Extraction of Medical Information Using CRFs, Context Patterns, and Dependency Parse Trees

Hui Yang, Anne de Roeck
Department of Computing, Open University, UK

Abstract

This paper describes an information extraction (IE) system for the identification of medical concepts (Task 1), assertion classification on medical problems (Task 2), and the extraction of relations between medical concepts (Task 3) developed in the context of the 2010 i2b2/VA NLP Shared Task for medical information extraction. We define a wide variety of linguistic and semantic features for the CRFs-based medical concept recognition. A context-pattern based approach is explored for assertion annotation and relation extraction, which in particular makes extensive use of hierarchical structures of dependency parse trees to capture the complicated syntactic relations underlying in assertion/relation statements.

Introduction

In order to drive forward research in medical information processing on clinical narratives, the 2010 i2b2/VA NLP Shared Task[1] concerned extraction of medical information from patient discharge summaries and progress notes. To facilitate evaluation on different aspects of medical information, the challenge consists of three tasks: Task 1 focused on extraction of medical problems, tests, and treatments. Task 2 concerned classification of assertions made on medical problems. Task 3 was a more advanced task focused on extraction of relations built on medical problems, treatment, and test concepts.

Our team participated in all of the three tasks of the Shared Task regarding medical information extraction. In this paper, we describe an information extraction (IE) system that tackles with three typical natural language processing (NLP) issues in the biomedical domain, namely, named entity recognition (NER) (Task 1), assertion annotation (Task 2), and relation extraction (Task 3) in the biomedical domain. Task 1 was approached as a classification problem using efficient, supervised machine learning techniques, Conditional Random Fields (CRFs)[2], which combined with a wide variety of feature set derived from the clinical text. The approaches to both Task 2 and 3 were characterized by heavy reliance on context patterns that were viewed as assertion cues (i.e. terms indicating an assertion category) or relation triggers (i.e. terms describing the relation between a given pair of medical concepts). Moreover, the approaches made extensive use of hierarchical structure of dependency parse trees[3, 4] derived from the Stanford Lexicalized Parser[5]. Hierarchical structure of dependency parse trees not only limits the scope of assertion/relation statements but also provides syntactic dependency paths between assertion cues and medical problems (Task 2) or those among relation triggers and their associated relation participants (Task 3). Such characteristics are particularly helpful to deal with long and complicated sentences that frequently occur in clinical texts.

Methods

Data Preprocessing

Each input clinical file was first chunked into six predefined section categories (e.g., diagnosis, examination, medication, etc) based on the surface features of the headings. Then the text was decomposed into text lines since the file was formatted to have one sentence per line. For each sentence line, tokenization, Part-of-Speech (POS) tagging, and Shallow Parsing were employed by the Genia Tagger toolkit[6], and relevant word information (e.g., tokens and their offset positions, POS taggers, basic phrase tags, and lemmas) was stored in the database.

Task 1: Medical Concept Identification

For Task 1, we investigated a CRFs model for the identification of medical concepts, which was implemented in the CRF++ toolkit[7]. CRFs are particularly useful for sequence segmentation labeling tasks, such as Named Entity Recognition (NER)[8].

Our CRFs model was built based on a wide variety of feature set, which not only considered *linguistic* information (e.g., word lemma, POS tagger, basic phrase type), but also made use of *semantic* types (e.g., medical-specific features) assigned to the token. Furthermore, we integrated *local* features of individual tokens with *context* features describing the relations between the candidate token and its neighbor tokens. We explored a window size of ± 3 neighbor tokens.

The properties for each candidate toke, which are used to generate various types of features, are described below:

I. Linguistic Properties

- Word features: word lemma, POS tagger, phrase type

- Orthographic features: the word form features like AllCaps, digit, punctuation symbols, etc.

- Morphological features: suffixes such as *-ase*, *-zyme*, *-ome*, etc.

II. Semantic Properties

- Stop-word List: stop words from NCBI PubMed[9]

- Entity-specific words: frequent words for each class of medical concepts, which were manually collected from the annotated training data

- Modifier [body/organ]: the words describing human body components or organs (e.g., *knee*, *kidney*, etc.)

- Modifier [direction]: the words indicating the direction such as *bilateral*, *peripheral*, *surface*, etc.

- The vb-ed nouns: the nominalizations of some verbs involving in treatment procedures (e.g., *removal*, *replacement*, etc)

- Modifier [assertion]: the words indicating assertion such as *no*, *possible*, etc.

- Medication-specific classes: the classes related to medication information, such as dosage, mode, frequency, etc.

III. Context Properties

- Section Categories: obtained in the data pre-processing step.

- Section header: judge whether the token is located in the section heading or within the section text.

To compare the performances of different CRFs classifiers, the binary CRFs classifier and the multi-class CRFs classifier, we constructed three groups of CRFs classifiers:

(a) Three separate binary CRFs classifiers, each of which was targeted to one specific medical entity type. The final integrated results were obtained by combining the results of the three binary CRFs classifiers.

(b) A multi-class CRF classifier with *forward* chunking model (i.e. left-to-right BIO representation)

(c) A multi-class CRF classifier with *backward* chunking model (i.e. right-to-left BIO representation)

We submitted three system runs for Task 1, each of which was the annotation results generated by one of the above CRFs classifiers.

Task 2: Assertion Annotation for Medical Problems

Task 2 is to classify each medication problem into one of six assertion categories, namely, Present, Absent, Possible, Conditional, Hypothetical, and Associated-with-Someone-Else. For the purpose of effective evaluation and analysis on both Task 2 and Task 3, it is assumed that the medical problem (P), treatment (Tr), and test (Te) concepts have already been marked in the input text files. In order to simplify sentence analysis, we first replaced the medical concepts with the concept-type names with the corresponding concept IDs. For example:

E1.1 '*A CT showed soft issue swelling obliterating the airway surrounding the ETT tube .*' (Original)

E1.2 '*Te_1 showed P_2 obliterating the airway surrounding Tr_3 .*' (Replaced)

In Examples E1.1 and E1.2, the medical concepts, *A CT*, *soft issue swelling*, and *ETT tube* are substituted with the corresponding concept IDs, *Te_1*, *P_2,* and *Tr_3*, respectively. We observed that the concept replacement step has two advantages: (a) it introduces semantic information (i.e., medical concept types) as the surface features of the sentence. (b) it has a potential of reducing the complexity of the sentence structure, particularly for the long sentences that contain a number of multi-word medical concepts, thus increasing the accuracy of the full parsing information by the Stanford Parser. Noted that both the original sentence set and concept-replaced sentence set will be used in Task 2 and the latter Task 3 depending on different scenarios.

The work on assertion annotation, in practice, consists of three main steps:

(1) Detect the sentences that contain relevant assertion cues (i.e. terms indicating an assertion category).

(2) Determine the scope of assertion statement within a given sentence

(3) Find the associated problem concept(s) in the scope of assertion statement.

A. Assertion Cues

Several classes of assertion cues have been determined based on corpus analyses, which are described below:

(a) Concept-related cues: the meaning of the problem concept contains somewhat assert clue, e.g., the concepts, *nontender*, *nka*, for *Absent* assertion.

(b) Section-related cues: some section categories imply the likelihood of some certain assertions. For example, the problem concepts appearing in the text of the *Medication* section category are likely to be assumed as *hypothetical*, e.g., '*Aspirin 325 mg q.d. for pain*'.

(c) Lexical cues: for example, the words, *no*, *nor*, *without*, for *Absent* assertion; and the words, *versus*, *vs*, for *Possible* assertion.

(d) Context cues: this type of cues comprises the majority of the assertion cues. Such cues mainly focus on noun phrases (NPs), verb phrases (VPs), adjective phrases (APs), and preposition phrases (PPs), e.g., *no evidence of*, *rule out*, *negative for*, and *with resolution of*, etc.

(e) Sentence-level cues: such cues always connect to some kind of clause sentences. For instance, the problem concepts, which appear in the cause sentences with subjunctive mood (e.g., *if*, *unless*, etc), are usually classified into the *Hypothetical* category.

We manually collected a set of assert cues based on their *goodness* scores. For a given assertion category, *C*, and an assertion cue, *a*, the *goodness* score *G*(*a*, *C*) is calculated as

$$G(a, C) = w(C{:}a)/T(a) \qquad (1)$$

where *w*(*C*:*a*) is the number of correctly identified concepts using the assertion cue *a* for assertion category *C*, and *T*(*a*) is the total concept number returned using the assertion cue *a*. If the *goodness* score of an assertion cue is above the given threshold, this cue will be selected. The *goodness* threshold is determined empirically.

B. Detection of Assertion Category

Given an assertion cue (except for the ones in the concept-related and section-related classes), the system searches the concept-replaced sentence set, and selects the sentences whose text contains the assertion cue using exact string matching. The selected sentences are submitted to the Stanford Parser which generates a dependency parse tree for each selected sentence.

We here present a simple example to illustrate how the system uses the dependency parse tree to determine the scope of an assertion statement and to find the required problem concepts. The context pattern that describes an *Absent* assertion clue is represented as the frame format (see Figure 1). It is assumed that the candidate problem concept should be contained in the preposition modifier (PP attachment) of the noun phrase '*no evidence'*.

Context pattern: *no evidence of* [P]
Assertion type: Absent

A. Assertion Cue
Cue Term: no evidence of
Headword: evidence
Headword POS tag: noun

B. Searched Concept
Concept type: Problem
Searched context: right context
Search depth (layer): 4 (from the headword leaf node)

Figure 1. The Frame-based representation of a sample context pattern for *Absent* assertion.

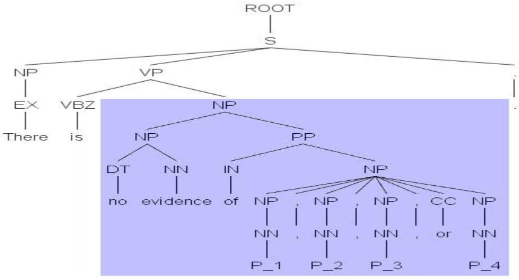

Figure 2. The dependency parse tree for the sentence E2.2 by the Stanford Parser. (The shade area is the chunk dependency tree used as the scope of assertion statement)

Given the dependency parse tree (Figure 2) for the concept-replaced sentence E2.2 for the original sentence E2.1,

E2.1 '*There is* ***no evidence of*** *intraaxial or extraaxial hemorrhage , mass effect , shift of normally midline structures , or acute major or minor vascular territorial infarction .*' (Original)

E2.2 '*There is* ***no evidence of*** *P_1 , P_2 , P_3 , or P_4 .*' (Replaced)

The system begins the search from the start-point, the leaf node of the headword *evidence*, and then tracks upward to the ancestor node (NP) of the fourth layer. The chunk dependency tree (the shade area of Figure 2) associated with the located ancestor node is trimmed from the whole tree as the scope of assertion statement. The system searches the leaf-node sequence of the chunk dependency tree, and extracts a list of candidate problem concepts, P_1, P_2, P_3, and P_4, from the right context of the assertion cue.

It is noticed that the search depth determines the cope size of an assertion statement. In our system, the search depth for each context pattern was decided empirically, which considered several factors, such as the POS tag of the headword of the assertion cue (e.g., in general, the search depth for verb, adjective, and noun is 3, 3, and 4, respectively) and the class types of the assertion cues.

Considering the overriding priority in assertion annotation, the annotation order is from the highest to the lowest, namely, associated-with-someone-else, hypothetical, conditional, possible, and absent. The system favors precision over recall in order to increase the chance for a concept to be classified into a low-priority assertion category.

Task 3: Relation Extraction between medical concepts

The relation extraction task is to identify how problems relate to treatments, tests, and other medical problems in the text. The relation extraction types are grouped into three classes: (a) the relations between treatments and problems, e.g., TrIP, TrWP, TrCP, TrAP, TrNAP. (b) the relations between tests and problems, e.g., TeCP, TeRP. (c) the relation between problems and other problems, e.g., PIP.

The approach to relation extraction is similar to the one for assertion annotation, which draws primarily from context patterns combined with dependency parse trees. However, the contexts for relation extraction are more complicated, because each context involves in the relationship that exists between two medical concepts within one sentence.

The procedure for relation extraction is a pipeline of three major processing steps: the recognition of relation trigger, the determination of the scope of relation statement, and the detection of relevant relation participants (i.e. medical concepts).

A. Context Patterns for Relation Extraction

Similar to assertion cues, we proceeded with constructing a dictionary of relation triggers (i.e. terms describing the relation between the given pair of medical concepts), which drew from the annotated training corpus. Our analysis suggests that, due to the long-range dependency paths between the relation trigger and the associated participants, it appears that more context constraints are needed for the determination of the scope of relation statements.

We group context patterns that reflect the constructions used in the relation statements into several construction classes described below:

I. Short-range dependency with the trigger

(a) Two concepts are separately located in each side of the relation trigger, e.g., [Tr] … *responding to* [P]

(b) Two concepts are located at the same side of the relation trigger, e.g., *tolerating* [Tr] … *without* [P]

(c) One of the concepts can be considered as the implicit relation trigger, e.g., [allergic] … to [Tr]

II. Long-range dependency with the trigger

(d) One of the concepts appears in a preposition phrase (e.g., *Upon*, *Despite*, *Given*, etc.) that is always located at the beginning of the sentence, e.g., *Despite* [Tr], … *continue to have* [P]

(e) One of the concepts appears in a relative clause that modifies a noun phrase in which another concept occurs, e.g., [Tr] *which/that* … *reveal* [P]

III. Non-trigger dependency (special cases)

(f) One of the concepts is located at the beginning of the sentence, and acts as a header, e.g., [P] : … [Tr]

(g) The concepts are immediately adjacent to each other. They are connected either with a preposition (e.g., *for*, *with*, *due to*) or with nothing, e.g., [Tr] *for* [P], [Tr] [P]

B. Recognition of Relation Types

Here we use the context pattern shown in Figure 3 to demonstrate how dependency parse trees can help find the dependency paths between the relation trigger and the associated relation participants.

Context pattern:	[Te] … show [P]
Relation type:	TeRP
A. Relation Trigger	
Trigger term:	show
Headword:	show (lemma)
Headword POS Tag:	verb
B. Searched Concept (NEAR)	
Concept type:	Problem
Searched context:	right context
Search depth (layer):	3 (from the headword leaf node)
Additional constraint:	null
C. Searched Concept (FAR)	
Concept type:	Test
Searched context:	left context
Search depth (layer):	4 (from the headword leaf node)
Additional constraint:	the left NP sibling node closest to the relation trigger

Figure 3. The Frame-based representation of a sample context pattern for the *TeRP* Relation

The procedure to extract the test-problem concept pairs with the *TeRP* relation contained in the sentence E3.1 is described as follows:

E3.1 '*At an outside hospital , The electrocardiogram* ***showed*** *atrial fibrillation with slow ventricular response and digitalis effect as well as left ventricular hypertrophy .*' (Original)

E3.2 '*At an outside hospital , Te_1* ***showed*** *P_2 with P_3 and P_4 as well as P_5 .*' (Replaced)

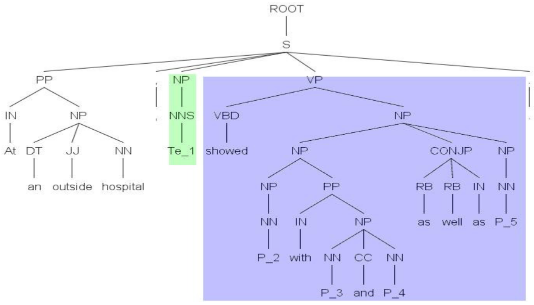

Figure 4. The dependency parse tree for the sentence E3.2 by the Stanford parser where the blue shade area is the chunk dependency tree used as the scope of relation statement for the NEAR concept, and the green shade area used for the search of the FAR concept.

STEP I: the sentence E3.1 that contains the relation trigger *showed* is firstly selected from the candidate sentence set. Then, the corresponding concept-replaced sentence E3.2 is submitted to the Stanford parser to obtain the dependency parse tree (in Figure 4).

STEP II: the system extracts the chunk dependency tree (i.e. the blue shade area in Figure 4) that leads from a start-point (the leaf node with the word of 'showed') to the ancestor node, VP, at the third layer.

STEP III: the system searches the right context of the headword 'showed' in the leaf node sequence of the chunk dependency tree, and see if it contains one or more required NEAR concepts, *Problem* concepts. If the NEAR concepts exist, the system continues to search the dependency parse tree for the FAR concept; otherwise, the search will be stopped and go to the next candidate sentence. For the sentence E3.2, a list of problem concepts, P_2, P_3, P_4, P_5, are extracted from the chunk dependency tree as the candidate NEAR concepts.

STEP IV: the system tracks upward to the ancient node (*S*) at the fourth layer associated with the headword leaf node, and looks for the left *NP* sibling node closest to the relation trigger (i.e. the green shade area in Figure 4). If the *NP* node exists and it contains at least one FAR concept, *Test* concept, it is said that the *TeRP* relation is held in the sentence, and the corresponding concept pairs (i.e. the NEAR and FAR concepts) will be assigned with the relation tag *TeRP*. In the sentence E3.2, four concept pairs, {(Te_1, P_2), (Te_1, P_3), (Te_1, P_4), (Te_1, P_5)} are separately extracted from the sentence, and are marked with the *TeRP* relation tag.

Conclusion

In this paper, we describe an information extraction (IE) system, which was developed for the three tasks of the i2b2/VA NLP Shared Task on medical information extraction, namely, the identification of medical concepts (Task 1), assertion annotation on medical problems (Task 2), and the relation extraction between medical problems, treatments, and tests (Task 3), respectively. We approach the named entity identification task using state-of-the-art machine learning techniques, CRFs models, which allow the use of a considerable number of linguistic and semantic features derived from clinical text. For Task 2 and Task 3, we explore various types of context patterns, and in particular make extensive use of hierarchical structures of dependency parse trees to capture the complex syntactic relations underlying in the assertion/relation statements. The two examples given in the assertion annotation and relation extraction tasks demonstrate the usefulness of dependency parse trees in dealing with the long, complicated sentences that frequently occur in clinical text.

References

1. https://i2b2.org/NLP/Relations/
2. Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labelling sequence data. Proceedings of the International Conference on Machine Learning (ICML-2001); 2001: p. 282-9.
3. Klein D, Manning CD. Accurate unlexicalized parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL'03); 2003: p. 423–30.
4. Fundel K, Kuffner R, Zimmer R. RelEx - relation extraction using dependency parse trees. Bioinformatics. 2007; 23(3):365–71.
5. http://nlp.stanford.edu/software/lex-parser.shtml
6. http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/
7. http://crfpp.sourceforge.net/
8. Yang H, Keane J, Bergman C, Nenadic G. Assigning roles to protein mentions: the case of transcription factors. J Biomed Inform. 2009; 42(5):887-94.
9. http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html#Stopwords