Medicine.Ask: an extraction and search system for medicine information

Vasco Mendes, Helena Galhardas, Luísa Coheur

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

Abstract. Health personnel deal with medicines in a daily basis. They need to have access to comprehensive information about medicines as fast as possible. Several books and web sites 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 having quick access to information about medicines. Despite all the electronic possibilities available nowadays, the search functionalities provided are usually based on keywords or are class-oriented (allowing, for instance, a search by laboratory or by ATC classification). Our proposal is to build a system that could be used not only by medical staff, but also by common users, with small medical knowledge. For this, we propose a facility to search for information about medicines through a (controlled) set of questions posed in Natural Language. An example of such a question is: “Which are the medicines for influenza that can be used during pregnancy?”. In this paper, we present Medicine.Ask which is a question-answering system about medicines that couples state of the art techniques in Information Extraction and Natural Language Processing. We present the architecture of the system and the main techniques used. Furthermore, we report the experiments that were carried on to validate the modules of the system. The results obtained show that we have achieved our goal of making a system easy enough to use, not only by medical staff but also by common users, with small medical knowledge.

Keywords: Natural Language, Information Extraction, Database, Medicine.

1 Introduction

Internet is increasingly being used to publish the most diverse types of information. Medical information is no exception. It is easy to find medical information regarding medicines, diseases, etc., available in the Web. This information is particularly important to medical staff, who usually need to quickly search through this information. However, with the spread of new technologies, more and more people seek out for this kind of information. Common people, who do not have
any background in medicine, also want to know about diagnosed diseases and prescribed medication, in order to complement the information given by doctors. Nowadays, in Portugal, when both, medical staff and common users want to get information regarding medicines or diseases, they must access specialized books, databases available on-line (e.g. through the INFARMED website \(^1\), pocket books, or more recently, pocket applications running in their smartphones or PDAs. However, this type of search, even when using on-line databases or software applications, is usually either made by keywords or extended search (e.g. find the desired information through an index, similar to a book).

Medicine.Ask is a software prototype that intends to solve some of the issues that medical staff encounter when using systems like the INFARMED Website. Medicine.Ask uses the information regarding medicines and active substances published online by INFARMED, and allows users to search through it, using Natural Language interface. This system extracts all the information present in the INFARMED website, and then structured it in such a way that users can search through it, using a simple Natural Language interface. This solution represents an improvement over the existing alternatives (e.g. eMedicine \(^2\), Epocrates \(^3\) and Drugs.com \(^4\)), because has an interface easier to learn how to use, even for users with few computer skills, and allows to answer the same questions, in less time and interactions.

The INFARMED website contains medical notes regarding active substances and medicines. In what concerns medicines and active substances it also provides the prices, indications, adverse reactions, precautions, interactions, dosage, etc. There are three ways for searching this information in the INFARMED website. The first one corresponds to the navigation through the INFARMED hierarchic structure, very similar to the index of a book. This kind of search requires a significant medical background, and therefore, it is not indicated to common users. A second one allows the user to directly search for information regarding a specific active substance or medicine, through a keyword search field. Finally, a third search method allows the user to search for any textual fragment, potentially present in the INFARMED website documents.

These search mechanisms are limited and present several drawbacks. For instance, there is no quick way to search for medicines indicated for a specific disease, “pneumonia”, for example. One way is to navigate through the INFARMED website hierarchy, which requires some kind of medical background. In this case, the user needs to know that “pneumonia” is caused by a bacteria, and therefore, search under “Antibacterial” medication section of the site hierarchy. Another way to answer that question is to use the third search mechanism, and search for all documents that contain the word “pneumonia”. Since this kind of search is blind, the system returns all documents containing the word “pneumonia”, regardless of its location. Instead of only returning medicines indicated

\(^1\) http://www.infarmed.pt/prontuario/index.php
\(^2\) http://emedicine.medscape.com/
\(^3\) https://online.epocrates.com/
\(^4\) http://www.drugs.com/
for “pneumonia”, this means that some results may contain medicines that need precautions in case of “pneumonia”, or that can cause “pneumonia” as adverse reaction.

Medicine.Ask offers a Natural Language interface, that allows users to inquire the system using Natural Language queries (e.g. “What are the indications of Paracetamol?”). This kind of search mechanism brings high improvements when compared to the INFARMED keyword-based search mechanisms, namely, it is easier to use, needs a smaller amount of time and obtains the answer with less number of clicks.

In this paper, we describe the Medicine.Ask system. The architecture of Medicine.Ask comprises the following modules:

- An Information Extraction module, responsible for extracting and processing the information published in the INFARMED website. The processing includes the resolution of entity references and annotation of medical entities, in order to improve the quality of the extracted data, and transform the non-structured data into structured data.
- A relational database that stores all the extracted and processed data.
- A Natural Language module, used to recognize, understand and answer the Natural Language queries posed by the users.
- A validation of each isolated Medicine.Ask module, and a validation of the global Medicine.Ask system with real users, highlighting the characteristics that make this system a better solution, when compared with the “Prontuário farmacêutico” from the INFARMED website.

The rest of 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 concepts, and the main techniques used. Section 4 presents the results of the Medicine.Ask validation with real users. Finally, Section 5 presents our conclusions and leaves some directions for future work.

2 Related Work

The widespread use of Internet and mobile handled technologies brought the opportunity to supply the general public, and doctors in particular, with access to medical information (Prgomet, Georgiou, & Westbrook, 2009). Information about medicines has special importance for physicians, particularly when prescribing drugs. Several studies show that the use of medical systems, such as quick-drug reference systems, by medical staff reduces the number of medication errors (Doormaal et al., 2009).
Physicians have at their disposal several web-based medical systems that offer helpful features. The most relevant web-based medical systems are the Epocrates Online 5, eMedicine 6, and Drugs.com 7.

These three systems can be described as quick drug and disease references, and have similar features. All of them allow the user to make a search by disease or drug, and return information about dosage, contraindications, adverse reactions, etc, about each drug. One of the system, Drugs.com, allows 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 distinct pediatric dosing from adult dosing.

Over the years, the amount of digitalized information has been increasing, particularly Electronic Medical Records (EMR). Despite this increase, there is still a lack organization of those records. There are software systems, such as MedEX (Xu et al., 2010) and cTakes (Savova et al., 2010), 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 this 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 on medication, like medicines, dosage, etc. MedEx is a system capable of identifying data concerning medicines in clinical notes, such as medicine 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 (Denny, Miller, Johnson, & Spickard, 2008) sentence boundary detection program. The second step, called SemanticTagging, each token that belongs to a sentence extracted from the Pre-Processing, is assigned to a medical classification (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. Finally, the Parsing step, parses 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 system is the sentence boundary detector, that returns as output 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

---

5 https://online.epocrates.com/home
6 http://emedicine.medscape.com/
7 http://www.drugs.com/
8 http://www.nlm.nih.gov/research/umls/
component, the **Normalizer**, replaces each token by its corresponding lemma. For example, the token “diseases” is replaced by its lemma “disease”. There are several tools to perform this, for example, TreeTagger\(^9\) [Schmid, 1994]. Next, in the fourth component, the **Part of Speech Tagger**, each token is annotated with part of speech information. 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. This dictionary maps each noun phrase to one of the five existing categories (Diseases, Symptoms, Procedures, Anatomy and drugs).

Both these systems, **MedEx** and **cTakes**, contain several relevant techniques for our Medicine.Ask system. For instance, they present techniques and tools to annotate texts written in Natural Language system, namely, the TreeTagger system. Medie.Ask uses TreeTagger to annotate Natural Language texts with Part of Speech information.

## 3 The Medicine.Ask system

Medicine.Ask is a system that extracts all the content published in the INFARMED website, processes all extracted data, giving it an appropriate structure, and stores it in a suitable database. This information is then used as knowledge source for answering the questions issued by the users in Natural Language. The Medicine.Ask system is composed by two main components, the **Information Extraction** and the **Natural Language Processing** modules. 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 the one responsible for processing the queries in Natural Language, posed by the users. It is through this Natural Language interface that the user interacts with the system, accessing this way to the database content.

### 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 \(\$\) shows the architecture of the Information Extraction module.

The information extraction module is subdivided into four main components as illustrated in Figure \(\$\). The **Web Data Extraction** component is responsible for navigating in the INFARMED website and extracting its data. There are five main outputs resulting from the Web Data Extraction module: a dictionary file and four XML files. The dictionary file contains names of active substances and medicine names. It also contains names of medical conditions, extracted from

---

\(^9\) [http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/](http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/)
the “Médicos de Portugal” website \textsuperscript{10}. This dictionary file is used by the other Information Extraction components, namely the Processing of Entity References and Annotation. The remaining 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, is already structured and ready to be inserted in the database. Other data, such as information about active substances (indications, precautions, etc.), is not structured, and needs further processing before it can be inserted in the database. This processing is performed by the two other components, the Processing of Entity References and Annotation components.

The processing of entity references handles the existence of references between different active substances. It is very common, during a navigation in the INFARMED website, to find in the description of an active substance a reference to another active substance. For example, in the indications text of the “Benzipenicilina Benzatínica” active substance, we can find the text “V. Benzipenicilina 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 “Benzipenicilina potássica” active substance. The processing of entity references involves two main tasks. First, we need to detect the presence of an entity reference in the text. To achieve this we developed an algorithm that, along with regular expressions, spots expressions that may indicate the presence of an entity reference. Algorithm 1 shows the algorithm for detecting and processing entity references.

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.

Second, we need to replace the detected entity references (“V. Benzipenicilina potássica”, in this case) by the text it refers to. This involves searching for

\textsuperscript{10} http://medicosdeportugal.saude.sapo.pt/glossario
the active substance referenced, copy out its text, and place it where the entity reference was detected. It is expected that, after the entity reference processing, 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 indications, precautions and adverse reactions fields contain free text with medical conditions (non-structured information), for which the active substance is respectively, indicated, requires care or produces as adverse reaction. As example, we have the description of an active substance indication: “Paracetamol is indicated in cases of fever and pain.”. This indications text contains the medical conditions for which the paracetamol is indicated for. The original texts, as extracted from the INFARMED website, are only 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 the information in a non-structured way it is impossible to answer that kind of question. It was then important to annotate and collect all the medical conditions present in these free texts. The fact that these texts are written in Natural Language means that this task needs to be addressed using annotating techniques.

To annotate the medical conditions present in the indications, adverse reactions and precaution texts we used a combination of techniques, such as 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 decided to include a Part-of-speech tagger technique.

We used the TreeTagger tool to annotate the indications, adverse reactions and precautions texts for part-of-speech. We could now use TreeTagger to

---

**Algorithm 1**

```plaintext
1: List of files ← files_with_name_ending_with("_Substancia.xml" OR "indicacoes.xml")
2: Regex ← “Regular expression present in Appendix ??”
3: for each f in List of files do
4:   for each line in f do
5:     if line:contains(“As dos componentes”) then
6:       TreatComponentReferences(f)
7:     end if
8:     if line:contains(“V.”) then
9:       List of entity_reference_container_text lct ← line.findMatches(Regex)
10:      for each er in lct do
11:        TreatEntitytReferences(er; f)
12:      end for
13:     end if
14:   end for
15: end for
```

---

11 http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
perform part-of-speech classifications, and therefore, find expressions that are 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 misbehavior medical conditions we used several handmade heuristics. For instance, we decided that, whenever a word tagged by the POS 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 combination of all this techniques and heuristics resulted in a high recall (94%) which means that almost all medical entities were correctly identified.

The interaction texts are different from the indications, adverse reactions and precautions texts. In these texts we expect to find, not medical conditions, but names of active substances or medicines with which that 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, with the dictionary that contains all the active substances and medicine names, published in the INFARMED website. Due to the high complexity of the interaction texts, the annotation process resulted in a smaller recall, were only 68% of the interactions were correctly identified.

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 separate, in an active substance dosage text, the adult dosage from the children dosage. The adult dosage is identified by the tag “[Adultos]” while the children dosage is identified by the tag “[Crianças]”. To isolate both adult and children dosage we decided to take advantage of this tag notation, and used regular expressions to identify them. Our validation process reveals a F-measure of 100% in the dosage annotation.

3.2 Database

After the extraction and processing of the information present in the IFARMED website we needed some structure to store it. The solution was to use a SQL relational database. This database will constitute the source for answering the questions issued by the users in Natural Language. This database stores the extracted entities and the relationships between them.

A part of the ER model of the database is represented in Figures 2, and uses the notation proposed in (Silberchatz, 2005).

3.3 Natural Language Processing

The Natural Language Processing (NLP) module is responsible for processing the queries posed by users. Most of the medical systems interfaces, such as the
Figure 2: Part of the Medicine.Ask database ER model, representing the main entities.
INFARMED website, are more oriented to specialized medical staff, and therefore, less specialized users find it difficult to use. A more user friendly approach will allow the user to interact with the system using its daily language, expressing himself as it would in front of a doctor. 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 SQL language. Figure 3 represents the architecture of the NLP module.

![Architecture of the Natural Language processing module.](image)

The process starts with the user posing a question to the Medicine.Ask system. Then, the Natural Language processing is divided into three different steps: (i) question type identification, (ii) question decomposition, and finally, (iii) question translation.

The **question type identification** step is responsible for identifying what is the purpose of the user question. The type of a question represents the purpose of that question. For instance, it identifies if the user is making a question about the indications of a medicine, or if 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 doing 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. The Strict mode 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** (Jacquemin, 2001) 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 dictionar-
ies 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 question about indications. The remaining words in the question are
ignored because they do not belong to the dictionary mentioned before.

The **question decomposition** step is responsible to identify the medical en-
tities that are inside the user question. For instance, it recognizes medicines,
interactions, medical conditions and active substances in the user question. This
is made using the annotated entities with the dictionary mentioned before. Re-
garding the previous example, the question decomposition step would identifies
the active substance “paracetamol”, and with it, fills the question type, resulting
in “Get_Indications(Paracetamol)”.

Finally, the **question translation** step is responsible for translating the user
question into a SQL question, taking into account the purpose of the question
and the medical entities in it, such as medical conditions, active substances or
medicines. The output returned by the database is used to create HTML code
with the system answer, and presented to the user in a web browser.

## 4 Validation

In order to evaluate the Medicine.Ask system, we prepared a set of scenarios for
real users. These scenarios consisted of several tasks that equally, medical staff
and common users, had to answer, using both systems, the INFARMED website
and Medicine.Ask. This way we were able to validate if our system really was
an improvement, face to the existing INFARMED website. We used a total of 7
scenarios, involving the following tasks:

- **Scenario 1** - Obtain indications of an active substance;
- **Scenario 2** - Obtain the adverse reactions of an active substance;
- **Scenario 3** - Obtain generic medicines containing a specific active sub-
  stance;
- **Scenario 4** - Obtain the cheapest medicines containing a specific active
  substance;
- **Scenario 5** - Obtain the indicated medication for a particular medical con-
  dition;
- **Scenario 6** - Obtain the children dosage for a particular medicine;
- **Scenario 7** - Obtain the medicines for a specific medical condition, without
  causing a particular side effect.
In order to evaluate each system, we collected several measures. The quantitative measures are the number of clicks needed to solve a scenario and the time taken to do it. The qualitative measures are the user satisfaction and the ease of use of each system.

Regarding the quantitative measures, Figure 4 shows a graphic with the collected quantitative measures. It shows the average of the time and necessary clicks needed to solve each scenario.

As observed, when using the Medicine.Ask system, both the time and the necessary number of clicks remains relatively stable, independently of the difficulty of the scenario. However, when using the INFARMED website, the necessary time and number of clicks varies according to the scenario, showing higher values in more complicated scenarios, such as scenarios 5 and 7.

In Figure 5 we can observe the qualitative measures, obtained by user type. It is natural that more experienced users, such as medical staff, find the INFARMED website easier to use than the common user.

As we can observe, from Figure 5, all users were very satisfied with the Medicine.Ask system. Even medical staff, that is used to use the INFARMED website, showed higher rates of satisfaction when using the Medicine.Ask system. We also can observe from this graphic that, there is a difference in the satisfaction of common users and medical staff. As expected, medical staff, that is more used to use the INFARMED website, present higher results of satisfaction. 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 more, we can observe the discrepancy between the common users and medical staff in terms of
ease of use. This difference is not observable in the Medicine.Ask system. This tells us that we accomplished our main goal: to build a system that can be used by both medical staff and common users with no medical training.

5 Conclusions

Our goal, when we started developing the Medicine.Ask system, was to build a system that could be used not only by medical staff, but also by common users, with small medical knowledge. According to our validation, we believe we have achieved the desired goals, and we present the Medicine.Ask system, as a system capable of answering questions, in Natural Language, 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. From it we extracted all the content of the pharmacy records. That extracted information was then processed, treating entity references and annotating existing medical entities. All the processed information was then stored 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. This means that the user can inquire the system as it would inquire a pharmacist about active substances or medicines, using its daily language.

Despite the interesting results, there are also many ideas for future improvements. The Natural Language module is always, somehow, limited. Natural Lan-

![Figure 5: Qualitative evaluation for both systems, containing the ease of use and satisfaction measures.](image-url)
guage embodies an enormous amount of expressiveness, variety, ambiguity and
vagueness. Therefore, there is always a user that surprise us, with a question
formulation that the system cannot interpret. The alternative may pass to in-
corporate 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, for which it is not currently ready to
answer. Due to time constraints, we did not explore some known techniques to
annotate medical entities. For instance, we did not explore any Machine Learning
technique to annotate medical conditions, interactions or dosages in the active
substance texts. Although our validation results are good, we cannot be sure
whether Machine Learning techniques would bring better results.

References

and evaluation of a clinical note section header terminology. AMIA Annual
Symposium proceedings / AMIA Symposium. AMIA Symposium, 156–160.
Doormal, J. E. van, Bemt, P. M. L. A. van den, Zaal, R. J., Egberts, A. C. G.,
Lenderink, B. W., Kosterink, J. G. W., et al. (2009, November 1). 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.
processing. MIT Press.
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.
Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler,
System (cTAKES): architecture, component evaluation and applications.
Journal of the American Medical Informatics Association : JAMIA, 17(5),
507–513.
Xu, H., Stenner, S. P., Doan, S., Johnson, K. B., Waitman, L. R., & Denny, J. C.
(2010). MedEx: a medication information extraction system for clinical