Extraction of Medical Concepts, Assertions, and Relations from Discharge Summaries for the Fourth i2b2/VA Shared Task

Kirk Roberts, Bryan Rink, Sanda Harabagiu, PhD
The University of Texas at Dallas, Richardson, TX

Abstract

This paper describes our submission for the Fourth i2b2/VA Shared Task. We describe supervised classification-based approaches for all three challenges: medical concept extraction, classification of assertions for medical problems, and identification of medical relations between two concepts. We use a combination of conditional random field (CRF) and support vector machine (SVM) classifiers.

Introduction

Hospital discharge summaries are filled with important medical information that is expressed in natural language, such as diseases and treatments. The task of automatically extracting this information is complex. Moreover, natural language is ambiguous at lexical, syntactic, and semantic levels. Automatically understanding discharge summaries requires several forms of pragmatic reasoning that impact the quality of information extraction from such medical texts. Aiming to evaluate the quality of automatic processing of such documents, the Fourth i2b2/VA (Informatics for Integrating Biology & the Bedside) Shared Task considered three challenges:

1. Extraction of medical (a) *problems* (e.g., disease, injury), (b) *tests* (e.g., diagnostic procedure, laboratory test), and (c) *treatments* (e.g., drug, preventative procedure, medical device) from discharge summaries. Together these are referred to as medical *concepts*.
2. Classification of assertions (belief status) for extracted medical problems for a given patient (e.g., a problem is present, absent, possible, conditional on another factor).
3. Identification of relations between medical concepts (e.g., test reveals problem, problem indicates problem).

To address the natural language processing (NLP) problems required for the three challenges of the i2b2/VA Shared Task we have developed separate supervised methods for each challenge. In this way, we cast all three challenges as supervised classification tasks. The decision to use supervised methods was made for three reasons. First, supervised systems have been shown to be robust to slight changes in the data and easily portable to different tasks. Second, no member of our team has significant medical expertise. The process of creating a supervised system through iterative feature creation, testing, and error analysis is easily accessible to a non-expert. Finally, the organizers provided what is perhaps the largest repository of sample data for these challenges. We believe this data to be sufficiently large to serve as training data in our approach.

Previous techniques for extracting medical information from text have used both rule-based[1] and supervised[2] approaches. Rule-based approaches have traditionally performed better when less information is available for training. In recent years the increased emphasis on annotated data has shifted the balance toward supervised and semi-supervised approaches.

The rest of this paper is organized as follows. The next three sections discuss our approaches for concept extraction, assertion classification, and relation identification, respectively, and contain the results for the individual challenges. Finally, we conclude with a discussion on our work as a whole.

Concept Extraction

We cast the problem of extracting medical concepts (problems, tests, and treatments) as a sequence classification problem. Extracting concepts from discharge summaries involves two decisions: (1) what are the boundaries of concepts (i.e., what are the first and last words of each concept), and (2) what is the type (i.e., problem, test, or treatment) of the identified concept. First, a classifier decides whether a word belongs to a concept or not, whereas a second classifier assigns the concept to one of the three concept classes. While it is possible to perform concept boundary and type classification simultaneously, we have decided, based on our initial analysis, that separate classifiers should perform better. In that analysis, concepts of different types often shared a similar context, while the concept's type was more likely local to the words in the concept itself.

Discharge summaries have a structure in which prose (or narrative text) written by medical professionals is intertwined with lists or special fields. For example, common special fields include the patient's admission date, date of birth, sex, and a list of past

medical problems or treatments. These fields do not form grammatical sentences. Context surrounding a word, including list numbers, section names, and punctuation do provide clues to the existence of a medical concept. But these clues are sufficiently different from natural language prose that we decided to use two classifiers: one for prose sentences, the other for non-prose. Our prose detection heuristic works in the following way. If a sentence ends with a typical prose-marking punctuation token such as a period or question mark, it is considered prose. If it ends with a typical section header punctuation token such as a colon, it is considered non-prose. In addition, it is a non-prose sentence if it has five or fewer tokens or if at least half the tokens are judged to be non-prose tokens. A non-prose token is defined as one containing punctuation, a digit, or is all upper-case. All remaining sentences are considered prose. Using this heuristic, 48% of sentences in the training data and 50% of sentences in the test data were judged to be prose. Furthermore, 85% of gold concepts in the training data and 83% of gold concepts in the test data were contained in prose.

For each set of sentences (prose and non-prose), a conditional random field (CRF) classifier[4] was trained to detect concept boundaries. CRFs are sequence classifiers in that they maximize the likelihood of a sequence (of tokens, in our case) instead of a single instance. They are commonly used in other natural language sequence classification tasks such as part-of-speech tagging, phrase chunking, and named entity recognition. The two CRF classifiers used different sets of features, shown in Tables 1 and 2. As might be expected, since the non-prose classifier has less context, it uses a smaller and simpler feature set. Most of the features are self-explanatory, but some merit discussion.

One of the features that we used extracts Quantitative ANNotations (QANN) such as dates, measurements, dosages, etc. Whenever values for the QANN feature were extracted from text, two possible values were obtained: either the semantic class (e.g., "DOSAGE") if one is recognized or the lower-cased word. Another feature that we used was named Generic#, which replaces all digits with the value 0. This allows the strings "9.1" and "2.0" to receive the same feature value.

The part-of-speech context features concatenate the parts-of-speech of the previous and next *N* tokens, including the current token.

Features based on MetaMap[5], UMLS[6], and GENIA[7] were also used for detecting a concept's boundaries.

Uncased word
QANN uncased word
Uncased previous word
Previous word's part-of-speech
3-token part-of-speech context
MetaMap semantic type
MetaMap CUI

Table 1. Features used for detection of concept boundaries in non-prose text.

Generic# stemmed word
Previous word
Uncased previous word
Last 2 letters of word
Previous part-of-speech
1-token part-of-speech context
UMLS concept hypernyms
MetaMap semantic type
Word's GENIA stem
Word's GENIA entity type
Word's GENIA phrase chunk type
Previous word's GENIA part-of-speech
Previous word's GENIA stem
Previous word's GENIA phrase chunk type
Next word's GENIA stem

Table 2. Features used for detection of concept boundaries in prose text.

Uncased unigrams
First 4 letters of each word
Stemmed previous word
Stemmed next word
Uncased previous bigram
Argument type + nearest predicate
UMLS concept type
Wikipedia concept type

Table 3. Features used for classification of concept types.

After concept boundaries are identified, we use a support vector machine (SVM) classifier[8] to determine whether the concept is a problem, test, or treatment. Since the surrounding context is not as important for this stage, the same classifier handles concepts from both prose and non-prose sentences. Context was still quite important. However, this classifier was more successful when using both sets as input. Table 3 shows the features used by the SVM classifier for concept type classification.

	P	R	F1
Exact Boundary	83.7	80.8	82.2
Exact Boundary + Class	81.0	78.2	79.6
Inexact Boundary	92.7	89.5	91.1
Inexact Boundary + Class	89.3	89.2	89.2

Table 4. Results for concept extraction.

The results for the concept extraction challenge are shown in Table 4. Clearly, boundary detection is a more difficult task than type classification, as the latter stage suffered little more than 3% degradation. Furthermore, using the inexact boundaries yields a 50% error reduction, suggesting that approximately half of the concepts guessed by the system were partially correct, but were missing leading or trailing words, or accidentally merged with or split from the following concept.

Features based on UMLS and MetaMap provided little improvement to system performance. We experimented with many features that captured far more knowledge from these resources, but ultimately simple features such as the semantic type were the only ones to improve results.

Assertion Classification

Determining the assertion of a medical problem (present, absent, possible, hypothetical, conditional, or associated with someone else) is performed with a single SVM classifier[8]. The features we used for the SVM classifier are shown in Table 5.

Generic# unigram
Indexed stemmed previous words
Previous word
Section name
Predicates in sentence
Sentence contains Negex
General Inquirer previous 5 tokens' categories

Table 5. Features used for assertion classification.

Two resources of note that aid in assertion classification are Negex[9] and the General Inquirer[10]. Negex is a negation detector that provides information about the negated medical term and type of negation. However, the only Negex feature we used was a boolean feature that fires if at least one Negex span is present in the sentence. The General Inquirer is a content analysis resource that contains categorical data for many common English words. Notable categories for this challenge are POSITIV and NEGATIV, given to words with positive and negative sentiment, respectively. Our General Inquirer feature returns the categories for each of the five words before the concept in the sentence.

Another feature that requires further explanation is the indexed stemmed previous words feature. This feature concatenates the stem of each previous word in the sentence with its token distance from the concept. Consider the sentence "[Chest x-ray] revealed [mild pulmonary edema]." For the concept "mild pulmonary edema", this feature would return the values "3.chest", "2.x-ray", and "1.reveal".

The results for assertion classification are shown in Table 6. Due to the vastly uneven distribution of classes, many of the under-represented classes have poor results.

Similar to the problems incorporating medical information from UMLS and MetaMap into concept extraction, we were unable to get any significant gain from Negex features. Negex provides a wealth of semantic information, but in the end only a single, overly simplistic feature provided better results where more semantically-informed features failed. That feature indicates whether a Negex span was located anywhere in the sentence, not just within the concept being classified.

	#	P	R	F
Present	13025	93.3	97.8	95.5
Absent	3609	95.0	90.8	92.9
Possible	883	76.8	51.9	61.9
Hypothetical	717	86.8	80.1	83.3
Conditional	171	67.2	26.3	37.8
Assc. w. someone else	145	85.6	73.8	79.3
Overall	18550			92.7

Table 6. Results for assertion classification.

Relation Identification

The relation challenge required teams to identify relations between the concepts in patient records. A relation exists between two concepts if they participate in one of eight defined relation types listed in Table 7. The relations require at least one of the concepts to be a medical problem. Teams were allowed to make use of the gold concept and assertion information during relation identification. However, no information was given about which pairs of concepts might possibly have a relation between them, except that relations were limited to a single sentence. This meant a relation could be present between any two concepts in a sentence. Relation identification then consists of (i) determining which pairs of concepts have a relation, and (ii) identifying the type of that relation. Only a small number of the pairs of concepts in a sentence actually had a relation. After discarding concept pairs that had more than 9 intervening concepts

between them, only about 15% of concept pairs had a relation between them in the training data. Thus, detecting which pairs of concepts had a relation was the more difficult part of relation identification.

Type	Description
TrIP	Treatment improves medical problem
TrWP	Treatment worsens medical problem
TrCP	Treatment causes medical problem
TrAP	Treatment is administered for problem
TrNAP	Treatment is not administered because of problem
PIP	Problem indicates problem
TeRP	Test reveals medical problem
TeCP	Test conducted to investigate problem

Table 7. Relation types.

Our relation system consists of a single multi-class SVM classifier, namely LibSVM's LIBLINEAR[11]. We tuned three parameters using cross validation on the training set: the penalty parameter, C, the tolerance of termination, epsilon, and a weight assigned to the class used for relation absence (C=0.5, eps=0.5, weight=0.025). Training instances for the classifier are formed by pairing together all concepts in a sentence and extracting features related to each pair. We did not remove pairs without a problem from consideration as they seemed to aid the classifier. The target class presented to the classifier for each concept pair corresponds to the relation type, or NONE if there is no relation between the concepts. Once the classifier is trained it can be used on the test set by presenting the values for each pair of concepts from a test sentence to the classifier and assigning the appropriate relation identified.

The features presented to the classifier can be categorized based on the type of information they provide about a relation. The types of information used are: (1) the context consisting of words between the two concepts, (2) information related to only one of the concepts, (3) the types of nearby concepts, (4) information from Wikipedia, and (5) contextual similarity to training concept pairs. Assertions are used in only one feature: the concatenation of the assertion types for both concepts.

Several features are based on the words between the two concepts. Those words often directly indicate the presence and type of any relation between the two concepts. These features include: the word itself, part of speech, and the concept type if the word belongs to a concept. Another set of features consider the entire multi-word phrase found between the concepts: the phrase itself, whether the phrase is in a list of conjunction phrases, and the sequence of GENIA[7] phrase chunk types, where concept type replaces chunk type if any concepts occur between.

Since all concepts in a sentence are paired together, an individual pair may have intervening concepts between them. The final feature examining context between two concepts considers the types of any relations present between intervening concepts. This feature is computed based on the manual annotations during the training phase, and using system output during testing. Concept pairs are classified in order of increasing distance to ensure any shorter-distance relations are classified first. Those relations are then used as a feature by the classifier when classifying other relations in the sentence.

Features in the second category consider each concept in the pair in isolation. Information about one argument can provide positive or negative evidence of relation presence. For each concept we extract: WordNet lemmas, General Inquirer positive and negative polarity, unstemmed tokens, exact phrase, one token before, a set of the 3 tokens after the concept, and concept type. Additionally, we extract predicates associated with the concept based on an SRL parse by ASSERT[12].

Often, determining that two concepts do not have a relation between them requires information about the other relations in the sentence. To capture this, we have two features which look at nearby concepts. One feature concatenates the types of both the first concept and the preceding concept. The second feature works analogously, concatenating the concept type of the second concept and the following concept. These features help indicate whether a relation is likely to be present nearby.

We make use of two types of knowledge from Wikipedia. For each concept in the text we attempt to map it to Wikipedia through exact match to a page name, following redirects if necessary. The first type of knowledge is the existence of a hyperlink from one of the articles to the other. This is a proxy for concept relatedness. The second uses the Wikipedia category hierarchy and indicates the depth of the least common subsumer. This is additional evidence for concept relatedness, at a semantic class level.

The relation features mentioned up to this point require an exact match of their value against the training set to be useful. To overcome this we also developed a set of features which can perform inexact matching. All of these features make use of the edit distance measure in combination with K-nearest neighbors (KNN). Rather than using strings of characters as input to the edit distance we compose sequences of elements which take the place of

characters. The types of elements considered are: parts of speech between the concepts, concept types in the entire sentence, stems of the words between the concepts extended to 2 words on either side of the concept pair, the words and dependency links of the shortest dependency path, and finally the chunk and concept types sequence mentioned earlier. Within the sequences, the two concepts from the concept pair are replaced with their concept types and neighbors are filtered to guarantee a match on those types. Features are formed by considering the percentage of top neighbors that belong to each relation type. Those percentages are used as continuous features, one for each sequence type/relation type pair. During training, any neighbors from the same document are excluded from consideration. We chose K between 15 and 100 based on how similar we judged the top neighbors to be for each sequence type.

Several features that proved useful for the concept task did not help at the relation stage, including UMLS information. One explanation for this could be that knowledge of the general type such as problem, treatment, and test is enough for relation extraction. We hypothesize that using fine-grained concept type pairs would help, but the current data set is too small to reliably learn their associations with relation types. Exploratory queries of concept pairs on Google revealed that lexical patterns could be learned for the different relation types. This resource would aid in the cases where two concepts are prototypically in a relation together. However, Internet usage at test time was disallowed.

Table 8 shows our system score for identifying relation concept pairs and classifying relation type. Our F1 score for relation type classification drops less than 8 percent versus our relation detection F1 score, representing a relatively small loss caused by type classification. This confirms the claim that relation detection is more challenging.

Score Type	Recall	Precision	F1
Relation presence	82.3	78.7	80.5
Presence and type	75.3	72.0	73.7

Table 8. Relation scores.

Discussion

We were disappointed by the lack of improvements provided by resources specific to this domain such as MetaMap, UMLS, and Negex. While they provided enough information to include in some features, they did not provide nearly the improvement we had expected. We therefore leave the problem of how to best extract information from these resources into a supervised framework as future work.

Conclusion

We have described our approaches for all three challenges: medical concept extraction, assertion classification, and relation identification. For each challenge, we employed a supervised framework of one or more classifiers with features developed specifically for this task, but generalizable to non-medical domains.

References

1. Childs L, Enelow R, Simonsen L, Heintzelman N, Kowalski K, Taylor R. Description of a rule-based system for the i2b2 challenge in natural language processing for clinical data. Journal of the American Medical Informatics Assoc. 2009;16(4):571-5.
2. Patrick J, Li M. A cascade approach to extraction medication events. Australasian Language Technology Association Workshop. 2009.
3. Cohen AM, Hersh WR. A survey of current work in biomedical text mining. Briefings in Bioinformatics. 2006;6(1):57-71.
4. McCallum AK. Mallet: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.
5. Aronson AR. Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. http://mmtx.nlm.nih.gov. 2001.
6. Lindberg DA, Humphreys BL, McCray AT. The unified medical language system. Methods of Information in Medicine. 1993;32(4):281-91.
7. Tsuruoka Y, Tateishi Y, Kim JD, Ohta T, McNaught J, Ananiadou S, Tsujii J. Developing a robust part-of-speech tagger for biomedical text. Advances in Informatics – 10th Panhellenic Conference on Informatics, LNCS 3746. 2005; 382-392. http://bit.ly/geniatagger.
8. Joachims T. Making large-scale SVM learning practical. Advances in Kernel Methods. http://svmlight.joachims.org. 1999.
9. Chapman W, Chu D, Dowling JN. A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of Biomedical Informatics. 2001;34:301-10.
10. Stone PJ, Dunphy DC, Smith MS, Ogilvie DM. The general inquirer: a computer approach to content analysis. M.I.T. studies in comparative politics. 1966.
11. Fan RE, Chang KW, Hsieh CJ, Wang XR, and Lin CJ. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research. 2008: 9:1871-1874.
12. Pradhan SS, Ward W, Hacioglu K, Martin JH, Jurafsky D. Shallow Semantic Parsing using Support Vector Machines. HLT/NAACL. 2004.