

# A System for Industrial-strength Linguistic Parsing of Medical Documents

Sven Schmeier, Dr. Martin Hirsch  
interActive Systems GmbH  
Dieffenbachstraße 33c  
10967 Berlin  
Germany

[sven.schmeier@interActive-Systems.de](mailto:sven.schmeier@interActive-Systems.de)  
[martin.hirsch@interActive-Systems.de](mailto:martin.hirsch@interActive-Systems.de)

## ABSTRACT

This paper describes SP MED, a system for robust and accurate linguistic parsing of medical documents which is used in several industrial products. The basic design criterion of the system is of providing a set of basic powerful, robust, and generic linguistic knowledge sources and modules which can easily be customized for processing different tasks in a flexible manner. The main application is seen in linguistic analysis of medical documents, yet the technology is easily applicable to other domains

## KEY WORDS

shallow linguistic parsing, clause identification, biomedical domain

## 1. Introduction

The amount of textual information in the medical domain electronically available is beyond its critical mass leading to the problem that the more electronic text is available the more difficult it is to find or extract relevant information. In order to overcome this problem several different approaches exist starting from scalable yet simple search engines based on statistical information retrieval and ending up with purely linguistic based analysis systems using lots of domain relevant knowledge. One line of such research is the combination of shallow information extraction (IE) and statistical information retrieval systems. The goal of such IE systems is to find relevant information from text data while ignoring extraneous and irrelevant information [1]. The main problem in medical documents, for example in Medline [2], is the complexity of sentences as their average number of words is more than 25 and the number of embedded clauses is more than 2.3. Furthermore current IE systems are focussed on detecting special information whereas in most cases the user cannot predict what the information may look like he is really interested in.

In this paper we report on SP MED, a linguistic parser for real world medical text processing. The system itself is designed for English and German text. The main focus in

this paper lies on English as it is the common used language in our domain.

## 2. The overall architecture of SP MED

The basic design criterion of SP MED is to provide a set of powerful, robust, and efficient natural language modules and linguistic knowledge sources that are especially designed for processing complex documents in the medical area. Furthermore its overall architecture is based upon modern software engineering standards. Hence we view SP MED as an *industrial-strength core language processing system* for this domain which can be easily adapted to other domains. Customization is achieved in the following directions:

- defining the workflow between the modules
- selection of linguistic knowledge sources
- formulating domain specific knowledge
- implementing additional modules

### 2.1 Main Components

Figure 1 shows a blueprint of the core system. The main components are:

A Tokenizer based on regular expressions: it scans the input text and detects special tokens like date and time expressions, special medical expressions – which are one main part of the system –, bracket information, abbreviations and words. (Information inside brackets is generally treated as proper names as in our domain brackets contain chemical or medical names in the most cases.)

The part-of-speech (POS) Tagger consists of well known technology trained on the Penn Treebank [3] and on the GENIA corpus [4]<sup>1</sup>.

---

<sup>1</sup> Both training corpora are used for this paper only. In industrial applications other resources that have been manually created by interActive-Systems GmbH are used.

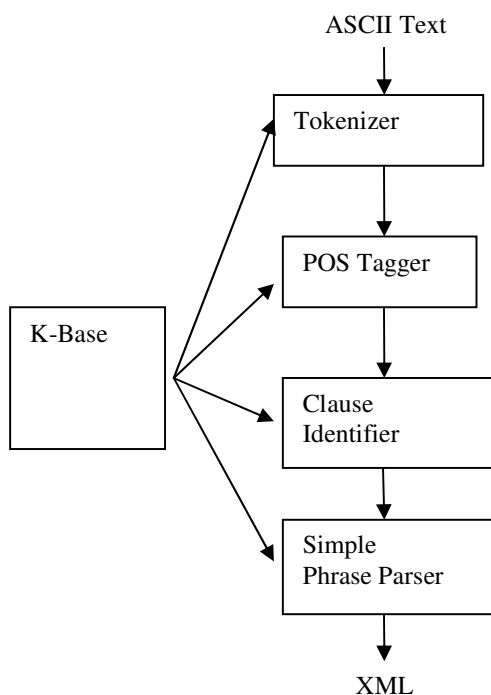


Figure 1: A blueprint of the core system

Its overall accuracy is ~96% for 10-fold cross validation on the combined corpora. Furthermore we implemented a detector for separating between prepositions introducing a prepositional phrase and prepositions that belong to the verbs (like “come from” or “opposed to”). Those prepositions are retagged from “IN” to “PTKVZ” which means verb particle adjunct.

Clause identification presents the second main part in the whole system. Especially in medical domains we found out that embedded clauses are used very often. Accurate clause identification simplifies the process of further parsing like phrase parsing or subject-verb-object (SVO) detection. Clause identification is a well known technology [5]. Nowadays systems most often make use of machine learning techniques where decisions are based on observations on manually tagged corpora. However our Clause Identifier consists of rules that are based on the POS tags and words in the sentence. (see section 3)

The Phrase Parser we use is called “simple” as we detect phrases that do not contain or consist of clauses only. We decided to build a self-learning Phrase Parser that extracts phrase information in annotated corpora, i.e. Penn Treebank and GENIA, and stores it in a *TRIE* data structure [6]. This gives us great flexibility in adding or deleting forms of phrases and retraining the Phrase Parser on the one hand and furthermore gives the possibility to easily handle domain specific phrases.

The results of analysis are represented as objects or XML. Additionally to the above mentioned components the whole system is embedded in a client/server system in order to work online.

There are two important properties of the system for supporting reusability in other domains:

- The system is designed as a whiteboard system with no destructive operations. This means components may be exchanged or added at each stage.
- The workflow of the system is not fixed. This allows for definition of cascaded as well as interleaved flow of control.

The system has already successfully applied to improve the results of ordinary full text search system. We identified SVO relations and added this information to the index as well as further linguistic information like prepositional phrases (PP) and genitive attachments. With this the full text search system may for example detect chains of information or build an online content specific thesaurus structure.

## 2.2 Reused Technologies and Linguistic Resources

As mentioned in the previous section we used well known technologies for some modules in our system. The Tokenizer is based on regular expressions that are formulated to detect special tokens. A text scanner reads in the text, (1) expands abbreviations, (2) applies the regular expressions on it, (3) marks information contained inside brackets and (4) performs extraction of special medical expressions. (2) is used for detection of time and date expressions as well as for measures like mg/l or mol/kg. Bracket information in (3) is tagged as proper name at the moment as in our application more than 95% of brackets simply contained expressions like: (NO<sub>2</sub>), (SU), etc. For other domains tagging of information inside brackets has to be clearly improved. (4) uses a lexicon of typical medical terms and phrases.

The used POS Tagger is the SVM-Tool [7] based on support vector machines [8]. We added some functionality concerning the distinction between “real” prepositions and those that are attached to a verb according to a verb subcategorization lexicon. Furthermore we occasionally revise decisions of the POS Tagger using a fullform lexicon for English<sup>2</sup>.

The Phrase Parser consists of a *TRIE* filled with POS chains denoting common and special nominal and prepositional phrases.

The basic linguistic resources we use consist of:

- English full form lexicon
- Verb subcategorization lexicon for English
- Special lexica (medical domain)
- Regular grammars for Tokenization
- Penn Treebank and GENIA corpus for training of the POS Tagger

<sup>2</sup> We noticed that this kind of revision must be handled with care because of the coverage of the lexicon – especially in medical domain – as well as some classified POS may contradict the “real” part of speech but make sense anyway.

### 3. Clause Identification

Clause Identification is one main part in the whole system and represents besides its overall architecture and our medical resources the core innovation. Several approaches for identifying clauses in sentences have been proposed in the past. The main design criterion of the systems according to the results achieved in [5] is the usage of learning algorithms that use previous clause identification results obtained by manually tagged corpora for their decisions. Despite of the results achieved on standard corpora like the Penn Treebank, we noticed that the systems produce errors in very complex or simply different domains. This is mainly because of the lack of special training data in these domains.

Hence we decided to build a Clause Identifier using the results of [9] as a start point. Here three rules are proposed for detecting the start of a clause and one rule for marking its end:

Clause starts:

- (1) Is the item a conjunction?
- (2) Is the item a subjectless verb?
- (3) Is the item the subject of a verb?

Clause ends:

- (1) Is the item followed by a clause initiator?

In this approach we found some practical problems as for example: it is almost impossible for a system to decide whether a preposition is used as adverb, conjunction or simply as preposition (rule 1). Similar problems arise to rule 2 and rule 3. Thus we introduced some additional steps and rules. We ended up with a system that performs a LR clause identification, i.e. we greedily scan the sentence for possible clause starting and end points from left to right with the risk of overgenerating markers (a), then reduce the found markers scanning the sentence from right to left (b) and merge the remaining part-clauses.

#### 3.1 Left to Right Stage

In (a) we use 7 different main rules for marking clause starting and end points. Furthermore SP MED contains 23 sub-rules to handle special cases and garden paths [10].

A new clause<sup>3</sup> may start if:

Expanding original rule 1:

:

- The POS is either preposition, wh-question word, conjunction or a comma. Several words

---

<sup>3</sup> Here we also generate part-clauses that will be merged to real clauses in the second stage. Example after stage 1: [Paul] [who is the husband of Mary] [stands in front of the building] ; after stage 2: [Paul stands in front of the building] [who (Paul) is the husband of Mary]

like “of”, “under”, “over”, etc. are excluded: “The boy [S who read the book]”

- The POS is “TO” which means it may introduce a infinitive clause (a). It does not introduce a new clause if it is preceded by all forms of the words “to have” and “to be” or “to” is followed by other POS than verb, adverb or adjective (which is then used as an adverb) (b): “I want you [S to read]”(a) vs “He has to read the book”(b).

Expanding rule 2:

Determine the form of the word. In case of

- Gerund: look for preceding form of “to be”(a) and mark a clause start if not found(b): “I am going home”(a) vs. “I like [S being an outlaw]”(b).
- Past: look for preceding form of “to have”(a) or an open noun that is not in the scope of any other word (b) and mark a clause start if not found (c): “The plane has started”(a) vs. “The plane started”(b) vs. “The boy saw the plane [S started from the airport]”(c).
- Presence: look for an open noun that is not in the scope of any other word(a) and mark a clause start if not found(b): “The boy sees the man”(a) vs. “The man goes into the kitchen, [S opens a tin]” (b)

Expanding rule 3:

- A personal pronoun or a determiner follows a noun: “The man [S the boy saw] smoked.”
- A noun or pronoun follows an intransitive verb: “In earlier times t-rex existed [S a researcher said]”

A clause may end if:

Using rule 4:

- The item is followed by a clause starter.

The sub-rules handle cases like occurrences of wh-words with “to” (... to which the horse belongs), constellations of coordinations (.. the house, the man, and the cat, or the dog), etc.

#### 3.2 Right to Left Stage

This stage uses linguistic topology rules and eliminates the over-generated markers. Note that these rules make use of some heuristics as we for example do not expect deep embedded clauses like “The man the boy the dog barked at lifted smiled”. The main rules are:

- The last clause must contain a verb. If not merge the last 2 clauses.

- For two adjacent clauses at least one clause must contain a verb
- A clause that is introduced by a subordinate, i.e. preposition, coordination or wh-question word, must contain a verb
- A clause must contain a noun or a (suppressed) pronoun.
- Merge verbless clauses with their corresponding clauses.

### 3.3 Results of Clause Identification

With these rules the system achieves an overall accuracy of about 85% on our data which mainly consists of Medline [2] abstracts. Ignoring irrelevant errors for later steps like SVO extraction the “application” accuracy raises > 90% which is comparable to current clause identification systems working on PENN Treebank.

Besides the performance the main advantage of this architecture lies in the simple adaptability to new domains. We adapted the system for dealing with abstracts from patent office; the abstracts are written in a very short telegram-style manner. The sentences are short and contain lots of abbreviations. The whole implementation needed 2 days. After this the system achieved an overall accuracy of about 87%.

## 4. Coverage of Linguistic Resources

The lexicon contains 175557 stem entries; the verb subcategorization lexicon contains 42388 entries (8502 stems). The lexicons itself are stored in a database so adding or removing entries is done by changing the database.

The time and date subgrammar covers a rather limited but powerful set of expressions as the standard way of expressing time and date is limited too in medical documents. The set of expressions clearly has to be expanded when dealing with text of different domains.

The NP/PP subgrammar covers all simple NP and PP constructs as they occur in the Penn treebank and GENIA corpus, and special NP and PP as they were detected in medical documents while using the system. Simple means we only use NP and PP that occur within one clause and do not contain or consist of any other clause. Again the representation is plain text so adding or removing phrase rules is done by changing the property file. On system startup this file is read and a TRIE structure is built. This process needs around 3 seconds.

The second main part of the system is the medical knowledge source containing so called “Technical Terms” (TT). The TTs have been extracted from [11], [12] and [13] and checked automatically and manually. The resulting TTs consist of about 463800 medical and domain specific complex terms, e.g.: “Vertebral Basilar Insufficiency” or “Glucose Urine Test – (Glucose Oxidase)”, etc. With this we simplify and improve the Phrase Parser component as well as the clause identification module.

## 5. Current and Future Applications

SPMED is or will be used in mainly four different applications:

- (1) Information Management system: analysis of documents that will be stored in a search index; extraction of subject-verb-object relations and suppressing redundant information
- (2) “Report on Demand” system; information extraction of several information; template filling
- (3) Generation of first and second order relationship between concepts.
- (4) Q-Tools: A tool for improving and optimising SPMED’s performance and for finding new relevant technical terms in documents.

For these application the main architecture as described above, the Tokenizer, the Tagger, the Clause Identifier and the Phrase Parser as well as the underlying linguistic resources could be used basically unchanged. We expect changes when the underlying documents in the application change (as already mentioned for (1) in context of patents).

In (1) SPMED is embedded in the *sciPlover* system, an information management system for medical applications. SPMED is used to analyze and prepare the underlying document storage system that may be seen as a highly sophisticated and automatically structured search index for several different search tasks. Especially for scientists or doctors information about documented adverse reaction of drugs or which active ingredients are used for which symptoms are of special interest. For this we use the automatically extracted SVO triples and let the user browse through them (see Fig. 2).

In (2) additional lexical resources and more elaborated XML output structures are used since the expected information is more detailed and complex.

For (3) we restricted the Clause Identifier part for being more precise with a reduced recall. Concept relation generation in that case turned out to be rather critical with respect to errors.

Finally in (4) we use a special GUI for adding new rules to the Tokenizer or the Phrase Parser. Adding is done by simply drag and drop certain text snippets or formulating new regular expressions. Furthermore collection of new Technical Terms can be semi-automated.

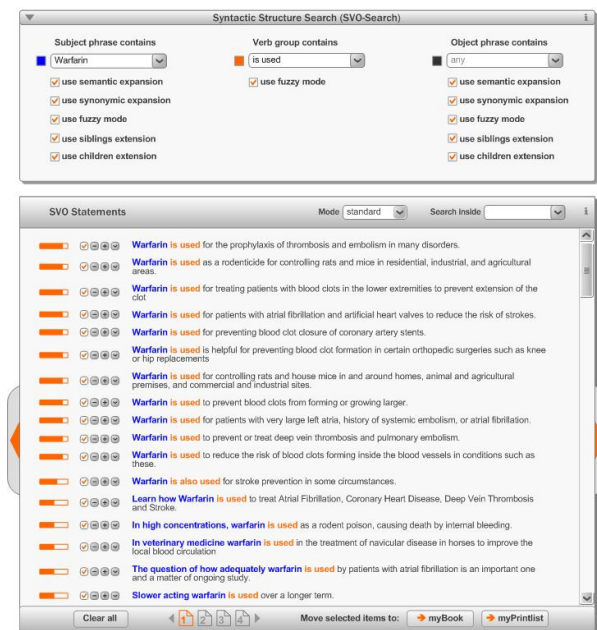


Fig.2: The syntactic structured search of the *sciPlover* system

Processing is very robust and fast – in average 0.1 CPU seconds (Intel 1.6 GHz Pentium M) per sentence (avg. 25 words) - yet the system is not fully optimized at the moment. In the first three applications we obtained high coverage and good results. However a systematic evaluation should be done thus we cannot claim that these results are comparable. We compared it with the PCFG parser by the Stanford group [14] for some specific cases and felt the results are comparable in short sentences and better on longer and more complex sentences. One reason for this is that the Stanford parser is a general parser whereas SP MED is especially designed for medical domains.

## 6. Conclusion

We have described SP MED, an industrial-strength system for robust and accurate linguistic parsing of medical documents. The basic design criterion of the system is of providing a set of basic powerful, robust, and generic linguistic modules and knowledge sources which can easily customized for processing different tasks in a flexible manner. The main features are: a sophisticated Clause Identifier based on rules, a flexible kind of usage of linguistic resources, well elaborated knowledge sources containing medical terms and an overall architecture based on modern software engineering standards. The system has been fully implemented in Java 5 (jdk 1.5) and is used in industrial systems.

Future activities will be in fully adapting the system to German language, i.e. building regular expressions for extracting German time and date expressions, and building the grammar for German clause identification.

The Phrase Parser will remain mainly unchanged as simple German (noun-) phrases are constructed similar to English; for the POS Tagger we simply retrain the module on German corpora.

## References

- [1] G.Neumann, R.Backofen, J.Baur, M.Becker, C.Braun, An Information Extraction Core System for Real World German Text Processing. *In Proceedings of 5th ANLP, Washington, March, 1997.*
- [2] National Library of Medicine, <http://www.nlm.nih.gov/>
- [3] M. Marcus, B. Santorini, B. Marcinkiewicz, *Building a large annotated corpus of English: the Penn Treebank* (Computational Linguistics, vol. 19, 1993)
- [4] Kim, Jin-Dong, Tomoko Ohta, Yuka Teteisi and Jun'ichi Tsujii, *GENIA corpus - a semantically annotated corpus for bio-textmining* (Bioinformatics. 19(suppl. 1). pp. i180-i182. Oxford University Press, 2003)
- [5] COLING 2000. Saarbruecken, Germany, 2000
- [6] A.Aho, J.Hocraft, and J.Ullmann, *Data structures and algorithms* (Addison Wesley, Reading, Mass, 1983)
- [7] Jesús Giménez and Lluís Márquez. SVMTool: A general POS Tagger generator based on Support Vector Machines, *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal. 2004*
- [8] Thorsten Joachims. Making large-Scale SVM Learning Practical, *No. 24, Universität Dortmund, LS VIII-Report, 1998*
- [9] Vilson J. Leffa, Clause processing in complex sentence. *Proceedings of LREC'98, Granada, Spain, 1998*
- [10] Matthew Crocker, *Mechanisms for Sentence Processing*. (Garrod & Pickering (eds), Language Processing, Psychology Press, London, UK, 1999)
- [11] Unified Medical Language System <http://umlsinfo.nlm.nih.gov/>
- [12] UniProt, The universal protein resource, <http://www.expasy.uniprot.org/>
- [13] The Gene Ontology, <http://www.geneontology.org/>
- [14] The Stanford Group <http://nlp.stanford.edu/software/index.shtml>