Developing natural language understanding applications for healthcare: a case study on interpreting drug therapy information from discharge summaries.

Werner Ceusters[1], Jan Lorré[1], Annelies Harnie[2], Bert Van Den Bossche[3], Georges Demoor[4]

[1]*Language and Computing nv, Zonnegem, Belgium*
[2]*University of Ghent, Faculty of Medicine, Ghent, Belgium*
[3]*Free University of Brussels, Faculty of Medicine, Brussels, Belgium*
[4]*Dept of Medical Informatics, University Hospital, Ghent, Belgium*

Computers perform best when dealing with structured information. People however communicate by means of natural language. Without proper mechanisms in place, this is very difficult to be understood by machines. Hence most applications use data entry forms combined with picking lists and point and shoot interfaces to get the data structured from the very beginning. In this paper, we describe the FreePharma software component that generates a structured XML representation out of drug therapy information expressed in natural language. Evaluated as a chunk parser, FreePharma performs with an overall recall of 96.3% and precision of 98.2%. As a full sentence parser, it shows recall and precision values of 85.2% and 94.7% respectively. It is shown that the problems related to such an endeavor are not just limited to the design of the natural language understanding component as such, but that integration in clinical host applications is a challenge on its own.

Introduction

Though most clinicians and other healthcare workers are gradually becoming convinced of the advantages of using computers, they still prefer to retrieve data stored by others, than to register data themselves. There are many reasons for this such as unavailability of systems at the point of care, incomplete integration in the primary care process, or the fact that only a subset of the activities for which clinicians would like to have computer support, are actually offered.

The need for structured data

The issue that deserves our particular attention in this paper is the *information structuring bottleneck*. Healthcare records, whether on paper or in computers, are originally kept as an external record for individual patient histories, such that future decisions can be based appropriately on past events. Electronic patient record systems have additional advantages over paper-based systems in their ability to allow for cross-patient studies, and to provide active decision management functionalities. While the former requires thorough structuring of the data inside the machine, the latter also requires representing and storing knowledge and information in the machine so that the machine *itself* can manipulate it, at least for tasks for which it is better suited than humans.

The need for structured *data representation and storage* being undeniable and very well understood, the need for structured *data entry* seems to be the logical consequence. This is at least the impression that we get from analysing the data acquisition interfaces of so many electronic healthcare record systems. There is structuring at the level of the data capture modalities such as rigorous data entry forms, point and click interfaces, structured menu's, etc. There is also structuring at the level of content by using coding and classification systems or controlled vocabularies. The question should be whether or not it is necessary to require the structuring be done by the user. Or as Tange et al. phrase it: "*Initiatives to facilitate the entry of narrative data have focused on the control rather than the ease of data entry*" ([1], p. 24). It is a fact, that most users don't like structured data entry at all, but that many accept it in the light of the benefits obtained when retrieving information. They accept the burden of structured data entry as the price to be paid for powerful information retrieval. But is this price affordable, let alone justifiable ? Many clinicians share the view that faithful recording of patient data can only be achieved by using natural language. This was already stated in the early eighties by Wiederhold who claimed that *the description of biological variability requires the flexibility of natural language and it is generally desirable not to interfere with the traditional manner of medical recording* [2]. Besides this theoretical and fundamental position in favour of natural language registration, there is also a practical reason: data entry by means of continuous speech recognition (CSR). CSR technology has now reached a functional threshold in transforming a speech signal into digital text what is all that is needed for dictation. However, inexperienced users quickly might infer from this evolution that all data entry could be done by voice, freeing them from the need to use a keyboard. Despite this demand, CSR is not that easy lined up with structured data entry forms or cascaded menu's. The command and control paradigm for navigating through forms and menu's is only acceptable in a "hands free" situation, but even that still requires visual feedback from the screen. The ideal situation would be one in which users can enter information or issue queries in natural language, upon which the machine would analyse and structure the input automatically. This calls for advanced natural language understanding.

Structuring drug prescription information

Electronic healthcare record systems tend to have a module for registering patient medication. The most advanced systems use detailed data entry forms in which the name of the drug, its strength, the pharmaceutical form, the dose prescribed, the frequency, the duration and the relationship with the meal to mention just a few, are to be filled out in different boxes. Once done, the clinician can generate a prescription on paper in the format legally required. Some systems also compare the data entered with a knowledge base to verify whether the doses prescribed are not too high or too low, whether incompatibilities may occur, or whether side effects are to be expected given the previous patient history. In addition, the clinician can query the database afterwards to find out what drugs have been prescribed in the past. All this is possible thanks to a deep structured and coded representation of the data.

Unfortunately, keeping the database consistent is not that easy a task in an out-clinic environment. In Belgium for instance, general practitioners (GPs) function as "gate keepers" of the healthcare record. If patients seek specialist advice, the GP is later informed by letter on the findings, conclusions, and therapy given or proposed. If the GP uses an electronic healthcare record, he can request a digital version of such a report or discharge summary, which usually consists of a document generated using a word processor, hence in free natural language. Messaging software takes care of sending the report over a private digital network from the medical specialist to the GP, and of classifying it automatically in the right place of the electronic healthcare record of the patient.

If that letter contains drug prescription information, the structured drug prescription database of the host system is not anymore accurate. GPs could take the burden to manually retype or cut and paste the information (present in the letter in the form of free natural language) to the different input boxes, but there is no need to say that this actually does not happen. "Don't ask to much !", is the - fully understandable - remark that univocally comes out.

This situation, together with the perceived need for speech enabled data entry, was an incentive to create a software plug-in (FreePharma) that analyses drug prescription information expressed in free natural language, and that structures it automatically for subsequent integration in host applications.

Materials and Methods

Corpus collection

A representative number (1849) of Dutch discharge summaries containing drug prescription information, were collected from five different pneumologists. The corpus was randomly divided in a training corpus of 1700 reports, and a test corpus of 149 reports. Only the training corpus was used to develop FreePharma, while the test corpus was set aside for testing purposes afterwards.

Semantic corpus tagging

Two fourth year medical students, informed about the development process, briefly trained in tagging drug prescription information and supervised by a GP with profound knowledge in programming, manually tagged the relevant sections of the training corpus. An initial set of semantic tags was defined by the supervising GP on the basis of a preliminary study on the use of drug prescription data components in existing electronic healthcare record systems. During the tagging process, the set was gradually extended when necessary. On such occasions, the part of the corpus already tagged was reviewed for occurrences that might require the newly added tag. This updating procedure was not applied when later the test corpus was tagged using the final set of semantic tags that arose from the training corpus. When relevant information was encountered that could not be tagged using the tag set, it remained untagged.

As in previous projects [3], we used the internally developed Cassandra® syntactic-semantic tagging technique to annotate the corpus [4]. The purpose of the Cassandra tagging technique is to re-introduce in an explicit and formal way the links between a predefined semantic model and the surface language [5]. The technique is also used to annotate parallel corpora of medical texts in different languages for marking similarities independent of a specific grammar formalism [6].

Table 1 gives an overview of the initial semantic tag set.

Grammar development

The tagged sections from the training corpus were then further processed by a language engineer to develop a lexicon and a suitable grammar. The grammar was built taking into account the various ways in which the semantic units of drug prescription information were found to be "internally" grammaticalised in the training corpus (see Table 2 for an example). The grammar accounted also for possible untagged components between the semantic units, and hence can be seen as a collection of templates.

Table 1 - Semantic tags used in manual corpus tagging

Tag	Description
(MED/FAV)	Pharmaceutical form of drug
(MED/FN)	Manufacturer name
(MED/GN)	Generic drug name
(MED/PN)	Brand name
{MED/AT}	Number of administration units
{MED/PPD}	Product units per dosage
{MED/RM}	Relationship with meal
{MED/TD}	Administration route
{MED/TE}	Administration unit
{MED/TF}	Administration frequency
{MED/VD}	Unit dose

The collection of templates was then transformed to make the resulting grammar suited for immediate processing by a robust finite state chunk parser.

Integration in an electronic healthcare record system

The parser was packaged in an API provided with a set of functions, the most important ones to accept a string containing drug prescription information from a host application, and to retrieve the result in XML format. The XML tags used correspond to the semantic representation tags of the grammar.

At the level of the host application, the XML tags were lined up with the column names of the relevant tables in which patient specific drug prescription information must be stored.

The part of the user interface dealing with the visualisation of incoming electronic reports and discharge summaries was slightly modified such that users could select the relevant paragraphs and with a button click activate the parser. This triggers the normal data input form to be displayed with the relevant boxes filled by information coming from the free text analysis. After verification and, when needed, making the necessary changes, the user can then click the OK button to accept the result.

Results

Evaluation of the system was done as closely as possible according to the *Criteria for Performing an Evaluation Study of a Natural Language System* as described in [7]. Criterium 15 and 21 on inter-rater variability and expert disagreement respectively were not applicable because consensus among both human taggers was required before a tag could actually be given.

Parsing results

From the 149 reports, 425 drug prescription information sentences were tagged manually as described above. After verification, 189 turned out to be syntactically and semantically different from each other and were used for evaluation of the parser. In other words, sentences in which just the name of the drug and/or the administration dose or strength were different, were not considered to be different sentences.

For each individual sentence, it was checked whether individual chunks (i.e. the sentence constituents being manually tagged according to table 1) were tagged correct, erroneous, or untagged by the FreePharma parser. The results are shown in Table 3. The first column shows the tags been studied. The second column gives the number of times the tag had to be found in the corpus. The 3rd, 4th and 5th column show the number of chunks that were correctly, erroneously, or not at all labeled with the expected tag. The last two columns show recall and precision based on the values in the other columns.

Table 2 - Some grammatical realisations of the semantic tag {MED/RM} (translated from Dutch)

Syntactic representation	Semantic representation	Example
{rm}	{rm}	having fasted
{prep-pre}{rm-meal}	{rm-prep-pre}{rm-pre-meal}	before breakfast
{value}{time-unit}{prep-pre}{rm-meal}	{rm-prep-spec}{rm-prep-unit}{rm-prep-pre}{rm-pre-meal}	30 minutes before dinner
{prec-ind}{value}{time-unit}{prep-post}{rm-meal}	{rm-prec-ind}{rm-prep-spec}{rm-prep-unit}{rm-prep-post}{rm-post-meal}	exactly 30 minutes after dinner

Table 3 - Results of the FreePharma parser

Tag	Occ	Right	Wrong	Untagged	Recall	Prec
(MED/FAV)	122	119	1	2	97,5	99,2
(MED/FN)	2	2	0	0	100	100
(MED/GN)	6	3	0	3	50	100
(MED/PN)	183	183	0	0	100	100
{MED/AT}	174	168	6	0	96,6	96,6
{MED/PPD}	23	17	6	0	73,9	73,9
{MED/RM}	78	75	0	3	96,2	100
{MED/TD}	28	25	0	3	89,3	100
{MED/TE}	169	166	1	2	98,2	99,4
{MED/TF}	145	143	2	0	98,6	98,6
{MED/VD}	60	58	2	0	96,7	96,7
missing tags	6	0	0	6	0	N.A.
Total chunks	996	959	18	19	96,3	98,2
Sentences	189	161	9	19	85,2	94,7

A detailed calculation at the level of the individual syntactic semantic tags (73 in total), has not yet been done. It is however important to note that when at least one element out of the internal semantic representation of a chunk was found to be erroneous, the complete chunk was considered to be erroneous. Evenly, when at least one chunk out of a sentence turned out to be wrong, the complete sentence was considered to be wrong. This explains why the overall percentage of correct parses at sentence level is only 85,2% as compared with the mean percentage of 96,3 % at chunk level. When more than one chunk in a sentence turned out to be wrong or untagged, the sentence was only once marked as being wrong. As such, the 37 incorrectly or untagged chunks occurred all together in 28 sentences.

Integration results

From the software engineering point of view, no problems were encountered to integrate the DLL and to achieve communication with the host application and vice versa.

More “semantic” problems were encountered when translating the XML tags in the parser’s output, to the datadictionary of the host application. This is discussed later.

After integration, processing time for even complex sentences turned out to be satisfactory. The parser itself processes sentences in the order of 0.05 - 0.1 second per sentence on a Pentium II 350 MHz. Analysis of the generated XML result by the host application and visualisation of the forms took about 0.5 sec per sentence.

Discussion

Partial parsing

Partial parsing or *chunk parsing* is considered to be a valuable strategy for bootstrapping broad coverage parsers from text corpora where first a tagger is induced from word distributions, a chunk parser from a tagged corpus and lexical dependencies from a chunked corpus [8]. A chunk is defined as the non-recursive core of major-phrases such as NP, VP, PP, AP or AdvP [9].

The technique is also convenient when a complete analysis of sentences is not required for solving a particular problem such as in information retrieval where detecting noun phrases is an important step in finding adequate indexing terms for documents [10, 11]. It has been shown that systems designed for detecting noun phrases in general language perform worse when used in the medical domain without modification. In [12], CLARIT’s and the Xerox Tagger’s ability to identify simple noun phrases in medical discharge summaries was tested. In twenty randomly selected discharge summaries, there were 1909 unique simple noun phrases. CLARIT and the Xerox Tagger exactly identified 77.0% and 68.7% of the phrases, respectively, and partially identified 85.7% and 80.8% of the phrases.

Partial parsing seems to be a broadly applied technique in medical natural language understanding with reasonable results when they are designed for medical sublanguage, and with a precise task in mind. The LSP system was originally designed to extract factual information from medical reports [13]. The medical sublanguage is viewed by this system as consisting of 6 information formats onto which a total of 54 semantic classes (represented in the lexicon) can be mapped. The system uses a parser that can deal with incomplete analyses. When it was used to find 13 important details of asthma management in a total number of 31 discharge summaries (testing set), recall appeared to be 82.1% (92.5% counting only omissions instead of errors) and precision 82.5% (98.6% idem) [14].

Haug et al. report recall and precision rates of 87% and 95% for detecting clinical findings in 839 chest x-ray reports by using SPRUS [15], and rates of 95% and 94% respectively for the detection of diagnoses [16]. SPRUS is mainly semantically driven, and is not able to exploit syntactic information. Hence complex noun-phrases and whole sentences cannot be processed.

The natural language processing tool SymText was used by Fiszman et al. to extract relevant clinical information from Ventilation / Perfusion lung scan reports. The overall precision was 88% and recall was 92% [17].

The CAPIS system was able to recall 92% of the relevant physical findings (156 in total in 20 reports on patients with gastro-intestinal bleeding), with a precision of 96% [18]. CAPIS uses a finite-state machine parser that is specifically designed for the more structured parts in medical narrative such as the clinical findings section.

Being aware of the dangers in comparing systems if the complexity of the task of the language processor is not exactly the same [7], we feel nevertheless comfortable with the recall and precision values as found in table 3.

The most important mistake committed by FreePharma turned out to be identifying the strength of a medication (such as in the sentence "Clamoxyl 500 tablets 3 times a day for 2 weeks") as an administration number and unit, or the other way round. This happened 6 times (though not always with the same disastrous effect if the patient actually would take 500 tablets three times a day). The phrase given as an example is not the best way to express the intended meaning, but if some physicians really express themselves in this way, and if the intended meaning can be captured correctly by a colleague, we should try to make such sentences also understandable by machines. This particular problem can be solved by giving the system more information on the various forms in which specific medications can be found on the market. This is further addressed in the section on integration. Note however that these kinds of mistakes in FreePharma only come up in case of ill-formed sentences and that mechanisms for distinguishing between strength of a medication and dose administered are in place. This is not the case for instance with the LSP-processor presented in [13] where these types of errors are systematically due to underspecification of the semantic tags. In [13, p 301] it is shown that the sentence "Catapres 0.2 mg tab, 4 tabs tid" is analysed as : (MEDICATION (MED = catapres) (RXDOSE (NUM = 0.2) (UNIT = milligram)) (RXDOSE (NUM = 4) (UNIT = tablet)) (RXFREQUENCY = "tid")).

RXDOSE is clearly an underspecified tag being used for both the administration dose and the strength of a medication. Underspecification of this kind causes problems to host applications where both types of information must be stored in a different field.

Another 6 mistakes had to do with tags that were not found in the training corpus (or not adequately tagged), but occurred in the test corpus. This was the case for the number of tablets in a drug package, as well as for some very specific sentence fragments such as "in gradually smaller doses over time", or "in addition to the medication currently taken".

The fact that recall and precision for drug product names are both 100% is no surprise. Drug names were added to the FreePharma lexicon from a drug dictionary, such that more drugs than just those encountered in the training corpus could be identified.

Integration issues

The FreePharma component was designed in a strictly controlled environment, such that few surprises were to be expected with respect to its natural language understanding capabilities. Integration of the tool in "foreign" host applications is a completely different issue.

To start with, the terminology used in FreePharma has to be mapped to the terminology of the host application. The system is shipped with an API and software development kit such that developers of host applications can easily interface their drug database with FreePharma. As such, all the drugs known by the host application can be represented in FreePharma by means of the unique DrugId used in the host application. This has as consequence that the practical recall of drug names in a host application may be less than 100%, or that FreePharma (using its own lexicon) generates output in which drugs are referred to by name, instead of the unique DrugId of the host application.

Next, the non-drug name related information generated by FreePharma must be dealt with.

First, the semantic tags of the FreePharma output (minimally at the level of specification as in Table 1, ideally as in Table 2) have to be mapped to the data dictionary of the database of the host application. Though standards for data dictionaries over different electronic healthcare record systems are already proposed [19, 20], they are not yet fully implemented. The consequence is that not for all tags proposed by FreePharma an equivalent field can be found in the database of the host application. In such a situation, the developer of the host application, when integrating FreePharma, must take a decision on what would be an appropriate action: throwing that particular piece of information away, storing it in a free text field when available, or change the design of the database.

Second, where mapping semantic tags to the data dictionary is an activity at the level of the "items" [19], there is the issue of the "values" for these items. From the linguistic point of view, one can easily shift detail back and forth between the tag itself and its value. In an extreme situation, the notion of "value" could possibly disappear, such that all semantic information would be stored in the tags, while the values just represent the actual linguistic forms used in the utterance. One can indeed represent "breakfast" in utterances such as "1 tablet half an hour before breakfast" as EVENT = "dinner", or MEAL = "dinner", or DINNER = "dinner". Electronic healthcare record systems on the market have made particular choices here, and as one can expect, all of them in different ways. A good reference terminology with formal compositional power would be nice on the condition that it is not too normative and that it takes cognitive issues into account on how knowledge is actually communicated and used [21].

Also, we tend to become rather skeptic about prescriptive claims in that "*the medical record should capture in a structured form all of the clinically significant information in the narrative notes, where by 'clinically significant' we mean the information which is within the medical domain rather than the domain of everyday life - e.g., 'aggravated by cold' rather than 'comes on* *when passing the freezer section of the supermarket*.'" [22, p 109].

We do agree with the need for *structured data storage*, but not with *structured data capture*. As such we believe that the *predictive data entry* paradigm [23, 24], in which data entry is guided (hence restricted) by a model, is indeed a slight improvement over the use of unmanageably large controlled vocabularies, but nevertheless also a temporal solution in absence of adequate natural language understanding applications.

It can also be questioned whether or not the above claim is biased by the evident reality that the development of large scale lexicons for unrestricted data input over natural language understanding applications is much more complex than for controlled data entry, or even a post-hoc argument for not being exhaustive as far as the model is concerned. In [3], we showed that half of the SNOMED International V3.1 labels used in the experiment could not be mapped by our natural language understanding system to the GALEN model, because of information missing in the GALEN model version of that time. The positive conclusion in that paper was that NLP indeed can assist in finding new information, hence in building more adequate and complete domain models. However, the same conclusion could be rephrased to state that domain models exclusively designed according to the principles described in [22, p 109] are of little use in NLU systems "at runtime", and that more attention must be given to cognitive aspects in dealing with terminologies when building such models [21].

Less relevant for this paper, but definitely extremely relevant for patients, there are also pure epidemiological reasons to fight the above cited position on what it means to be "clinically significant". E.g. if this advise was followed, some recent outbreaks of Legionellosis with nearly 40 mortal cases in Belgium and The Netherlands would not have been identified as being due to public trade exhibitions in which water fountains and bubble baths were used, such that effective measurements to prevent this in the future would not have been taken by the Ministries of Health of both countries.

Expanding FreePharma's applicability

After modifying FreePharma based on the evaluation described above, other adaptations were realised.

Porting to French

The system is based on Dutch discharge summaries, but nevertheless designed following our "linguistic ontology" approach [5]. Where developing the Dutch version took about 6 man-months, porting to French was just a matter of days. The initial port could be realised fully automatically and yielded in overall performance rates of 89,2% recall and 91,0 % precision at chunk level, and 73,8% recall and 82,5% precision at sentence levels. Three days were needed to adapt some lexical entries and specific parts of the grammar. An in-depth evaluation after these manual modifications has not yet been conducted but preliminary tests indicate that behaviour will not be worse than for Dutch.

The French version has also been integrated in the same host application as the Dutch version. Users can chose two operation modes. Either they specify in which language (French or Dutch) the text to analyse is expressed, resulting in processing speeds as described earlier, or they just let the system find out what language is used. This almost triples processing time to an average of 0.15 - 0.3 seconds per sentence, what is still acceptable.

Dictating medical prescriptions using speech recognition

Somewhat more disappointing - at least at first glance - was a first, rather naïve, attempt to use FreePharma directly as a tool to dictate prescriptions after integration with speech recognition software. The idea was to allow users to dictate medical prescriptions such that the speech recognition software would turn the speech signal into digital text, and FreePharma subsequently would analyse the digital text to generate the structured representation. This would free the user from typing all relevant information in the relevant boxes of a structured input form, but still enjoy the advantages of such a structured input, e.g. with respect to detecting contra-indications [25], wrong doses [26], or adverse drug events [27]. It would also be a tremendous improvement of the "command and control" paradigm used in speech recognition in which speech commands are used to navigate through structured entry forms instead of the mouse or tab-key.

Adapting the lexicon and language model of a commercially available general purpose speech recognition system for our purposes was easy. The next step was however less straight forward. We discovered very quickly that the old notion of "*sublanguage*", being defined as a natural language used in a particular semantic domain for a specific purpose, and differing from a "general" natural language by being restrictive, deviant and preferential with respect to vocabulary, syntax, semantics and pragmatics, is still valid [28]. As explained above, FreePharma's grammar was based on templates extracted from discharge summaries in which medical specialists reported on what medications, including doses, administration route, etc., they had proposed or prescribed to patients. This grammar actually is similar, though not the same as the one used by physicians when dictating such prescriptions on the spot. In sublanguage terms: the semantic domain is the same, but the purpose for which language is used in this setup, is different !

The solution was to register a number of actual prescriptions, and then to modify the grammar for this purpose. This was not difficult because modifications were almost exclusively required at the level of chaining chunks, and only in a few cases at the level of the internal representation of the chunks.

Towards a reliable natural language understanding system

Though the results of FreePharma are good, the system is not perfect. A relevant question is whether physicians can afford to use systems that are not perfect ? Developers tend to say that this is not that much of a problem if users always have the possibility to verify the results before they are written to the database. Though this might be true, we still see it as trying to escape from certain responsibilities. In our view, the solution can be found in clearly separating usefulness and perfection. What is extremely encouraging in the FreePharma results presented above, are the very high values for precision, except for the semantic tag "strength of medication". A precision of 100% effectively means that if the system provides a result, one can be sure that the result is correct. A reliable system can hence be defined as a system with 100% precision, but eventually sacrificing on recall. This means that for certain tasks, it might be better to design a system in such a way that it can solve a problem with 100% certainty in 80% of cases, than to make it solve a problem in 95% of cases, but that it is not possible to tell what 5% are actually the wrong solutions. This is one precise direction in which future work with FreePharma will be conducted.

Conclusions

Generalizing from our experiences with FreePharma and reviewing some of the systems described in the literature, we argue that specific natural language understanding components are technically feasible today and that they are mature enough to be integrated in electronic healthcare record applications. With "specific" we mean that they must have been designed within a close semantic domain, and with a specific purpose in mind. They should not be built on the basis of existing reference terminologies, but should start from language corpora collected for the task at hand. Reference terminologies, just as standards for electronic healthcare record architectures, do have a place to bridge the gap between the output of language processors and the terminology of the host application. An important condition however is that such reference terminologies are not too normative or prescriptive.

We came close to develop a system that reliably can transform free text drug prescription information into a structured representation. We were able to port it with little effort to another language and another task. We are aware that this is just one tiny task in understanding the natural language contents of a medical record under a "narratological framework" as proposed by Kay and Purves [29], or to base data entry and information retrieval completely on natural language. However, just as the "general problem solver" [30] turned out to be an unreachable dream in the long run, a "general unrestricted text understanding system" is probably also doomed to fail, unless it is built out of many much smaller components that each independently fulfill a specific language understanding task. Making such components cooperate when developed by various authors will require similar efforts in linguistic ontology sharing, as are currently being conducted related to medical domain ontologies.

References

1. Tange HJ, Hasman A, de Vries Robbe PF, Schouten HC. Medical narratives in electronic medical records. International Journal of Medical Informatics, 1997, 46(1): 7-29.
2. Wiederhold G. Databases in healthcare. Stanford University, Computer Science Department, Report No. STAN-CS-80-790, 1980.
3. Ceusters W, Rogers J, Consorti F, Rossi-Mori A. Syntactic-semantic tagging as a mediator between linguistic representations and formal models: an exercise in linking SNOMED to GALEN. Artificial Intelligence in Medicine 1999; 15: 5-23.
4. Ceusters W, Waagmeester A, De Moor G. Syntactic-semantic tagging conventions for a medical treebank: the CASSANDRA approach. In van der Lei J, Beckers WPA, Ceusters W, van Overbeeke JJ (eds.): Proceedings MIC'97, Veldhoven, The Netherlands, 1997, 183-193.
5. Ceusters W, Buekens F, De Moor G, Waagmeester A. The distinction between linguistic and conceptual semantics in medical terminology and its implications for NLP-based knowledge acquisition. Methods of Information in Medicine 1998; 37(4/5):327-33.
6. Wermuth C, Ceusters W, Decruyenaere J. A contrastive analysis of the semantic and syntactic structures of surgical procedure expressions and empirical validation of the Cassandra tagging technique. In: Contrastive Linguistics and Translation Studies: Empirical Appoaches, Université Catholique de Louvain, 5-6 February 1999 (in press).
7. Friedman C, Hripcsak G. Evaluating natural language processors in the clinical domain. In Chute CG, Baud RH, Cimino JJ, Patel VL, Rector AL. (eds). Special Issue on Coding and Language Processing. (Proceedings of the 1997 International Medical Informatics Association (IMIA) Working Group 6 Conference on Natural Language and Medical Concept Representation. Methods of Information in Medicine 1998;37(4/5):334-344.
8. Abney S. Partial parsing via finite-state cascades. In Proceedinsg of the ESSLLI'96 Robust Parsing Workshop, 1996.
9. Abney S. Parsing by chunks. In Berwick R, Abney S, Tenny C (eds.) *Principle-Based Parsing*, Kluwer Academic Publishers, Dordrecht/Boston/London, 1991, 257-278.
10. Evans DA, Lefferts RG. Design and evaluation of the CLARIT TREC-2 system. Proceedings of the Second

Text Retrieval Conference (TREC-2), Gaithersburg, MD, 1993, 137-150.

11. Strzalkowski T, Carballo JP, Marinescu M. Natural language information retrieval: TREC-3 Report. Overview of the Third Text Retrieval Conference (TREC-3), Gaithersburg, MD, 1994, 39-53.

12. Spackman KA, Hersh WR. Recognizing noun phrases in medical discharge summaries: an evaluation of two natural language parsers. Proc-AMIA-Annu-Fall-Symp. 1996: 155-8.

13. Sager N, Friedman C, Lyman MS: Medcial Language Processing: Computer Management of Narrative Data. Reading, MA: Addison - Wesley, 1987.

14. Sager N, Lyman MS, Bucknall C, Nhan N, Tick LJ: Natural language processing and the representation of clinical data. J. Am Med Infomatics Assoc, 1994, (1), 142-160.

15. Haug PJ, Ranum DL, Frederick PR: Computerized extraction of coded findings from free-text radiologic reports. Radiology, 1990 (174); 543 - 548.

16. Ranum DL. Knowledge based understanding of radiology text. Proceedings of the 12th Annual Symposium on Computer Applications in Medical Care, Washington, DC, 1988, 141 - 145.

17. Fiszman M, Haug PJ, Frederick PR. Automatic extraction of PIOPED interpretations from ventilation/perfusion lung scan reports. Proc-AMIA-Symp. 1998: 860-4.

18. Lin R, Lenert L, Middleton B, Shiffman S. A free-text processing system to capture physical findings: Canonical Phrase Identification System (CAPIS). Proc-Annu-Symp-Comput-Appl-Med-Care. 1991: 843-7.

19. CEN ENV 12265:1997. Medical Informatics - Electronic Healthcare Record Architecture, Brussels; CEN 1997.

20. Devlies J. The Belgian Prorec Medical Data Dictionary. MS-Huge Conference, Bruges, 3-4/12/1999 (in press).

21. Patel VL, Kushniruk AW. Understanding, navigating and communicating knowledge: issues and challenges. In Chute CG, Baud RH, Cimino JJ, Patel VL, Rector AL. (eds). Special Issue on Coding and Language Processing. (Proceedings of the 1997 International Medical Informatics Association (IMIA) Working Group 6 Conference on Natural Language and Medical Concept Representation. Methods of Information in Medicine 1998;37(4/5): 460-70.

22. Rector AL, Nowlan WA, Kay S, Goble CA, Howkins TJ. A framework for modelling the electronic medical record. Methods of Information in Medicine 1993;32: 109-119.

23. Nowlan WA, Rector AL. Medical knowledge representation and predictive data entry. In: Proceedings of AIME 91. Lecture notes in Medical Informatics No 44, Stefanelli M, Hasman A, Fieschi M, Talmon J (eds.). Berlin: Springer, 1991: 105-106.

24. Kirby J, Cope N, Souza AD, Fowler H, Gain R. The PEN & PAD Data Entry System. In Brender J, Christensen J, Scherrer J-R, McNair P (eds.) Medical Informatics Europe (MIE 96), Copenhagen: IOS Press 1996: 430-434.

25. Liu JH, Milstein C, Sene B, Venot A. Object-oriented modeling and terminologies for drug contraindications. Methods of Information in Medicine 1998; 37(1): 45-52.

26. Bates DW, Teich JM, Lee J, Seger D, Kuperman-GJ, Ma'Luf N, Boyle D, Leape L. The impact of computerized physician order entry on medication error prevention. Journal of the American Medical Informatics Association 1999; 6(4): 313-21.

27. Kuperman GJ, Cooley T, Tremblay J, Teich JM, Churchill W. Decision support for medication use in an inpatient physician order entry application and a pharmacy application. Medinfo 1998; 9 Pt 1: 467-71.

28. Kittredge R., Lehrberger J. (eds.): Sublanguage: studies of language in restricted domains. de Gruyter, Berlin, 1982.

29. Kay S, Purves IN. Medical Records and Other Stories: a narratological framework. Methods of Information in Medicine 1996; 35: 72-87.

30. Newell A, Simon HA. GPS, a program that simulates human thought. In: Feigenbaum EA, Feldman J (eds.) *Computers and Thought*, McGraw-Hill, New York, 1963.