the used technique were the SVMs (Support Vector Machines) with the assessment of 77% precision and recall. In this case, it worked well because the text of the dosage had some features that allow a learning process. These features are that the text was divided into three parts (adults, children and all), and in the beginning of each part was the information if that part was for adults or children, or for all. For the other categories, these techniques were not so successful. The best was the CRFs (Conditional Random Fields) with 38% for recall and 59% for precision. So for the other categories were created inverted indices that eventually showed very good performance for indexing long texts.

The interface is the visible part of the system as it, allows the user to perform the research tasks needed. The long term goal of this interface is to accept a set of questions in natural language for the user to freely formulate his/her question. To this end, it is necessary that the interface can interpret the questions that it will receive. Thus, we built a parser that interprets templates of questions. The analysis of the questions was obtained from the questionnaires made. 20 questions were retrieved from questionnaires that were organized into 11 different templates.

Finally, in order to help the user in his/her search, 5 algorithms were used to help search:

- **soundex** is algorithm that is based on the phonetics of words. It was implemented for when a user cannot write a name of a medicine, but knows how it sounds. So, write as he/she thinks it is the algorithm and suggests the right medicine.
- **detection of incomplete names** an algorithm that uses the SQL LIKE. The aim is to help the user to reach the right medicine. The user can know the beginning of the name, but doesn't know the end. If he/she only writes the beginning this algorithm detects these situations and suggests the right medicine.
- **links addiction** with the goal to accelerate research. When, for example, the user search for something and the medicines are shown. These medicines are shown as links for the user can get a rapid access to the information of this medicine. Avoiding the user repeats the search.
- **addition of related research** with the goal to present suggestions to the user search. When, for example, the user search for indications of voltaren, the interface suggests the adverse reactions, contraindications, precautions and dosage.
- **addition of suggestions** with the goal to help in the search. When the user is writing the question in the text box, the interface will indicate possible questions to the user based on what he/she is writing.

4 Example

The interface have a text box where the user puts the question, as illustrated in Figure 2. The answers to questions may be presented in a table if the information requested is on drugs, or text if the information requested is about substances.

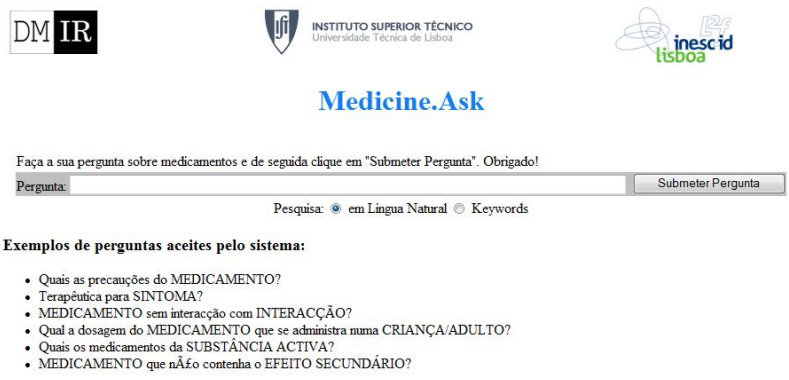

Fig. 2. *System interface*

When the user is writing a question, this will give suggestions based on what he/she is writing. Figure 3, illustrates an example of how auto-complete works.

Fig. 3. *Auto-complete*

If the user makes a mistake to write the name, the interface suggests alternatives, taking into account what the user wrote. If the user typed the name as it sounds, the algorithm that goes into action is the **soundex**, if was written only the beginning or just the end of the name, the algorithm that is called is **detection of incomplete names**. Figure 4, shows that the result is shown in a table. If the user wrote, for instance voltarn instead of voltaren the interface, using the **soundex**, suggested voltaren. It shows also that the names of medicines are links that can be loaded to go directly into the product page.

5 Evaluation of the solution

This section describes the tests that were performed to evaluate the system described in previous sections. The evaluation of the system consists of multiple users interacting with the system in order to obtain answers to questions about medicines. The selected users were health professionals, because the Medicine.Ask is directed to them.

The evaluation consisted of two different scenarios in order to observe some differences between the two groups, as well as in their opinions. The scenarios consisted of the users seeking drugs with some clues in the first 5 questions and no clues in the last 5. The first scenario corresponded interact with the page Infarmed submitting 10 questions. After interacting with the Infarmed, the user submitted the same 10 questions to Medicine.Ask.

The second scenario is similar to the first, but you can only interact with the Medicine.Ask.

Fig. 4. *Soundex*

Were submitted to the interface 124 questions. Being 10 questions per user and 10 users, should be 100 questions submitted. However, there were two situations that lead to this: the answer was not answered and the user reworded the question, and the fact that the users become enthusiastic with the interface and submitted more.

Of the 124 questions submitted, 69 were placed in Natural Language and 55 were made by keywords. As noted were placed more Natural Language, which supports the decision to have made a Natural Language Interface.

Of the 69 questions submitted to Natural Language, 57 were answered and 12 got no response. There were two main reasons for this. The fact that Natural Language is very open and a user can ask something in different ways. For example, the question type is "Medicines reimbursed the flu shot" and the question was submitted reimbursed "Vaccines against the flu?". The other reason was that there is no information on the question referred. For example, the question "h1n1?". The 55 questions submitted by keywords had always correct answer.

While users interact with the interface there was always someone to observe and record what they did. So it might take information about the algorithms used during the study.

- For the functionality of suggestions there were 40 searches that used it,
- About the functionality of detection incomplete names, 8 questions used it,
- With regard to the soundex, 5 of the questions submitted use it,
- Finally, although users are all health professionals, and though they were familiar with the names of medicines, only 6 issues of the keywords did not use any of the previous algorithms.

There was also the time and clicks that users spent to interact with the system. It was found that users interact fast with Medicine.Ask and did so with fewer clicks. We compared the times of the two groups to see if there was learning

by the fact that the first were interacting with the Infarmed, what didn't happen because the times were the same. Finally, we compared the differences of time and clicks between the first 5 questions with the last 5 to see which system is more intuitive. In Infarmed time and clicks remained, but in Medicine.Ask decreased, which indicates learning.

6 Conclusion

Obviously, in construction of any system, it should have all the information about the area. The first phase of the data extracting from web proved a success, because we were able to extract the information of the page Infarmed with values of 100%.

After the data was extracted, it is essential that the data is indexed and organized so that the interface can use it. Moreover, it is necessary to transform data into useful information. With regard to structured information, all existing information in the XML documents passed to the database. The unstructured information, it was more difficulty to index. Of all the categories, the only index that has been achieved using machine learning techniques was the dosage, with values close to 76%. This indicates that there is scope to improve this step, as well as in other categories. The solution to index the rest was inverted indexes, which show good performance when indexing text in Natural Language.

After building the knowledge base, we developed a Natural Language Interface to answer the questions about drugs. Overall the interface was well accepted by users. Compared with the Infarmed, this interface is easier to use and more intuitive, because the time and clicks to get the response are lower, and the adaptation of the user is very quickly. It is essential that a new system is easy, fast and effective way to achieve the same results as the previous systems. But even with these results and with a more ambitious perspective, this is the module that needs more development, because as is normal in a Natural Language interface there is always room for evolution.

References

Ferreira, L., Teixeira, A., & Cunha, J. P. S. (2008). Ontology-driven vaccination information extraction. In *Nlpcs* (p. 94-103).

Friedman, C., Alderson, P. O., Austin, J. H. M., Cimino, J. J., & Jonhson, S. B. (1994). A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, *1*(2).

Gaizauskas, R., Demetriou, G., Artymiuk, P. J., & Willett, P. (2003, January). Protein structures and information extraction from biological texts: the pasta system. *Bioinformatics*, *19*(1), 135–143.

Grishman, R., Huttunen, S., & Yangarber, R. (2002). Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, *35*, 236–246.

Hahn, U., Romacker, M., & Schulz, S. (2002). Medsyndikate–a natural language system for the extraction of medical information from findings reports. In (p. 63-74).

Subramaniam, L. V., Mukherjea, S., Kankar, P., & Srivastava, B. (n.d.). Information extraction from biomedical literature: Methodology, evaluation and an application. *Conference on Information and Knowledge Management*, 410 - 417.