Medicine.Ask: a Natural Language Search System for Medicine Information

Helena Galhardas, Vasco D. Mendes, Luísa Coheur

Technical University of Lisbon and INESC-ID
Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
helena.galhardas@ist.utl.pt vascodmendes@hotmail.com
luisa.coheur@inesc-id.pt

Abstract. Health personnel deals with medicines on a daily basis. They need to have access to comprehensive information about medicines as fast as possible. Several books and websites are at their disposal, as well as independent software packages with extra search capabilities that can be used in Pocket PCs or mobiles. The public, in general, is also interested in quickly accessing information about medicines. Despite all the electronic possibilities available nowadays, the search functionalities provided are usually based on keywords or class-oriented (allowing, for instance, a search by laboratory or by ATC classification). We propose Medicine.Ask which is a question-answering system about medicines that couples state of the art techniques in Information Extraction and Natural Language Processing. It supplies information about medicines through a (controlled) set of questions posed in Natural Language (Portuguese). An example of such a question is: "*Which are the medicines for influenza that can be used during pregnancy?*". We claim that Medicine.Ask is easier to use and that successful answers can be obtained in a shorter amount of time than with the search mechanisms available in the INFARMED "Prontuário Terapêutico" website. In this paper, we present the architecture of the system and the main techniques used. Furthermore, we report the results obtained when comparing the system with the INFARMED "Prontuário Terapêutico" website.

Keywords: Natural Language, Information Extraction, Databases, Medicines.

1 Introduction

Medical staff need to be able to quickly search for medical information. Common users might also be interested in this type of information, either to learn about diagnosed diseases and prescribed medication, or to complement the information given by physicians. On-line databases, pocket books, or more recently, pocket applications running in smart-phones or PDAs, contribute to this information quest. Nevertheless, the provided search mechanisms are usually limited to keywords or navigation through an index. This is the search scenario of the Portuguese INFARMED website, which supplies information regarding the "Prontuário Terapêutico"[1].

In this paper, we present Medicine.Ask, a software prototype that intends to solve some of the issues that people have to deal with when using the INFARMED website[2], by offering the possibility to formulate questions in Natural Language (Portuguese). We hypothesize that Medicine.Ask is easier to use than the actual search mechanisms in the INFARMED website and that with Medicine.Ask a smaller amount of time is needed to obtain a successful answer. Besides the Natural Language interface, Medicine.Ask architecture comprises the following modules:

- An Information Extraction module, responsible for extracting and processing the information published in the INFARMED website.
- A Relational Database that stores all the extracted and processed data.

We also report the results obtained when validating each individual Medicine.Ask module, as well as when comparing the use of the entire system by real users with the INFARMED website.

This paper is organized as follows: Section 2 presents previous work concerning medical extraction systems as well as some existing web-based systems, used by medical staff and common users to search medical information in the Internet. Section 3 presents the Medicine.Ask system, describing its architecture, and the main techniques used. Section 4 presents the results of the Medicine.Ask validation. Finally, Section 5 presents our conclusions and leaves some directions for future work.

2 Related Work

The widespread use of Internet and mobile handheld technologies brought the opportunity to supply the general public, and physicians in particular, with access to medical data [8]. Information about medicines has special importance for physicians, particularly during the prescription process. Several studies show that the use of medical software systems, such as quick-drug reference systems, by medical staff reduces the number of prescription errors [5].

Physicians have at their disposal several web-based medical systems that offer helpful features. Three important web-based medical systems are: *Epocrates Online*[3], *eMedicine*[4], and *Drugs.com*[5].

These three systems can be described as quick drug and disease references. All of them allow the user to search by disease or medicine name in English, and return information (also in English) about dosage, contraindications, adverse

[1] http://www.infarmed.pt/prontuario/index.php.

[2] For simplicity reasons, we refer to the INFARMED "Prontuário Terapêutico" website as the INFARMED website.

[3] https://online.epocrates.com/home.

[4] http://emedicine.medscape.com/.

[5] http://www.drugs.com/.

reactions, etc, about each drug. They also offer a drug interaction checker. One of the systems, *Drugs.com*, supports phonetic and wildcard search in order to help identifying the correct medicine whenever the spelling of a medicine's name is unknown and only the pronunciation is well-known. The *Epocrates* system clearly distinguishes pediatric dosing from adult dosing. In terms of the information visualization, both systems, *Drugs.com* and *eMedicine*, show the results of a search almost as free text (in the first one), and without a clear distinction between titles and text (in the second one). In opposition, *Epocrates* presents information about medicines in a structured way, very suitable for visualization.

Over the years, the amount of digitalized medical information has been increasing, particularly with the introduction of Electronic Medical Records (EMR). EMRs are digitally stored medical records that contain information about a patient [3]. There are software systems, such as *MedEX* [11] and *cTakes* [9], that aim at giving some kind of structure to clinical records (mostly discharge summaries), that are often unstructured and written as free-text. The goal of this kind of systems is to automate a usually hand-made process, which can be both error-prone and labor-intensive.

MedEX is a Natural Language system that seeks to extract medication information from clinical notes, such as discharge summaries. Discharge summaries typically contain information and instructions about medication, like medicines, dosage, etc. *MedEx* is a system capable of identifying data concerning medicines in clinical notes, such as names, dosage, administration route, etc. To retrieve such information, *MedEx* relies on three main steps. The first step, named *Pre-Processing*, identifies sentences containing medication information, using the *SecTag* [4] sentence boundary detection program. In the second step, called *SemanticTagging*, each token that belongs to a sentence extracted by the Pre-Processing step, is assigned to a medical class (e.g., active substance, brand name, dosage, etc.). This labeling is performed using lexicons that contain medical terms. These lexicons were created from medical dictionaries such as UMLS[6]. Finally, the *Parsing* step translates the tagged sentences into structured forms, using a context free grammar.

cTakes is also a Natural Language Processing system aiming at extracting medical information from medical records, such as discharge summaries. However, unlike *MedEx*, it aims at, not only extracting information regarding medicines, but also regarding diseases, medical procedures, etc. *cTakes* works through six pipelined components. The first one, similar to the *MedEx* Pre-Processing, is the *sentence boundary detector*, that returns all the sentences contained in the clinical note given as input. The second component, the *Tokenizer*, splits each sentence into tokens according to spaces and punctuation. A third component, the *Normalizer*, replaces each token by its corresponding lemma. For example, the token "diseases" is replaced by its lemma "disease". Several tools can perform this task, for example, *TreeTagger*[7][10]. In the fourth component, the *Part-of-Speech Tagger*, each token is annotated with Part-of-Speech information-

[6] http://www.nlm.nih.gov/research/umls/.

[7] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

tion. This allows the fifth component, the *Shallow parser*, to identify all existing noun phrases. Finally, the *Named Entity Recognition* component classifies each noun phrase using a dictionary that maps each noun phrase to one of the five existing categories: diseases, symptoms, procedures, anatomy and drugs.

Both these systems, *MedEx* and *cTakes*, use rule-based techniques to extract relevant information from text. Analogously, the Information Extraction module of Medicine.Ask (described in Section 3.1) relies on the same type of techniques, in particular regular expression recognizers and dictionary-based annotation. In addition, we took advantage of one of the tools used in *cTakes* (*TreeTagger*) used to annotate Natural Language texts.

To conclude, we must say that we do not focus our related work in Question/Answer (QA) or Natural Language Interfaces with Databases (NLIDB). Although being a domain-specific QA system, Medicine.Ask does not add nothing new to these topics. In fact, as many QA systems – see, for instance, the work described in [1], the winning system of the QA@CLEF task[8] for the Portuguese language – Medicine.Ask pre-processes its information sources by using Information Extraction techniques, and, then, feeds a database with the attained structured information. Afterwards, a NLIDB allows users to search for information in Natural Language. As many NLIDB – see [2] for a tutorial – Medicine.Ask follows a rule-based approach, being limited to the medicines' domain. In a point of fact, as we will see, two different approaches allow to "interpret" posed questions. One is to support very specific questions, the other one is based on keyword spotting. Although the later allows to move from a "controlled language", the fact is that only the questions that can be successfully mapped into pre-defined SQL queries will return a correct answer.

3 The Medicine.Ask system

Medicine.Ask is a system that extracts the content of the INFARMED "Prontuário Terapêutico" website, processes all extracted data, giving it an appropriate structure, and stores it in a relational database. This data is then used as a source for answering the questions issued by the users in Natural Language. Medicine.Ask is composed by three main components, *Information Extraction*, *Relational Database*, and *Natural Language Processing.* In this paper, we do not detail the database model. The Information Extraction module is responsible for extracting the information from the INFARMED website. It is also responsible for processing that information, giving it a suitable structure, and store it in a relational database. The *Natural Language* module is responsible for processing the queries posed by users in Natural Language and transform them into SQL queries posed against the database.

[8] `http://www.clef-initiative.eu//..`

3.1 Information Extraction

The Information Extraction module is responsible for extracting the information from the INFARMED website, and storing it in a database. In between, the extracted information undergoes a sequence of processes, such as *Processing of Entity References* and the *Annotation* of medical entities in active substance texts. Figure 1 shows the architecture of the Information Extraction module.

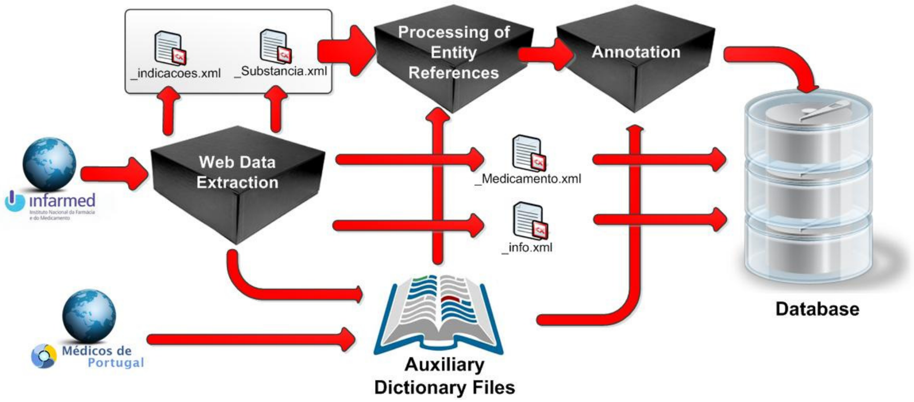

Fig. 1: Architecture of the Information Extraction module.

The Information Extraction module is subdivided into four main components. The *Web Data Extraction* component is responsible for navigating through the INFARMED website and extracting data. There are five main outputs resulting from the Web Data Extraction module: a set of dictionary files and four XML files. The dictionary files contain active substance and medicine names. They also contain names of medical conditions, extracted from the "Médicos de Portugal" website[9]. These dictionary files are used by other Information Extraction components, namely *Processing of Entity References* and *Annotation*. The XML output files contain data regarding the extracted active substances (indications, adverse reactions, precautions, interactions and dosage), medicines (name, price, laboratory, etc.) and overall notes about groups of active substances. Some of the extracted data, such as medicine data stored in "_Medicamento.xml" files, is already structured and ready to be inserted in the database. Other data, such as information about active substances (indications, precautions, etc.) stored in "_Substancia.xml" files, is not structured, and needs further processing before it can be inserted into the database. This processing is performed by two components, Processing of Entity References and Annotation.

The *Processing of Entity References* component handles the existence of references between different active substances. It is very common, in the IN-

[9] http://medicosdeportugal.saude.sapo.pt/glossario.

FARMED website, to find a reference to another active substance in the description of an active substance. For example, in the indications text of the *"Benzipenicilina Benzatínica"* active substance, we can find the text *"V. Benzilpenicilina potássica"*, which is a reference to another active substance. This means that we need to get the indications text of the *"Benzipenicilina Benzatínica"* active substance from the *"Benzilpenicilina potássica"* active substance. The processing of entity references component involves two main tasks. First, we detect the presence of an entity reference in the text. To achieve this, we developed an algorithm that uses regular expressions to spot expressions that may indicate the presence of an entity reference. For example, if we spot the *"V. "* expression, followed by an active substance name (we spot active substance names using the dictionaries created in the Web Data Extraction module), this means that it is a reference to another active substance. More details about this process can be found in [7].

Second, we replace the detected entity references (*"V. Benzilpenicilina potássica"*, in this case) by the text it refers to. This involves searching for the active substance referenced, copy the corresponding text, and place it where the entity reference was detected. It is expected that, after executing the Processing of Entity References component, all the entity references are replaced by the text they refer to.

The data regarding active substances produced by the Processing of Entity References component, contains five fields: indications, adverse reactions, precautions, interactions and dosage. The indication, precaution and adverse reaction fields contain free text with medical conditions (non-structured information), for which the active substance is indicated, requires care, or produces as adverse reaction, respectively. As an example, we have the description of an active substance indication: *"Paracetamol is indicated in cases of fever and pain."*. This indication text contains the medical conditions for which the paracetamol is indicated for. The original texts, as extracted from the INFARMED website, are useful to answer questions such as *"What is Paracetamol indicated for?"*. What about if the user wants to know *"Which are the active substances indicated in cases of fever?"*. With unstructured information, it is impossible to answer that kind of question. It is then important to annotate and collect all the medical conditions present in these free texts.

To annotate the medical conditions present in the indication, adverse reactions and precaution texts we use a combination of techniques, namely dictionary based and Part-of-Speech tagger techniques, along with other handmade heuristics. The dictionary used by the Dictionary based annotator was created from the names of medical conditions, extracted from the "Médicos de Portugal" website, and previously presented. However, with this technique, and according to our evaluation, the dictionary based annotator only collected 50% of the existing medical conditions. To improve the recall, we use a Part-of-Speech tagger technique.

We used the TreeTagger tool to annotate the indication, adverse reaction and precaution texts for Part-of-Speech. In particular, we used TreeTagger to perform

Part-of-Speech classifications, and therefore, find expressions that correspond to medical conditions. For this, we defined a set of Part-of-Speech patterns to be considered as medical conditions. For example, whenever the pattern "NOUN + ADJ" is found ("acute pain", for example), it is considered as a medical condition.

Not all medical conditions fit in the pre-defined patterns, nor exist in the medical conditions dictionary. To catch these exception medical conditions we used several handmade heuristics. For instance, whenever a word tagged by the Part-of-Speech tagger is alone between commas, we considered it as a medical condition. Furthermore, whenever the TreeTagger was unable to classify a word we take that word as a probable medical condition.

The interaction texts are different from the indication, adverse reaction and precaution texts. In these texts, we expect to find names of active substances or medicines with which an active substance interacts. Furthermore, the interaction texts are much more complex in terms of medical language. To annotate interactions within these texts we decided to use a dictionary-based annotation technique. The dictionary used contains all the active substances and medicine names extracted from the INFARMED website.

The dosage texts have their own particularities. These texts contain the recommended dosages for each active substance. Furthermore, in these texts, the dosage is usually distinct for adults and children. It is the goal of the dosage annotation, to identify these two kinds of dosages in an active substance dosage text. The adult dosage is identified by the tag *"[Adultos]"* while the children dosage is identified by the tag *"[Crianças]"*. To distinguish adults from children dosages we took advantage of this tag notation, and used regular expressions to identify them.

3.2 Natural Language Processing

The Natural Language Processing (NLP) module is responsible for processing the queries posed by users. Most of the medical system interfaces, such as the INFARMED website, are more oriented to specialized medical staff, and therefore, less specialized users find it difficult to use. A more user friendly approach allows the user to interact with the system using his/her daily language, and expressing as in front of a physician. The NLP module is used to interpret what kind of question the user is posing (e.g., a question about indications versus a question about adverse reactions), which are the main components of the question (medicines, active substances, etc.), and finally, to translate the Natural Language question into a form that the system understands, which is the SQL language. Figure 2 represents the architecture of the NLP module. It is composed by three different modules: *(i)* Question Type Identification, *(ii)* Question Decomposition, and finally, *(iii)* Question Translation.

The *Question Type Identification* module is responsible for determining what is the purpose of the user question. The type of question represents the purpose of that question. For instance, it identifies if the user is making a question about

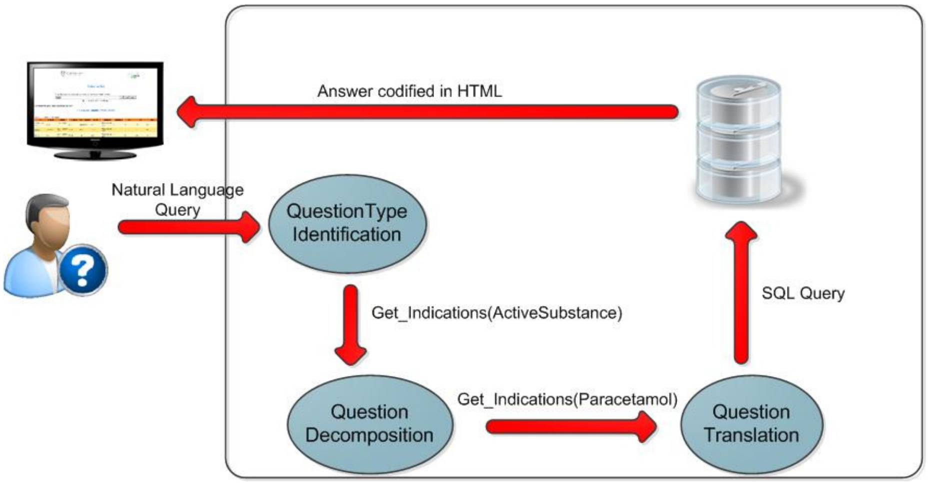

Fig. 2: Architecture of the Natural Language processing module.

the indications of a medicine, or if (s)he is asking for medicines to treat a specific medical condition. As example, the question type for the question *"Which are the indications of paracetamol?"* is Get_Indications(ActiveSubstance). The system has two modes for performing this task, a strict and a free mode. The *strict mode* uses a regular expression technique to match the user query to one of the types recognized by the system. Therefore, it requires the user to pose the question exactly how the system is expecting to. Only this way the user question will match the regular expression of that specific type. For example, the question *"Which are the indications of paracetamol?"* will match the regular expression "Which are the indications of ", and therefore, will be classified as a certain type. Using the *free mode*, the user has a certain degree of freedom when posing the question, allowing different ways of composing the same question. The Free mode uses the *keyword spotting* [6] technique to find important keywords in the user question that can help identifying the purpose of the user question. To find those important keywords, we use dictionaries containing active substance and medicine names, medical conditions, and special keywords that can lead to the identification of the question purpose. We use these dictionaries to annotate the user question. For instance, after the question annotation, because we found the special keyword "indications" and the active substance "paracetamol", the user question *"What are the indications of paracetamol?"* is classified as a question about indications. The remaining words in the question are ignored because they do not belong to the dictionary mentioned before.

The *Question Decomposition* module is responsible for identifying the medical entities that are inside the user question. For instance, it recognizes medicines, interactions, medical conditions and active substances in the user question. This is achieved by using the annotated entities with the dictionary mentioned before. Regarding the previous example, the question decomposition step identifies the "paracetamol" active substance, and with it, fills the question type, resulting in Get_Indications(Paracetamol).

Finally, the *Question Translation* module is responsible for translating the user question into a SQL query, taking into account the purpose of the question and the medical entities in it, such as medical conditions, active substances or medicines. For example, the question type expression Get_Indications(Paracetamol) is translated into a SQL query that applies a selection to the ActiveSubstance table with predicate "name = paracetamol" and then projects the value of the indicationsText attribute. The output returned by the database management system is used to create HTML code with the system answer, and then present it to the user in a web browser.

4 Validation

We validated Medicine.Ask in two different perspectives. First, we validated each module of the system. In this paper, we report the results obtained for the annotation module (Section 4.1) and recommend [7] for the detailed results of the evaluation conducted for the other modules. Second, we compared the Medicine.Ask system and the INFARMED website, when used by real users to solve a set of scenarios (Section 4.2).

4.1 Validation of the annotation module

We considered three types of annotated information. First, we evaluated the annotation process of the indication, adverse reaction, and precaution text. Second, we validated the annotation process of the interaction text. Finally, we validated the annotation of the dosage text. In the three validations, we measured the F-measure obtained, comparing the results obtained to a golden set annotated by hand.

The combination of a dictionary-based and a POS-based techniques for annotating indication, adverse reaction and precaution texts, resulted in a F-measure of 0.77. Due to the high complexity of the interaction texts, the annotation process resulted in a smaller F-measure of 0.47. Finally, we obtained a F-measure of 1.00 in the dosage annotation.

4.2 Comparison with the INFARMED website

In order to evaluate the Medicine.Ask system, we prepared a set of scenarios for real users. These scenarios consisted of tasks with different complexities that medical staff and common users had to answer using both systems, the INFARMED website and Medicine.Ask. The goal was to validate if our system was an improvement, with respect to the existing INFARMED website. We used the following seven scenarios:

- **Scenario 1 -** To obtain indications of an active substance;

- **Scenario 2 -** To obtain the adverse reactions of an active substance;
- **Scenario 3 -** To obtain generic medicines containing a specific active substance;
- **Scenario 4 -** To obtain the cheapest medicines containing a specific active substance;
- **Scenario 5 -** To obtain the indicated medication for a particular medical condition;
- **Scenario 6 -** To obtain the children dosage for a particular medicine;
- **Scenario 7 -** To obtain the medicines for a specific medical condition, without causing a particular side effect.

We gathered a set of 18 users to solve these scenarios. Among these, 10 of them were physicians or Medicine students (i.e., medical staff) and thus had medical background, and 8 were common users with no particular medical knowledge. None of the groups had previous knowledge about Medicine.Ask. Medical staff is, in general, used to the INFARMED website interface while the common user is not. In order to evaluate each system, we collected two kinds of measures. Quantitative measures consisted in: the number of clicks needed to solve a scenario and the time taken to perform it. Qualitative measures were the user satisfaction and the ease of use of each system.

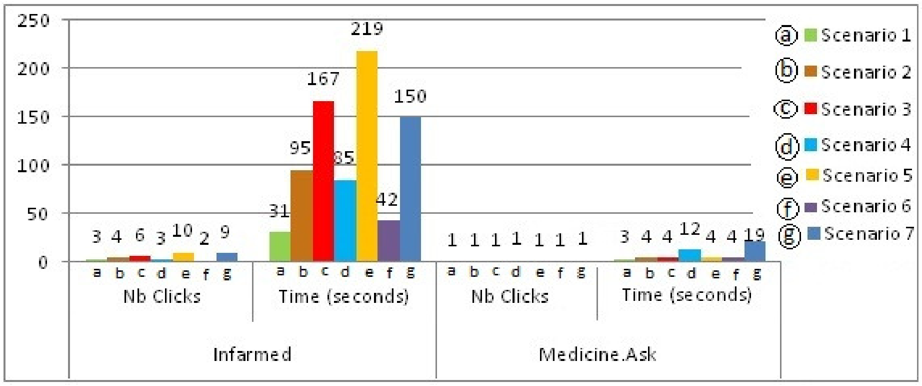

Fig. 3: Average time and number of necessary clicks needed to solve each scenario.

Figure 3 shows a graphic with the collected quantitative measures. We observe that both the time and the number of clicks remain relatively stable, independently of the difficulty of the scenario when using the Medicine.Ask system. However, when using the INFARMED website, the values of these two measures vary according to the scenario, showing higher values in more complicated scenarios, such as scenarios 5 and 7.

In terms of the qualitative measures obtained for each user type, all users were very satisfied with the Medicine.Ask system. Even medical staff who is used to the INFARMED website, showed higher rates of satisfaction when using the Medicine.Ask system. We also observed a difference in the satisfaction of common users and medical staff. As expected, medical staff, who is more used to the INFARMED website, showed higher values of satisfaction when using this system than common users. In terms of ease of use, the results are even more notorious. Common users find the INFARMED website very difficult to use. On the other hand, they did not show significant difficulties when using the Medicine.Ask system. As expected, the medical staff presented fewer difficulties using the INFARMED website. However, they also found the Medicine.Ask system easier to use. Once again, there was a discrepancy between common users and medical staff in terms of ease of use of the INFARMED website. This difference was not observable for the Medicine.Ask system.

5 Conclusions

One of the goals when developing Medicine.Ask was to have a system to be used not only by medical staff, but also by common users, with small medical knowledge. According to the validation reported here, we have achieved this goal. Medicine.Ask is a system capable of answering questions, in Portuguese, about active substances and medicines, such as *"What is paracetamol indicated for?"*, or *"What are the medicines indicated in cases of fever?"*. To accomplish that, we used the INFARMED website as source of information. Relevant data was extracted and stored in a relational database in such a way it could be used to answer user questions. These questions are, unlike the other studied systems, that mostly use a keyword-based search, posed to the system using a Natural Language interface.

Despite the interesting results, there are ideas for future improvements. First, the current Natural Language module is limited. Natural Language embodies an enormous amount of expressiveness, variety, ambiguity and vagueness. Therefore, there is always a user who may surprise the system with a question formulation that cannot be interpreted. We plan to incorporate Machine Learning techniques in the interpretation of a user question. With this kind of techniques, we may be able to improve the system capability for answering new question formulations. Second, it would be interesting to explore some known techniques to annotate medical entities. For instance, so far we have not explored any Machine Learning technique to annotate medical conditions, interactions or dosages in the active substance texts. Third, the validation of the system should be improved. In particular, we plan to compare Medicine.Ask with the INFARMED website using a larger number of users, to report the quantitative measures obtained per user type, and to perform statistical significance tests on the results obtained.

6 Acknowledgements

This work was partially supported by national funds through FCT – Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2011.

References

1. Amaral, C., Cassan, A., Figueira, H., Martins, A., Mendes, A., Mendes, P., Pinto, C., Vidal, D.: Priberam's question answering system in qa@clef 2007. In: Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers. pp. 364–371. Springer-Verlag, Berlin, Heidelberg (2008)
2. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural language interfaces to databases–an introduction. Journal of Language Engineering 1(1), 29–81 (1995), `citeseer.ist.psu.edu/androutsopoulos95natural.html`
3. Carter, J.C.: Electronic health records: a guide for clinicians and administrators. American College of Physicians, 1st edn. (2008)
4. Denny, J.C., Miller, R.A., Johnson, K.B., Spickard, A.: Development and evaluation of a clinical note section header terminology. AMIA Annual Symposium proceedings / AMIA Symposium. AMIA Symposium pp. 156–160 (2008), `http://view.ncbi.nlm.nih.gov/pubmed/18999303`
5. van Doormaal, J.E., van den Bemt, P.M.L.A., Zaal, R.J., Egberts, A.C.G., Lenderink, B.W., Kosterink, J.G.W., Haaijer-Ruskamp, F.M., Mol, P.G.M.: The Influence that Electronic Prescribing Has on Medication Errors and Preventable Adverse Drug Events: an Interrupted Time-series Study. Journal of the American Medical Informatics Association 16(6), 816–825 (November 2009), `http://dx.doi.org/10.1197/jamia.M3099`
6. Jacquemin, C.: Spotting and discovering terms through natural language processing. MIT Press (2001), `http://books.google.com/books?id=W6AB06SBAGMC`
7. Mendes, V.D.: Medicine.Ask: an extraction and search system for medicine information. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa (October 2011)
8. Prgomet, M., Georgiou, A., Westbrook, J.I.: The Impact of Mobile Handheld Technology on Hospital Physicians' Work Practices and Patient Care: A Systematic Review. Journal of the American Medical Informatics Association 16(6), 792–801 (November 2009), `http://dx.doi.org/10.1197/jamia.M3215`
9. Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C., Chute, C.G.: Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the American Medical Informatics Association : JAMIA 17(5), 507–513 (2010), `http://dx.doi.org/10.1136/jamia.2009.001560`
10. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing. Manchester, UK (1994)
11. Xu, H., Stenner, S.P., Doan, S., Johnson, K.B., Waitman, L.R., Denny, J.C.: MedEx: a medication information extraction system for clinical narratives. Journal of the American Medical Informatics Association : JAMIA 17(1), 19–24 (2010), `http://dx.doi.org/10.1197/jamia.M3378`