Erasmus MC Approaches to the i2b2 Challenge

Ning Kang,[*] Rogier J. Barendse,[*] Zubair Afzal, Bharat Singh, Martijn J. Schuemie, Erik M. van Mulligen, Jan A. Kors

Erasmus University Medical Center, Rotterdam, The Netherlands

Abstract

Erasmus MC participated in the concept annotation and assertion classification subtasks of the fourth i2b2 challenge. For concept annotation, our approach consisted of tagging the clinical data with a variety of named entity recognizers and combining the resulting annotations into a final annotation set by means of a simple voting scheme. For assertion classification, we manually crafted decision rules using tools from a text engineering package. Our results show that a combined annotation system performs significantly better than any of the individual systems. The assertion classification system shows a good overall performance, but performs less well for "possible" and "conditional" assertions.

Introduction

The biosemantics group at the Department of Medical Informatics of the Erasmus University Medical Center (Erasmus MC) participated in two of the three subtasks of the i2b2 challenge: (1) extraction of medical problems, tests, and treatments from clinical data (annotation task), and (2) classification of assertions made on medical problems in clinical data (assertion task). This paper briefly describes our methods and main results for these tasks.

Methods

Annotation task

Our approach for the extraction of concepts (problems, tests, treatments) from the clinical data consisted of tagging the data with a variety of named entity recognizers and chunkers, and combining the resulting annotations of the systems into a final annotation set by a simple voting scheme.

Concept annotation systems

We selected six annotation systems. One was a locally developed concept recognition and normalization tool, the other five were publicly available named entity recognizers and chunkers, which could be downloaded directly from their official websites. All of them are briefly described below.

1. ABNER (A Biomedical Named Entity Recognizer) (http://pages.cs.wisc.edu/~bsettles/abner/) is a software tool for text analysis in molecular biology.[(1)] The core of the system is a statistical machine learning system using linear-chain conditional random fields (CRFs)[(2)] with a variety of orthographic and contextual features. We used version 1.5, released in 2005.

2. Lingpipe (http://alias-i.com/lingpipe) is a suite of Java libraries for natural language processing, including part-of-speech (POS) tagging, named entity recognition, spelling correction, etc. The Lingpipe chunker supports rule-based, dictionary-based, and statistical chunking. We used the statistical chunker, which is based on a hidden markov model (HMM) [(3)] and according to the Lingpipe website, is the most accurate one. The version we used is 3.8, released in 2009.

3+4. OpenNLP Chunker and OpenNLPNer (http://opennlp.sourceforge.net) are made available by OpenNLP, an organizational center for open source projects related to natural language processing. An OpenNLP UIMA wrapper has been developed by JULIE Lab (http://www.julielab.de). The wrapper divides the OpenNLP package into small modules that perform sentence detection, tokenization, POS tagging, chunking (OpenNLP Chunker), named entity recognition (OpenNLPNer), etc, which makes it easy to configure the pipeline for different purposes. We used both the chunker and the named entity recognizer since they use different training models:

[*] Both authors contributed equally to this work.

OpenNLP Chunker is based on a maximum entropy model (MEM), and OpenNLPNer is based on a CRF model. The version we used is 2.1, released in 2008.

5. Peregrine is a thesaurus-based concept recognition tool, developed by the biosemantics group at Erasmus MC (http://biosemantics.org). Peregrine is extremely fast and includes a number of disambiguation rules to improve performance.[4] For the i2b2 challenge, we used the Unified Medical Language System (UMLS)[5] 2009 thesaurus filtered for relevant semantic types, in combination with post-processing rules to improve precision.

6. StanfordNer (http://nlp.stanford.edu/software/CRF-NER.shtml) is a named entity recognizer developed by the Stanford Natural Language Processing Group. It is based on a linear chain CRF model. The version we used is 1.1, released in 2009.

Concept annotation steps

The following processing steps were done to generate the concept annotations:

1. All tools except Peregrine were trained on the i2b2 training corpus (349 clinical records with concept annotations). For the chunkers, the corpus was converted to an appropriate (IOB) input format.

2. All six tools were integrated into the Unstructured Information Management Architecture (UIMA) framework.[6]

3. The UIMA Collection Reader read the 477 clinical records in the i2b2 test corpus , and called each of the six tools to annotate the records.

4. Using majority voting theory,[7] the annotation results of the systems were used to generate a combined annotation result. If a concept annotation provided by any three of the six systems was the same (i.e., the same start and end position, with the same concept type), then this annotation became the combined annotation. The threshold of three was selected because it gave optimal performance on the training data.

5. The combined annotations were output according to the i2b2 annotation file format.

Assertion task

For the assertion task we used the GATE (General Architecture for Text Engineering) 5.1 (http://gate.ac.uk) package,[8] developed by the University of Sheffield. In particular, we manually crafted a number of decision rules using the JAPE language provided by GATE.[9]

For each of the assertion categories (“absent”, “present”, “conditional”, “hypothetical”, “associated with someone else”, and “possible”) we manually searched the sentences with that assertion for common combinations of terms. These terms were assumed to be indicative for a category and were used in rules, in combination with the word-distance between the term and the concept about which an assertion was made. In the following, we briefly describe for each assertion category the indicative words that are used in our rules. For each concept, the rules were tested in the order below. A concept was always assigned to the first category that applied.

Associated with someone else

We identified indicative terms like ‘mother’, ‘sister’, ‘caregiver’, etc. The rule for this category would not be executed if our system classified a record as neonatal. In these types of records concepts could relate to mother or child, who are both the subject of the report and therefore are marked ‘present’ in the gold standard.

Hypothetical

We identified indicative terms like ‘as needed’, ‘prn’, ‘if’, ‘call’, etc.

Conditional

We looked for concepts followed by indicative terms like ‘exertion’, ‘with medication’, ‘with allergies’, etc.

Possible

We identified the indicative terms like ‘suspect’, ‘likely’, ‘possible’, etc.

Absent

We identified words like ‘no’, ’not’, ‘without’, etc. If negation term was followed by words like ‘but’, ‘except’, ‘with’, the part of the sentence preceding these words was filtered out. We also generated a list of concepts which by themselves were indicative for this category, such as ‘afebrile’, ‘non-tender’, or ‘atraumatic’.

Present

All concepts that could not be classified as one of the assertions above were classified as present.

Results

Concept annotation

The performance of the six annotation tools and the combined annotation are shown in Table 1.

Table 1. Performance of six annotation systems and the combined annotation on the i2b2 testing corpus.

Software	Recall	Precision	F-score
ABNER	0.693	0.796	0.741
Lingpipe	0.740	0.731	0.735
OpenNLP Chunker	0.633	0.784	0.700
OpenNLPNer	0.765	0.789	0.776
Peregrine	0.400	0.565	0.468
StanfordNer	0.723	0.820	0.768
Combination	0.811	0.831	0.821

Recall, precision, and F-score of the combined system are all higher than those of any individual system. The three trainable named entity recognizers (ABNER, OpenNLPNer, and StanfordNer) perform best, Peregrine performs worst. However, when the annotations of Peregrine (or any other system) were not used in the combined annotation, the F-score for the combined annotation decreased. When we increased the threshold for agreement from 3 to 6, precision increased and recall dropped (Figure 1). F-score was highest for a threshold of 3.

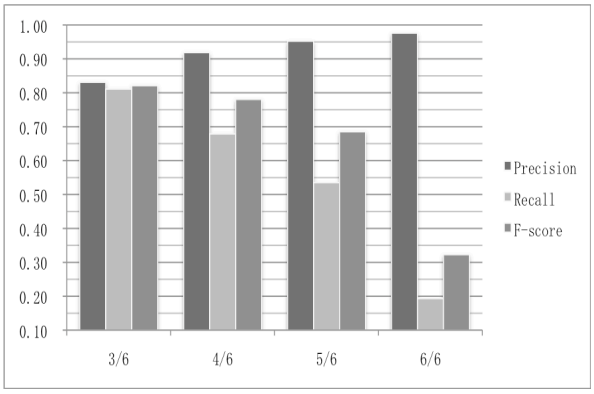

Figure 1. Performance of the combined annotation on the i2b2 test corpus for varying thresholds of agreement between the six annotation systems.

Table 2 shows the detailed performance results of the combined annotation on the i2b2 test corpus for exact matching concpets and inexact matching concepts (i.e., spans have at least one matching boundary), with and without matching concept class. The output was generated with the i2b2 evaluation software (note that the counts in row "Class Inexact Span" do not match the performance figures).

Assertion classification

Table 3 gives the detailed performance results for the assertion classification task. Due to a parsing error in two documents, 35 concepts failed to be classified by our system. While overall performance figures appear to be quite high, our decision rules for "conditional" and, to a lesser extent, "possible" assertions are performing less well. Results on the test set were similar to those on the training set (data not shown).

Table 2. Performance of the combined annotation system on the i2b2 test corpus.

	True Positive	False Negative	False Positive	Recall	Precision	F-score
TESTING 1.1 - Exact span for all concepts together						
Concept Exact Span	37554	7455	6334	0.834	0.856	0.845
Class Exact Span	36492	8517	7396	0.811	0.831	0.821
TESTING 1.2 - Exact span for separate concept classes						
Exact Span for Problem	15560	2990	2991	0.839	0.839	0.839
Exact Span for Treatment	11192	2368	1860	0.825	0.857	0.841
Exact Span for Test	10802	2097	1483	0.837	0.879	0.858
Exact Span With Matching Class for Problem	15422	3128	3592	0.831	0.811	0.821
Exact Span With Matching Class for Treatment	10741	2819	2150	0.792	0.833	0.812
Exact Span With Matching Class for Test	10329	2570	1654	0.801	0.862	0.830
TESTING 1.3 - Inexact span for all concepts together						
Concept Inexact Span	40957	4052	2931	0.910	0.933	0.921
Class Inexact Span	40957	4052	2931	0.907	0.901	0.904
TESTING 1.4 - Inexact span for separate concept classes						
Inexact Span for Problem	17086	1464	1370	0.921	0.926	0.923
Inexact Span for Treatment	12241	1319	865	0.903	0.934	0.918
Inexact Span for Test	11630	1269	696	0.902	0.944	0.922
Inexact Span With Matching Class for Problem	16868	1682	2146	0.909	0.887	0.898
Inexact Span With Matching Class for Treatment	11673	1887	1218	0.861	0.906	0.883
Inexact Span With Matching Class for Test	11017	1882	966	0.854	0.919	0.886

Table 3. Performance of the assertion classification system on the i2b2 test corpus.

	True Positive	False Negative	False Positive	Recall	Precision	F-score
TESTING 2.1 - Exact span for all assertions						
Exact Span With Matching Concept	18515	35	0	0.998	1.0	0.999
Exact Span With Matching Assertion	16630	1920	1885	0.896	0.898	0.897
TESTING 2.2 – Exact Span and Matching Assertion						
Present	12332	693	1028	0.947	0.923	0.935
Absent	3189	420	249	0.884	0.928	0.905
Possible	411	472	240	0.465	0.631	0.536
Hypothetical	539	178	224	0.752	0.706	0.728
Conditional	27	144	25	0.158	0.519	0.242
Associated With Someone Else	132	13	119	0.910	0.526	0.667

Discussion

Our results for the concept annotation task indicate that the combined annotation of a variety of annotation systems yields an F-score that is 4.5% higher than the best single system, OpenNLPNer. The trainable concept recognizers performed better than the chunkers and Peregrine, as might be expected, but all systems contribute to the performance of the combined annotation since removal of any system results in an F-score decrease,

The low performance of Peregrine may partly be explained by the use of UMLS, which is not geared towards terms in clinical records. Also, Peregrine performs concept normalization, which is known to be a more difficult task than concept recognition. A better filtering of UMLS terms, possibly expanded with the reference annotations in the i2b2 training corpus, would likely improve Peregrine's performance.

Our approach offers the possibility to vary precision and recall of the combined annotation by varying the voting threshold for agreement (cf. Figrue 1). For example, a threshold of 5 would give a high precision (0.95) with a reasonable recall (0.53); with a threshold of 6 an even higher precision (0.98) would be possible, but at the expense of a poor recall (0.20).

Regarding the assertion task, our system shows good overall performance but did not perform well for the 'conditional' and 'possible' categories. A possible explanation is that the numbers of these categories in the training set were too low to cover the full breadth of occurrence, which made it difficult to build decision rules with good generalization properties.

Because of time constraints, we did not yet explore the use of statistical methods to identify indicative terms automatically, as a starting point for our manual procedures. Also the added value of a (partly) manual approach over a fully automatic approach has to be further investigated.

Conclusion

A system that combines the annotations of clinical records by different annotation systems, performs substantially better than any of the individual systems. The combination approach is straightforward and allows the balancing of precision versus recall.

The advantages and limitations of a manual approach to building an assertion classifier have yet to be demonstrated.

Acknowledgements

This study was supported by the European Commission FP7 Program (FP7/2007-2013) under grant no. 231727 (the CALBC Project).

References

1. Settles B. Biomedical named entity recognition using conditional random fields and rich feature sets. Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA) 2004; Geneva, Switzerland. p. 104-7.
2. Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceeding of 18th International Conference on Machine Learning 2001; San Francisco, USA. p. 282-9.
3. Buyko E, Wermter J, Poprat M, Hahn U. Automatically adapting an NLP core engine to the biology domain. Proceedings of the Joint BioLINKBio-Ontologies Meeting 2006; Fortaleza, Brasil; 2006. p. 65-8.
4. Schuemie MJ, Jelier R, Kors JA. Peregrine: Lightweight gene name normalization by dictionary lookup. Proceedings of the BioCreAtIvE II Workshop 2007; Madrid, Spain. p. 131-3.
5. Bodenreider O. The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research. 2004;32(Database Issue):D267.
6. Ferrucci D, Lally A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering. 2004;10:327-48.
7. Penrose LS. The elementary statistics of majority voting. Journal of the Royal Statistical Society. 1946;109(1):53-7.
8. Cunningham H, Maynard D, Bontcheva K, Tablan V. GATE: A framework and graphical development environment for robust NLP tools and applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.
9. Cunningham H, Maynard D, Tablan V. JAPE: a Java Annotation Patterns Engine (second edition). Technical report CS-00-10, University of Sheffield, Department of Computer Science, 2000.