Automated extraction of clinical concepts – an I2B2 experience

Jung-Wei Fan, PhD, Yang Huang, PhD, Rommel M. Yabut, MD
Daniel S. Zisook, MD, John E. Mattison, MD
Kaiser Permanente Southern California, Pasadena, CA

Abstract

Extracting and classifying concepts in clinical documents is essential in medical language processing and can benefit various applications ranging from administration to research. Participating in the 2010 i2b2/VA challenge of extracting problems, tests, and treatments from progress notes and discharge summaries, we used open-source programs to develop a system consisting of rule-based and machine learning components. The system achieved F-measure of 0.709 on class exact span and 0.825 on class inexact span. The experience of dealing with such rich clinical corpora was valuable and the error analysis provided useful insights for future improvements.

Introduction

Vast amount of useful information resides in unstructured clinical text and serves a critical role in administration, clinical practice, as well as research. Since computable clinical text became available, the medical language processing (MLP) community has been motivated to develop automated programs to extract meaningful concepts such as diagnoses, procedures, and medications.[1] Extraction and classification of the concepts form an essential step in MLP. Not only are the classified concepts themselves useful in applications such as coding/billing, but also the developed software modules can be building blocks of more advanced applications such as decision support that involves relation mining and integration with structured data.

To facilitate MLP research, the i2b2/VA Challenges in Natural Language Processing for Clinical Data (https://www.i2b2.org/NLP) have been organized with systematic data preparation and evaluation. The tasks are directly or indirectly associated with clinical applications and draw considerable attention and participation of the community. In 2008, thirty teams participated in the Obesity and Comorbidities extraction; in 2009, twenty teams participated in the Medication extraction challenge. The challenge of this year was to extract problems, tests, and treatments mentioned in de-identified progress notes and discharge summaries. Three incrementally related sub-challenges were organized: (1) extracting concepts of the three target classes, (2) assigning the assertion type of the extracted concepts, and (3) identifying the relations between the concepts. Each registered team was allowed to participate in any subset of the sub-challenges and access the manually annotated training corpus about three months before the final evaluation.

We participated only in the concept extraction sub-challenge. Our methods involved manually-crafted rules and machine learning models built from the training corpus (349 documents). The system was developed by customizing open-source software. On class exact span we achieved F-measure 0.709 (recall 0.649 precision 0.782). On class inexact span the F-measure was 0.825 (recall 0.768 precision 0.891).

Methods

Since the sentence and token boundaries had been detected and provided in the input text, our process pipeline (see Figure 1) started directly with part-of-speech (POS) tagging. Predefined POS patterns and machine-learning/rule-based chunkers were then applied to extract candidate phrases for looking up the UMLS[2] (2010AA) Metethesaurus strings. The UMLS semantic types and a machine-trained classifier were used to determine the class of the phrases. In the end, a rule-based post-processor split composite phrases (e.g., noun phrase with prepositional attachment) if its individual sub-phrases were classified as qualified target concepts. Each step of the pipeline is elaborated as follows:

1. POS tagging

For our potential need to be compatible with other programs of the computational linguistics community, we modified and re-trained the dTagger[3] program to make it output the Penn Treebank POS tags.[4] dTagger was chosen as it uses a comprehensive lexicon based on the UMLS and is able to tag multi-word phrases (e.g., “cord compression” is tagged as NN). A major step in the adaptation was UMLS-Treebank tag set mapping, which was based on Huang et al.’s earlier work of augmenting a statistical parser with the UMLS lexicon to identify clinical noun phrases.[5]

2. Phrase chunking

According to the annotation guideline, the target concepts have to be complete noun phrases or adjective phrases and are subject to specific syntactic constraints (e.g., structurally only up to one prepositional phrase can be attached). To meet the criteria, we manually created regular expressions to search longest-spanning sequences of POS tags for the syntactically qualified phrases. As we observed considerable false-positive substrings (e.g., the prepositional phrase in "a C5-6 ACDF by Dr. Lastname") extracted by our semantically insensitive POS patterns, we developed a two-step filtering to truncate substrings that should not be part of the target concepts: (1) Sub-phrases right before/after a preposition (or conjunction) were determined as desired or undesired by a machine learning classifier (SVM of the Weka toolkit[6]). The classifier was trained with the gold standard annotations, using unigrams and bigrams as features; (2) Regular expression rules were created to trim undesired beginning/ending words (e.g., negation determiners such as "no" that should not be part of the concepts).

3. Lexical lookup

The candidate phrases output by the phrase chunker were then used to look up into the UMLS, mainly to obtain the semantic types of the matched concepts that can be clearly mapped to one of the three target classes. We manually determined a mapping table between some semantic types (29 out of the 133) and the challenge's classes. For example, T034 Laboratory or Test Result and T060 Diagnostic Procedure were mapped to the test class. In constructing our lookup list from Metathesaurus, heuristic filtering was performed to exclude the non-English, NCI, OMIM, and strings with special characters such as "@" and ";". To increase recall of looking up the domain-specific strings, general English words such as determiners (e.g., "the") and possessive pronouns (e.g., "her") were not included in the matching. However, we did not perform normalization or error-correction for each word of the candidate phrase before the lookup.

4. Semantic classification

The lexical lookup resulted in three possible conditions and we handled them separately in the semantic classification. One common component in processing each condition was a machine learning classifier (also based on SVM of Weka) able to assign a candidate phrase one of the four classes: problem, test, treatment, or else. The classifier was trained with the gold standard annotations, using unigrams, head word, and ordered context unigrams as features. The else class instances were candidate phrases output by our chunker (step 2 of pipeline) that did not overlap with any concept annotations. The process for each condition is elaborated as follows: (1) Unique concept match – if the semantic type(s) corresponded to a single target class, the class was assigned; if not (e.g., T033 Finding was not included in the direct mapping table), the SVM classifier was used to determine the class (could be else); (2) Multiple concept matches – if the semantic types of all the matched concepts corresponded to a single target class, the class was assigned; if not, the SVM classifier was used to determine the class (disambiguation was literally performed for concepts associated with different target classes); (3) No concept match – the SVM classifier was directly used to determine the class.

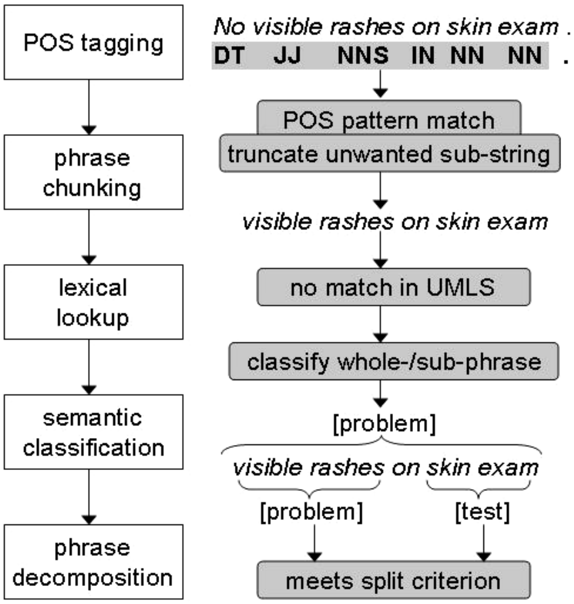

Figure 1. System pipeline (left side) and an example (right side).

5. Phrase decomposition

According to the annotation guideline, target concepts as parts of a composite phrase should be annotated separately. We specifically crafted some heuristics to decompose noun phrases with prepositional attachment where the two sub-phrases were both qualified target concepts. The sub-phrases before and after the preposition were classified by the SVM individually, and were annotated as separate concepts if each of them belonged to one of the three target classes. For example, in Figure 1 "visible rashes" and "skin exam" are classified as problem and test respectively, so the composite concept is split into two. Additionally, some rules based on the preposition and the concept classes were later created to refine the split conditions. For example, if both sub-phrases before and after "of" were classified as treatment concepts (e.g., "an IV load of Decadron"), most likely it should remain a single concept.

Results

The evaluation set consisted of 477 documents, and an official evaluation script was released for computing recall, precision, and F-measure based on exact or inexact matching. Table 1 shows the performance of our system based on the class exact span and class inexact span. The difference of overall recall (0.119) between exact and inexact match indicates that our system got a lot of phrases partially overlapping with the gold standard boundaries. For exact match, we missed more tests but the precision was higher; on the other hand, the system was more sensitive in extracting treatments but less precise. The performance on extracting problems lied in between, and the F-measures of the three classes were close. For inexact match, the system appeared to do relatively well on the problem class.

Class exact span			
	Recall	Precision	F-measure
Problem	0.641	0.790	0.708
Test	0.634	0.806	0.710
Treatment	0.673	0.751	0.710
Overall	0.649	0.782	0.709
Class inexact span			
	Recall	Precision	F-measure
Problem	0.738	0.910	0.815
Test	0.715	0.909	0.801
Treatment	0.763	0.852	0.805
Overall	0.768	0.891	0.825

Table 1. System performance based on exact/inexact span, considering correct class assignment.

Errors were observed in every stage of the pipeline. Below we provide some stage-wise examples:

1. POS tagging

In a sentence "He will start prednisone…" the "start" was mis-tagged as NN (singular noun), resulting in an inexact noun phrase "start prednisone" with the POS pattern "NN NN" matched by our chunker.

2. Phrase chunking

Due to the limited coverage of our manually crafted POS patterns, considerable phrases were dropped. For example, "Wt gain of 25 + pounds" was not captured because the "CD SYM NNS" POS sequence of the prepositional phrase was not covered.

Inexact chunking also occurred when invisible formatting boundaries were not detected by our pure POS-based patterns. For example, a sequence of lab results like "WBC – 20.0* RBC – 3.96*…" where the boundaries can be inferred to lie between the pairs of test – value. However, our chunker took "20.0* WBC" as one phrase.

The phrase truncation did not always work correctly, either. For example, in a sentence "…decrease his IV fluids in an effort to…" the chunker marked "his IV fluids in an effort" as the candidate phrase, apparently failing to trim off the undesired prepositional phrase.

3. Lexical lookup

Our pre-cleaning before lookup was not perfect. Although we considered disregarding the sub-string "patient's" when looking up the UMLS, we missed its abbreviated form "pt's" and therefore failed in some matches.

The UMLS (2010AA) did not cover some of the clinical abbreviations (e.g., "UreaN", "Creat", "chemotx", "Vit.C", etc), and they would be missed if also misclassified by the SVM.

4. Semantic classification

Misclassifications have been observed. For example, "elevated one hour glucose tolerance test" was classified as test; "respiratory rates" was classified as else; "bicarbonate" was classified as treatment; and "sister-in-law" was classified as problem.

5. Phrase decomposition

There were cases where a single concept should not have been split, but split by our decomposition rules. For example, "MRI of the lower extremity" was split into "MRI" and "the lower extremity" because the latter was misclassified by the SVM as problem, making the two sub-phrases qualify for the splitting pattern: test of problem. This example also shows errors in one component could result from that in another.

Discussion

This year's challenge offered valuable resource and experience for the MLP community, both in terms of the diversity of the tasks and the size of the corpora.

For example, co-reference resolution appeared to be embedded in the concept extraction challenge: in a sentence “Similac 22 calories per ounce… decreasing to 20 calories per ounce” the “20 calories per ounce” was annotated as treatment in gold standard (0451.con). It can be projected that these resources will remain influential over years as benchmark for training and evaluating the tasks. With continuous refinement (e.g., by community effort like the AMIA NLP Working Group), it is expected that the accuracy and value of the annotations will also increase gradually.

To tackle the concept extraction challenge, we took a conventional syntactic-then-semantic approach in building our system pipeline. Error analysis indicates that each of our components still has much room to improve. For example, re-train the POS tagger with more annotated clinical texts and implement more flexible matching for the lexical lookup. In hindsight, handling syntax first probably cost us more effort to get the phrase boundaries right and still could not avoid considerable false positives and false negatives. On the contrary, directly applying greedy string match with a comprehensive semantic lexicon and followed by tweaking for the syntactic requirements might have yielded better performance. However, the conjecture needs to be verified yet.

Since release of the training corpora, errors have been reported in the gold standard annotations. It is not difficult to find errors in the final test annotations, either. For example, determiners are sometimes dropped by human annotators: in 0014.con “uric acid crystals” was annotated without the preceding “some”. Inconsistencies can be found even in the same document. For example, in 0341.con “status post uterine artery embolization” is annotated as treatment in line 49, while it is annotated in line 13 as “uterine artery embolization”, missing the “status post”. It is not clear the proportion, but apparently they contribute to inexact matches in the evaluation.

Conclusion

By using open source software, we developed a system for the 2010 i2b2/VA challenge of extracting problem, test, and treatment concepts from progress notes and discharge summaries. The system achieved F-measure 0.709 (recall 0.649 precision 0.782) on class exact span and 0.825 (recall 0.768 precision 0.891) on class inexact span. Useful information was gained from inspecting the corpora and from analyzing the results.

Acknowledgements

De-identified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY.

References

1. Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inform. 2008:128-44.
2. Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med. 1993;32(4):281-91.
3. Divita G, Browne AC, Loane R. dTagger: a POS tagger. AMIA Annu Symp Proc. 2006:200-3.
4. Part-of-speech tagging manual of the Penn Treebank Project ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz (accessed 04/18/10)
5. Huang Y, Lowe HJ, Klein D, Cucina RJ. Improved identification of noun phrases in clinical radiology reports using a high-performance statistical natural language parser augmented with the UMLS specialist lexicon. J Am Med Inform Assoc. 2005;12(3):275-85.
6. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The Weka data mining software: an update. SIGKDD Explorations. 2009;11(1).