I2B2 2010 Challenge: Machine Learning for Information Extraction from Patient Records

Peter Anick, PhD[1], Pengyu Hong, PhD[1], NianwenXue, PhD[1], David Anick, MD[2]
[1]Brandeis University, Waltham, MA; [2]Marino Center, Cambridge, MA

Abstract

We applied machine learning techniques to three tasks in the I2B2 2010 Challenge in Natural Language Processing for Clinical Data: concept detection, problem assertion classification and relation detection/classification. Using a rich feature set of primarily shallow lexical and syntactic features, we achieved good overall classification results, observing that effectiveness of features varied considerably across individual classes.

Introduction

Electronic medical record (EMR) systems are now being widely implemented to help manage patient records, increase the ability of analysts to assess quality of healthcare, and lessen patient sufferance due to medical errors [1]. Decision support tools are essential components of EMR systems. Such tools may allow doctors to interact with EMRs to develop patient-specific decisions. While textual description in natural languages is one of the main modalities in EMR data, tools are yet to be developed to automatically, robustly, and accurately extract useful information from patient records. The I2B2 2010 Challenge provided an opportunity to evaluate approaches for information extraction from such texts. [1] The challenge consisted of three separate tasks:

(1) Concept detection: identifying the spans of concepts within medical records and labeling them as problems, treatments or tests.

(2) Assertion classification: further classifying problems into the assertions: present, absent, hypothetical, possible, conditional and "associated with someone else".

(3) Relation identification and classification: for each combination of concepts occurring within the same sentence, determine (a) whether the concepts are related and, if so, (b) which of the following relation types they represent: treatment improves problem (TrIP), worsens problem (TrWP), causes problem (TrCP), is administered for problem (TrAP) or is not administered for problem (TrNAP); problem indicates problem (PIP); and test reveals (TeRP) or is conducted to investigate (TeCP) problem.

A set of training records annotated by experts was provided for participants to develop their own tools within a given timeframe. We treated the three tasks as classification tasks and applied statistical machine learning techniques to them. For each, we designed a set of features which can for the most part be extracted from free texts. In the first task, classifiers based on Conditional Random Fields [2] were trained using the extracted features to deal with concept span detection and type identification. For the second and third tasks, support vector machines [3] were trained to learn problem assertion types and the relationships between concepts. In the rest of this paper we will explain in detail our feature design, classifier training processes, and validation results for our approaches.

Overall System Design

Training data consisted of 349 anonymized clinical patient records, comprising 27,837 concept instances, of which 12,152 concepts were classified as problems and 5,264 concept pairs as related pairs.

Although the patient records came from different sources, they shared common characteristics. Short lines that terminated in a colon were usually meant to serve as field headings for the text that followed. (e.g., "Family History:") Unstructured text within a field was often composed of standard, albeit syntactically complex English sentences such as

She had a workup by her neurologist and an MRI revealed a C5-6 disc herniation with cord compression and a T2 signal change at that level.

[1] The 2010 i2b2/VA challenge and the workshop are funded in part by the grant number U54-LM008748 on Informatics for Integrating Biology to the Bedside from National Library of Medicine. This challenge and workshop are also supported by resources and facilities of the VA Salt Lake City Health Care System with funding support from the Consortium for Healthcare Informatics Research (CHIR), VA HSR HIR 08-374 and the VA Informatics and Computing Infrastructure (VINCI), VA HSR HIR 08-204. MedQuist, the largest transcription technology and services vendor, generously co-sponsored the 2010 i2b2/VA challenge meeting at AMIA.

Also common were text lines consisting of elliptical medical sublanguage phrases and lists such as drug prescriptions or symptoms, as in

Significant for hypertension , hyperlipidemia .

During text preprocessing, lines suspected of serving as field names were identified and each subsequent line within the scope of a field was passed to the Stanford parser [4] to generate both part of speech tags and a dependency graph. Tokens were lemmatized into dictionary root forms based on their part of speech. Numbers and dates were each replaced by a meta-token. To facilitate feature generation, each line was stored within a "chart" data structure in which each column stored attributes related to the token at that position in the sentence. For the assertion and relation tasks, concept information, such as phrase start, end, and type was associated with corresponding chart columns.

To deal with the prevalence of coordinated concepts, chart functions which identified the start and end columns of a list of concepts of the same type separated by commas or the conjunctions {*and, or}* were used to generate *concept regions*. An n-gram generator utilized this information to generate n-grams within windows beginning at the start or end of a concept *region*, so that, for example, the first, second and third concepts in a list of coordinated concepts would all share the same n-grams to their right and left, essentially ignoring any intervening concepts within the shared region.

Concept span detection and classification

We framed concept detection and classification as a sequential tagging problem, and attempted to solve it with a Conditional Random Fields (CRF) [2] classifier.[2] Each token in a sentence is assigned a standard BIO tag [6], with B indicating the beginning of a concept, I meaning inside a concept, and O outside of a concept. The concept class is then attached to the BIO tag for the token. The following is an example of this representation:

This/O morning/O while/O landscaping/O the/O patient/O had/O unremitting/B-Problem 12-05/I-Problem pain/I-Problem with/O shortness/B-Problem of/I-Problem breath/I-Problem and/O diaphoresis/B-Problem ./O

When devising and selecting features, one concern was that many of the concepts in the training data might not be found in the test data, leading to a data sparseness problem. We addressed this issue by using the semantic network in the UMLS® Thesaurus [7] to bridge the potential gap between the training and test set vocabularies. By mapping concepts to their classes, which are relatively small in number, we hoped this problem would be diminshed.

We used two types of features in concept detection: features that are derived from the context of each token and features derived from the UMLS® Thesaurus. Features derived from the immediate context included:

- Current token
- Lemma of current token
- POS tag of current token
- Next verb
- Lemma of next token
- A conjunction of the lemmas of the current and next tokens
- Lemma of the previous token
- A conjunction of the lemmas of the current and previous tokens.
- Whether the current token is longer than 12 letters

In addition to the contextual features, we also used semantic features derived from UMLS® semantic types. We took a subset of the concepts in UMLS (concepts originated from English), and mapped them to their semantic types. The lemma of each token in the text was checked against the concept in this mapping. If the lemma of a token matched the first word in a concept name, then the feature 'Start_Unigram' feature is invoked. If it matched the last word in a concept name, then the feature 'End_Unigram' was invoked. If the current and next token matched the first two words in a concept name, then the feature 'Start_Bigram' was invoked. Conversely if the current and previous tokens matched the last two words in a concept name, then the feature 'End_Bigram' was used. Finally, if the current token matched a single-word concept, the feature 'Singleton' was invoked. The semantic type of the concept was also appended to this feature. So an actual semantic feature might look like

- End_Unigram=Therapeutic_or_Preventive_procedure

Assertion classification

For the assertion task, we began by training a classifier (linear kernel SVM[3]) using as features the 1,

[2] We used the Mallet toolkit [5].

[3] We used the SVM and chi-square feature ranking implementations available in the weka toolkit [8].

2, and 3-grams terminating within a distance of at most 4 terms from the start or end of the concept (or concept *region* if the concept was part of a coordinate construction). To better mimic modifier scoping observed in problem lists, we limited coordinate constructions to those concepts connected by *and* or *or*, not those connected by commas. We used chi-square to rank features and tested models using different numbers of features (3000, 5000, and 10,000), eventually deciding on a cut-off of 10,000 features. Starting from the baseline feature set, we manually inspected prediction errors generated by successive models run over the training data. Our inspection tool displayed for each error:

- The concept
- Predicted and actual class
- The line of text containing the concept
- The field name (i.e., most recent line terminating in a colon)
- Features of this instance, including features that were not actually used during training because of pre-filtering of the feature set

This feedback allowed us to hone our feature set and to determine whether diagnostic textual features were actually available in some form to the learner. The high feature cut-off supported the retention of useful but rare lexical features.

However, the great variability in linguistic expressions relative to the number of training instances for some classes led to circumstances in which good lexical clues did not occur often enough to be deemed statistically significant by the learning methods. To compensate, we developed lists of terms (primarily through manual analysis of errors) to serve as de facto synonym sets for useful semantic classes. For example, words such as *family, fam, sister, aunt, sibling, someone, person, individual* made a useful class of "words not likely to be references to the patient". A class of negative terms included not only *no, not, non*, but also *deny, unlikely, resolve, free*, and *immune*.

Reviewing problems annotated as "hypothetical", we noted that they often occurred within instructions given to the patient on discharge. To help capture these, a class of terms indicative of instructions was constructed, including *if, fear, detect, avoid, call, report.* Since many hypotheticals could not be determined from the contents of the sentence alone, we also created features that coded both the full field name and individual terms from the field name (such as *discharge, instructions*). Field names, such as *family history,* were also useful in recognizing problems classified as "associated with someone else".

Determining the scope over which lexical clues ranged posed difficulties given the complexity and variability of sentences. Many sentences referenced multiple problems, with some modifiers extending across conjuncts and others not. As noted earlier, we decided to limit concept regions for the assertion task to sets of adjacent problems conjoined by explicit *and* or *or.* Most n-gram matching was constrained to a window around each problem that did not cross a comma boundary. We did, however, allow for some features, such as words suggesting family members or hypotheticals, to be captured regardless of their positions in the sentence.

Recognizing that the absence/presence distinction sometimes required an understanding of the meaning of the concept itself[4], we first tried adding a feature for those prefixes (such as "non" or "a") that often serve as morphological negatives. However, this generated too many false alarms and we later compiled an explicit list of concepts known to indicate the absence of a problem. The eventual feature set comprised the following classes of features:

Ngrams

- 1, 2, 3-grams containing lemmas within 4 terms of start column and not crossing a comma
- ngram containing one of a list of terms indicating "possible"
- if concept is within 4 terms of the end of the sentence, ngram consisting of all lemmas to the right, including commas

Fields

- Field name as a whole
- Individual terms within a field name

Syntactic relations

- Syntactic subject of the sentence (based on dependency graph) if it matches a non-patient term such as a family member.
- Nearest verb to left and right
- Nearest noun to the left
- Nearest adjective to the left

Semantic categories (term lists)

- Single lemma within a window of 10 terms (based on concept region boundaries) not crossing a comma, matching one of the semantic categories: family, negative, possible, probable.

[4] For example, "afebrile", "non-dilated", "nabs", "unlabored" all indicate absence of a problem.

- Single lemma anywhere in the sentence matching one of the semantic categories: hypothetical, family, negative.
- Sentence containing lemmas matching both family and hypothetical classes (used to help learn precedence relationship between assertion classes)
- Lemma indicating *improve* or *worsen* to immediate left or right

Within–concept features

- First term of concept matches a definite determiner
- Term known to imply absence of a condition
- Term matching category*: family, possible, probable*

Relation identification and classification

We divided the relation classification task into two subtasks. Given the set of all concept pairs of type problem-problem, test-problem, and treatment-problem, the first subtask was to separate related pairs from those whose co-occurrence in the same sentence did not reflect a true relationship. The second subtask classified those concept pairs previously identified as related into one of eight relationship types (as described in the introduction.).

The majority of features were shared by both subtasks. For features referring to both concepts, the types of the two concepts and their order were incorporated into the name of the feature. This was necessary since, for example, the cues for relating a treatment to a subsequent problem may be different from those relating the same problem to a treatment occurring later in the sentence. For each concept pair, we demarcated three separate regions within the sentence – the text located between the concept occurrences and the text to the left and right of their respective *leftmost and rightmost coordinate concepts*. For concepts of type treatment or test, their coordinate concepts were (transitively) adjacent concepts of the same type separated by either conjunctions or commas. When determining problem-problem relations, however, no coordination was applied. Ngrams were captured within each region.

Subtask 1: Related pair identification

While explicit textual clues relating concepts exist in many cases (e.g., "test A reveals problem B"), there are also many cases in which the relationship is completely implicit, relying on a doctor's knowledge to make the connection. This is particularly true for pairs of problems; many text lines contain long lists of patient symptoms with no indication how or if the symptoms are related. To a limited extent, lexical clues for relatedness can be extracted from the concepts themselves through shared terms and affixes. Medical affixes (such as *ventr, derm, cephal*) typically refer to body parts or types of conditions, and it is a reasonable assumption that tests, treatments and problems that refer to related medical or anatomical concepts via affixes or shared terms are more likely to be related in the patient at hand.

The full feature set for related pair identification follows (where c1 = concept 1 and c2 = concept 2). All features referring to both concepts in the relation contain a prefix indicating the order of the concepts in the sentence (c1c2 or c2c1) and their respective concept types. For features referring to a single concept, its type is included in the prefix.

c1 type and c2 type (as single feature)
c1 and c2 share an affix
c1 and c2 share affix *a*
c1 and c2 share a term (of length > 4 characters)
c1 and c2 share term *t*
noun occurs between c1 and c2 regions
token distance between concept regions c1 and c2 (binned)
1,2,3-grams between c1 and c2 regions if no nouns appear between them
1,2,3-grams between c1 and c2 regions regardless of intervening nouns
1,2,3-grams to left and right of left and right concept regions, within window of 4 tokens
c1/c2 grammatical subject is a non-patient term
c1/c2 and its adjacent preposition to left/right
c1 and c2 are conjuncts
c1/c2 assertion class

Subtask 2: Relation classification

To minimize classification bias due to the uneven distribution of relation instances within the training data, we resampled the training set with replacement to generate a relatively balanced training set that is about three times the size of the original data set.

Error analysis again showed high variability in diagnostic features, many of which were relatively infrequent. In an attempt to generalize some of the more diagnostic ngram features, we created short lists of semantically related words for the categories *family, negative, possible, probable, worsen, improve, start, stop, test, reveal, rule-out.*[5] We defined a "semgram" as an ngram in which a term matching a member of one of these lists is replaced by a symbol indicating its semantic class

[5] E.g., the list for the category "improve" contains the words *improve,ameriolate,help,correct.*

The feature set follows:

c1 type and c2 type (as single feature)
c1/c2's syntactic head word
c1/c2's part of speech
token distance between concept regions for c1 and c2
1,2,3-grams and semgrams between c1 and c2 (with and without intervening nouns in the region)
1,2,3-grams to left and right of left and right concept regions, within window of 4 tokens
c1/c2 and its adjacent preposition to left/right
c1 and c2 are conjuncts
c1/c2 assertion value

Results

For evaluation, I2B2 Challenge organizers provided a new data set comprising 477 patient records. A reference set of 45,009 "correct" concept annotations was provided for the annotations task. A reference set of 18,550 problem annotations was provided prior to the relations task evaluation and a reference set of 9,070 correct relation annotations was made available afterwards. Results reported here are based on the reference sets provided.

For the concept task (Table 1), the overall F score was .76 if only the text span is considered and .74 for matching both span and type. This score assumes an exact match between the text span in the reference corpus and that output by the system. The F scores for inexact span matches on these two conditions were .85 and .84 respectively.

Concept span	True pos	False neg	False pos	recall	precision	F score
all	31756	13253	6702	0.705	0.826	0.761
problem	12823	5727	3042	0.691	0.808	0.745
treatment	9548	4012	1915	0.704	0.833	0.763
test	9385	3514	1745	0.728	0.843	0.781
span w/ matching class						
all	31005	14004	7453	0.689	0.806	0.743
Problem	12650	5900	3339	0.682	0.791	0.733
treatment	9257	4303	2173	0.683	0.810	0.741
test	9098	3801	1941	0.705	0.824	0.760

Table 1. Results for concepts (Exact span match)

For the assertion task (Table 2), the overall F score was .92 (compared to a baseline of .70 for classifying all instances as "present"). This score reflected good classification performance for most categories (especially the frequent classes *present* and *absent*), overcoming lower performance for the less frequent *conditional* and *possible* categories.

Category	**true pos**	**false neg**	**false pos**	**recall**	**precision**	**F score**
All	17128	1422	1422	0.923	0.923	0.923
Present	12558	467	801	0.964	0.94	0.952
Absent	3372	237	233	0.934	0.935	0.935
Possible	455	428	199	0.515	0.696	0.592
Hypothetical	591	126	139	0.824	0.81	0.817
Conditional	37	134	39	0.216	0.487	0.3
Associated with s/e	115	30	11	0.793	0.913	0.849

Table 2. Results for assertion task

For the relations task, we found 26,197 concept pairs in the patient records, from which our relation identification process (subtask 1) labeled 9,005 as related. Comparing these against the reference set of 9,070 related pairs (Table 3) reveals that 74% of our relations were true positives (6653 out of 9005) while we missed 2471 pairs. The distribution of scores by actual relation type was fairly even, although the .559 F score for PIP (relations between two problems) shows that this category was the most difficult to detect. Note that "unrelated" pairs (i.e., those potentially categorized as NoneTrP, NonePP, NoneTeP) were not included as categories in the computation of the scores in Table 3.

Category	**true pos**	**false neg**	**false pos**	**recall**	**precision**	**F score**
all	6653	2417	2352	0.734	0.739	0.736
TrIP	122	76	18	0.616	0.871	0.722
TrWP	99	44	7	0.692	0.934	0.795
TrCP	336	108	176	0.757	0.656	0.703
TrAP	1929	558	761	0.776	0.717	0.745
TrNAP	139	52	10	0.728	0.933	0.818
PIP	997	989	582	0.502	0.631	0.559
TeRP	2616	417	655	0.863	0.8	0.83
TeCP	415	173	143	0.706	0.744	0.724

Table 3. Results of relation identification (subtask 1)

As seen in Table 4, the overall F score for relation classification (subtask 2) was .663. Excluding the outlier class TrWP (.026), individual class performance ranged from a high of .804 to .241, with microaveraged score of .52.

To evaluate the performance of relation classification independent of relation identification, we ran our

SVM relation classification model over the reference set of 9070 correct concept pairs. This yielded an overall F score of .89

Category	true pos	false neg	false pos	recall	precision	F score
all	5992	3078	3013	0.661	0.665	0.663
TrIP	47	151	42	0.237	0.528	0.328
TrWP	2	141	9	0.014	0.182	0.026
TrCP	249	195	314	0.561	0.442	0.495
TrAP	1817	670	1059	0.731	0.632	0.678
TrNAP	30	161	28	0.157	0.517	0.241
PIP	997	989	582	0.502	0.631	0.559
TeRP	2571	462	791	0.848	0.765	0.804
TeCP	279	309	188	0.474	0.597	0.529

Table 4 Results of relation classification (subtask 2)

Discussion

For the concept task, we were surprised by the modest contribution from the semantic features derived from the UMLS® semantic network. Adding the semantic features only improved the concept detection accuracy by about one percentage point in F score. We were also somewhat surprised that the system did as well as it did with just very simple contextual features. While this issue deserves more study, we suspect that linguistic conventions for describing concepts within the sublanguage of medical records may be responsible.

For the assertion and relation tasks, syntactic complexity and variability, along with the highly unbalanced class distributions, posed serious challenges for our statistical supervised learning approach. Nonetheless, for many categories, our models were reasonably effective. Shallow syntactic features such as n-grams, the presence of an intervening noun or an adjacent preposition or adjective served well in lieu of more sophisticated sentential processing. Short “synonym” lists compensated to some extent for the low frequency of some diagnostic lexical features. Regularities in medical terminology and morphology could be exploited to make up in part for the lack of deeper domain knowledge.

The large range in F scores for individual classes may reflect where we dedicated our effort tuning features; we focused our attention on those prediction errors for which lexical cues in the text were most salient. While further tuning could well improve performance in some categories, others are likely to require deeper analysis. For the TrWP category (our worst performing relation class), text instances were extremely idiosyncratic and usually required both medical knowledge and complex syntactic analysis to interpret. For example, consider the sentence:

In 1980 she had quadruple coronary artery bypass graft surgery by Dr. Elks at Feargunwake Otacaa Community Hospital and did well until 1988 when she had exertional angina and a positive stress test and found that three or four grafts were occluded .

It is unlikely that superficial features could do well on such sentences even if many more examples were provided in the training data. Similarly, identifying whether two problems are related is a matter of medical intuition for which explicit textual clues may not exist at all in many instances.

This suggests that a hybrid two-step approach might be appropriate. The first step would use shallow lexical and syntactic features to classify instances into categories for which such cues are sufficiently powerful. A second step would apply more sophisticated syntactic or domain knowledge to classify the remaining instances.

Conclusions

The nature of the task, the quantity and quality of training data, and the choice of features all play critical roles in the determining the feasibility of machine learning approaches. Our results for the I2B2 2010 challenge suggest that statistical machine learning using primarily shallow lexical and syntactic features can provide good classification accuracy for many classes even though a few classes appear to rely more heavily on knowledge not expressed in text or require more sophisticated linguistic inference.

References

[1] Hillestad R, Bigelow J, Bower A, Girosi F, Meili R, Scoville R, Taylor R. Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. Health Affairs, 24:5, 1103-1117, 2005.

[2] Lafferty J, McCallum A, Pereira F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning* (ICML-2001), 2001.

[3] Vapnik V, Golowich S, Smola A. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In: Neural Information Processing Systems. Cambridge, MA: MIT Press; 1997.

[4] Marneffe M de, MacCartney B. and Manning C. Generating Typed Dependency Parses from Phrase Structure Parses. *LREC 2006.*

[5] McCallum A. MALLET: A Machine Learning for Language Toolkit. url: mallet.cs.umass.edu

[6] Ramshaw L and Marcus M. Text Chunking using Transformation-based Learning. In Proceedings of the Third Workshop on Very Large Corpora.

[7] UMLS url: semanticnetwork.nlm.nih.gov

[8] WEKA url: www.cs.waikato.ac.nz/ml/weka