A flexible framework for recognizing events, temporal expressions, and temporal relations in clinical text

Kirk Roberts, Bryan Rink, Sanda M Harabagiu

Human Language Technology Research Institute, The University of Texas at Dallas, Richardson, Texas, USA

Correspondence to
Kirk Roberts, Human Language Technology Research Institute, University of Texas at Dallas, PO Box 830688; MS EC31, Richardson, TX 75083-0688, USA; kirk@hlt.utdallas.edu, kirkroberts@gmail.com

Received 3 January 2013
Revised 9 April 2013
Accepted 13 April 2013
Published Online First 18 May 2013

ABSTRACT

Objective To provide a natural language processing method for the automatic recognition of events, temporal expressions, and temporal relations in clinical records.

Materials and Methods A combination of supervised, unsupervised, and rule-based methods were used. Supervised methods include conditional random fields and support vector machines. A flexible automated feature selection technique was used to select the best subset of features for each supervised task. Unsupervised methods include Brown clustering on several corpora, which result in our method being considered semisupervised.

Results On the 2012 Informatics for Integrating Biology and the Bedside (i2b2) shared task data, we achieved an overall event F1-measure of 0.8045, an overall temporal expression F1-measure of 0.6154, an overall temporal link detection F1-measure of 0.5594, and an end-to-end temporal link detection F1-measure of 0.5258. The most competitive system was our event recognition method, which ranked third out of the 14 participants in the event task.

Discussion Analysis reveals the event recognition method has difficulty determining which modifiers to include/exclude in the event span. The temporal expression recognition method requires significantly more normalization rules, although many of these rules apply only to a small number of cases. Finally, the temporal relation recognition method requires more advanced medical knowledge and could be improved by separating the single discourse relation classifier into multiple, more targeted component classifiers.

Conclusions Recognizing events and temporal expressions can be achieved accurately by combining supervised and unsupervised methods, even when only minimal medical knowledge is available. Temporal normalization and temporal relation recognition, however, are far more dependent on the modeling of medical knowledge.

INTRODUCTION

With the ever-growing importance and utilization of electronic medical records (EMR), there is increasing demand for techniques to reconstruct the patient's clinical history automatically from natural language text. Central to this is the ability to extract clinically relevant events and place them on a timeline. This would enable automated reasoning about a patient's conditions. Causal relationships are difficult to establish without temporal relationships. Furthermore, without temporal information, one cannot distinguish between a condition that occurred years ago and one that has occurred within a patient's current hospital visit, and therefore whether it might be related to other symptoms the patient is experiencing.

The 2012 Informatics for Integrating Biology and the Bedside (i2b2) shared task[1] focuses on the recognition of temporal relations between events and temporal expressions in clinical documents. It follows the TimeML standard for representing events (EVENT), temporal expressions (TIMEX3), and the relations between them (TLINK).

This article presents three distinct methods for resolving these three tasks. These methods are characterized by the central role of supervised techniques, while additionally using temporal normalization rules and unsupervised word clustering from large, unlabeled corpora. Moreover, instead of manually selecting the best features for the supervised methods employed in each task, we rely on a flexible method automatically to select the optimal subset of features from a large set of features.

BACKGROUND

Temporal information is used in clinical documents to ground events chronologically. However, as with all natural language, temporal information may be ambiguous and requires pragmatic reasoning to be fully grounded. Consider the following sentence from a clinical progress note:

> Cardiovascular stable, significant hypertension was noted on 9/7/93 at 5:10 a.m. and therefore 10 cc per kilo albumin was given.

This sentence discusses a change in a patient's medical condition and the response that was taken by the hospital. In order to represent this response formally, several steps are necessary. First, the relevant medical events must be noted. Second, temporal expressions must be extracted and grounded on a timeline. Finally, relations must be detected between these events and temporal expressions in order to enable chronological reasoning about the clinical note.

Temporal relation recognition has been well studied in non-clinical settings. Evaluations for this task on newswire include the TempEval-1[2] and TempEval-2[3] challenges, as well as an upcoming TempEval-3[4] challenge. The interest in these tasks has generated numerous automatic methods[5–17] for recognizing events, temporal expressions, and the temporal relations between them. These automatic methods have taken a variety of different approaches: (1) rule-based classifiers incorporating real-world knowledge; (2) supervised classifiers (eg,

To cite: Roberts K, Rink B, Harabagiu SM. *J Am Med Inform Assoc* 2013;**20**: 867–875.

support vector machines (SVM), maximum entropy, and conditional random fields (CRF)) using common natural language processing (NLP) features; and (3) joint inference-based classifiers, such as Markov Logic.[15] Many methods use some combination of these, such as a supervised classifier and a limited set of rules. Our work most closely falls under the second category, as we present a largely supervised approach. We differ from these previous methods, however, by focusing on how to create a flexible method that is customizable to changes in the data, as methods for temporal information extraction in clinical text need to be robust to changes in data (different hospitals, doctors, electronic medical records (EMR) systems, etc).

Previous i2b2 tasks have studied important elements in the processing of clinical text, such as extracting medication information;[18] identifying medical concepts, assertions, and relations between concepts;[19] and recognizing co-reference relations between medical concepts.[20] The most relevant of these to the 2012 i2b2 shared task is the recognition of clinical concepts performed in the 2010 task. The three types of medical concepts in that task (PROBLEMS, TREATMENTS, and TESTS) form the core subset of the medical events studied in the 2012 task.

TASK DESCRIPTION

The 2012 i2b2 shared task[1] adapts the TimeML[21] temporal annotation standard for use on clinical data, providing 310 annotated documents (190 development, 120 test). The i2b2 standard contains three primary annotations:

- EVENT: any situation relevant to the patient's clinical timeline. EVENTS have one of six types: (1) PROBLEM (eg, disease, injury); (2) TREATMENT (eg, medication, therapeutic procedure); (3) TEST (eg, diagnostic procedure, laboratory test); (4) CLINICAL_DEPT (ie, the event of being transferred to a department); (5) EVIDENTIAL (eg, report, show); and (6) OCCURRENCE (ie, all other events). In addition, EVENTS have a polarity (POSITIVE, NEGATIVE) and modality (FACTUAL, CONDITIONAL, POSSIBLE, PROPOSED).
- TIMEX3: temporal expressions that enable EVENTS to be chronologically grounded. TIMEX3s have one of four types (DATE, TIME, DURATION, FREQUENCY) and a modifier (MORE, LESS, START, MIDDLE, END, APPROX, NA). In addition, TIMEX3 annotations have a value, which is the normalized form of the temporal expression.
- TLINK: temporal relations between either EVENTS or TIMEX3s. For the official evaluation, a simplified set of temporal relations were used (BEFORE, AFTER, and OVERLAP).

For the example sentence above, the relevant annotations are:

- EVENT{text="*Cardiovascular stable*"; type=OCCURRENCE; polarity=POSITIVE; modality=FACTUAL}
- EVENT{text="*significant hypertension*"; type=PROBLEM; polarity=POSITIVE; modality=FACTUAL}
- EVENT{text="*albumin*"; type=TREATMENT; polarity=POSITIVE; modality=FACTUAL}
- TIMEX3{text="*9/7/93 at 5:10 a.m.*"; type=TIME; mod=NA; val="1993–09–07T05:10"}
- TLINK{from="*Cardiovascular stable*"; to="*significant hypertension*"; type=BEFORE}
- TLINK{from="*significant hypertension*"; to="*9/7/93 at 5:10 a.m.*"; type=OVERLAP}
- TLINK{from="*albumin*"; to="*9/7/93 at 5:10 am*"; type=AFTER}

From these annotations, we can gather that after an OCCURRENCE ("*Cardiovascular stable*"), a PROBLEM ("*significant hypertension*") emerged around a TIME ("*9/7/93 at 5:10 a.m.*"), after which a TREATMENT ("*albumin*") was administered. To detect EVENTS, TIMEX3s, and TLINKS automatically, we have designed a large set of features to train classifiers to detect such expressions in clinical records.

FEATURES

Supervised machine learning methods are composed of three primary elements: (1) input features to represent the data; (2) a model to act as the classification function; and (3) a learning process to estimate the parameters of the model. While all three of these elements are important, we have found the proper selection of features to be especially critical for NLP tasks. Importantly, parameter estimation techniques tend to perform poorly when given highly redundant or noisy features. As our goal is to experiment with as many features as possible, we utilize a technique to select the best subset of features automatically. We first give a brief overview of the feature types used in feature selection. We then describe how, from all these potential features, the highest performing subset is automatically selected for each sub-task.

Feature types

In this article, we only provide a high-level overview of the types of features we use due to space limitations. However, we have included supplementary materials (available online only) that give detailed feature descriptions, as well as examples of the feature values on actual data from this task. The list of feature types below is largely organized by the resource that best exemplifies their purpose. Many of the discussed feature types actually correspond to dozens of specific feature types. Additional feature types were considered but discarded. The feature types are:

Common NLP features

These features (eg, bag-of-words, previous token, previous EVENT type) are simple, largely self-explanatory features common in the NLP literature. They are primarily lexical in nature or rely on specific attributes of the 2012 shared task. Sections 'event recognition', 'temporal expression recognition' and 'temporal link recognition' explain any such feature when necessary.

GENIA features

The GENIA tagger[22] is a biomedical text tagger. We use it for part-of-speech tagging, lemmatization, and phrase chunking.

UMLS features

These rely on a lexicon built from the unified medical language system (UMLS) metathesaurus.[23] The lexicon contains 4.6 million terms.

Third-party TimeML features

The 2012 i2b2 shared task largely follows the TimeML temporal annotation guidelines. Thus, third-party TimeML systems can provide a useful source of automatic annotations. The four third-party systems we incorporate are: TARSQI,[5] HeidelTime,[24] SUTime,[25] and TERNIP.[26]

i2b2 concept features

Previous i2b2 tasks evaluated concept extraction, which overlaps significantly with event extraction. We use a pre-existing system[27] trained on pre-existing i2b2 data as a source of features.

Quantitative pattern features

These are based on regular expressions that recognize nine different types of entities, largely quantitative in nature: age, date,

ICD-9 disease ID, dosage, list element, measurement, (de-identified) name, percent, and time.

Statistical word features

These features use pointwise mutual information to determine automatically the most relevant words for each output class in the data. This acts as a filter to improve the classifier''s ability to handle bag-of-words style features.

Brown cluster features

Brown clustering[28] is an unsupervised technique that has been shown to be effective when used as features in an otherwise supervised setting.[29] Brown clustering groups words by their common contexts, based on the distributional hypothesis that similar words occur in similar contexts. We created Brown clusters from 10 different corpora, including the i2b2 data itself, other medical texts, newswire, and Wikipedia.

Relation features

The discourse-level TLINKs are unique within the 2012 i2b2 shared task, as they involve classifying pairs of objects (EVENTs and TIMEX3s). The relation features leverage this by providing information about how the two arguments relate, what is textually between them, and characteristics of one or both arguments.

In total, close to 300 features were available to the feature selector for each of the sub-tasks. The supplementary materials (available online only) describe the candidate feature types in more detail and provide example feature values of sentences from the i2b2 data.

Feature selection

When all possible features are incorporated into a model, the result is a model that is often too big to fit into memory, too slow to train on a single processor, and—most importantly—performs worse than a model with a few well-chosen features. This problem is traditionally called the curse of dimensionality, but it can typically be described by more practical causes. First, many features are noisy, incomplete, or not well suited for a particular task. Second, features are commonly redundant, expressing only minor differences in the data. Natural language tasks are complex, requiring many different types of information, and encoding this information into features that are both noise free and non-redundant is virtually impossible. The task of feature engineering, therefore, is to determine experimentally the best subset of features. Not only does manual experimentation not scale to hundreds of features, but it commonly results in one-size-fits-all solutions in which multiple classifiers use the same feature set even though those features may be suboptimal for a given task.

In this article, we aim to overcome this feature engineering bottleneck by utilizing automated feature selection for the extraction of temporal information in clinical texts. This flexible framework allows our system to determine automatically the best set of features for each sub-task given (a) a large collection of features, (b) a classifier, (c) a set of labeled data, and (d) a feature selection strategy. The features, as described above, are considered at the type level (eg, previous token) instead of the specific instantiation level (eg, previous token is *the*). Otherwise, instead of selecting from among hundreds of features, it would be selecting among tens of millions. The classifier is largely a 'black box' function: given data and features, return a score. The score is obtained through a fivefold cross-validation on the training data. For sequence classification (eg, EVENT boundary detection), each fold is evaluated using the F_1-measure. For multi-class classification (eg, TIMEX3 type classification), each fold is evaluated using accuracy. Instead of taking the arithmetic mean of the scores for each fold, as is typical, we use the harmonic mean, which is less susceptible to changes in a single fold, favoring moderate increases in all folds over larger increases in one or two. This results in a more conservative method for accepting new features. Given the ability to test the utility of a single feature set, the feature selection strategy dictates how features will be experimentally chosen.

The feature selection strategy we employ is known as floating forward feature selection,[30] sometimes referred to as greedy forward/backward. This greedy algorithm starts with an empty feature set and iterates until it fails to find new features that improve the cross-validation score. Let **F** be the best known feature set at the current stage of the algorithm. At the start of each iteration (the forward step), for every unused feature *g* not in **F**, the feature set $\mathbf{F} \cup g$ is given to the classifier for testing. Let g^* be the feature g that corresponds to the best performing feature set. If the score for $\mathbf{F} \cup g^*$ is greater than the score for **F** and some small margin ε (we use $\varepsilon = 0.0001$), then g^* is added to **F**. If not, the algorithm has failed to find any beneficial features and terminates. If a new feature was found, then (in the backward step), for every currently selected feature f in **F**, $f \neq g^*$, the feature set $\mathbf{F}-\{f\}$ is given to the classifier for testing. If some reduced feature set obtains a better score than **F**, the feature(s) corresponding to the reduced feature set is greedily removed.

In short, the algorithm iteratively adds the best unused feature, and after each feature is added the detrimental features are pruned. Intuitively, the backward step is necessary because, as features are added, redundancies are found. While a feature might have improved the classification performance during the second iteration, that may no longer be true during the ninth iteration. Furthermore, it is common for the first selected feature to be a good predictor, yet have a high level of noise. As more and more features are added, they act as a better combined predictor than the noisy feature, and so it may be pruned. In a manual feature engineering process, such pruning is rarely performed, resulting in diminished classification performance.

The result of this technique is not only a near optimal subset of features, but one that was achieved without manual intervention. This method is largely agnostic to both the classifier (we use both SVMs and CRFs) and data (we report results on eight separate subsets). All the features reported for EVENT and TIMEX3 recognition, as well as the section time TLINKs, were chosen by this feature selection technique. Unfortunately, there was insufficient time to incorporate discourse TLINK recognition. While it uses many of the same features, its classification method was based on a separate platform and could not be reliably integrated before the submission deadline. A post-hoc evaluation of discourse TLINKs was conducted, showing small improvements. This evaluation is discussed further in the Results section.

Event recognition

Figure 1 illustrates the architecture of our EVENT recognition system. First, we identify EVENT boundaries with a CRF classifier. Then we detect type, modality, and polarity using separate SVM classifiers. In particular, we use the Mallet[31] CRF implementation and LibLinear[32] SVM implementation. The chosen features for all EVENT classifiers are shown in table 1.

For boundary detection, the feature selector primarily chose features based on GENIA and Brown clusters. GENIA lemmas, parts of speech, and phrase chunks were all found to be important EVENT indicators. Four different Brown cluster features were chosen from four different corpora. The two medical corpora

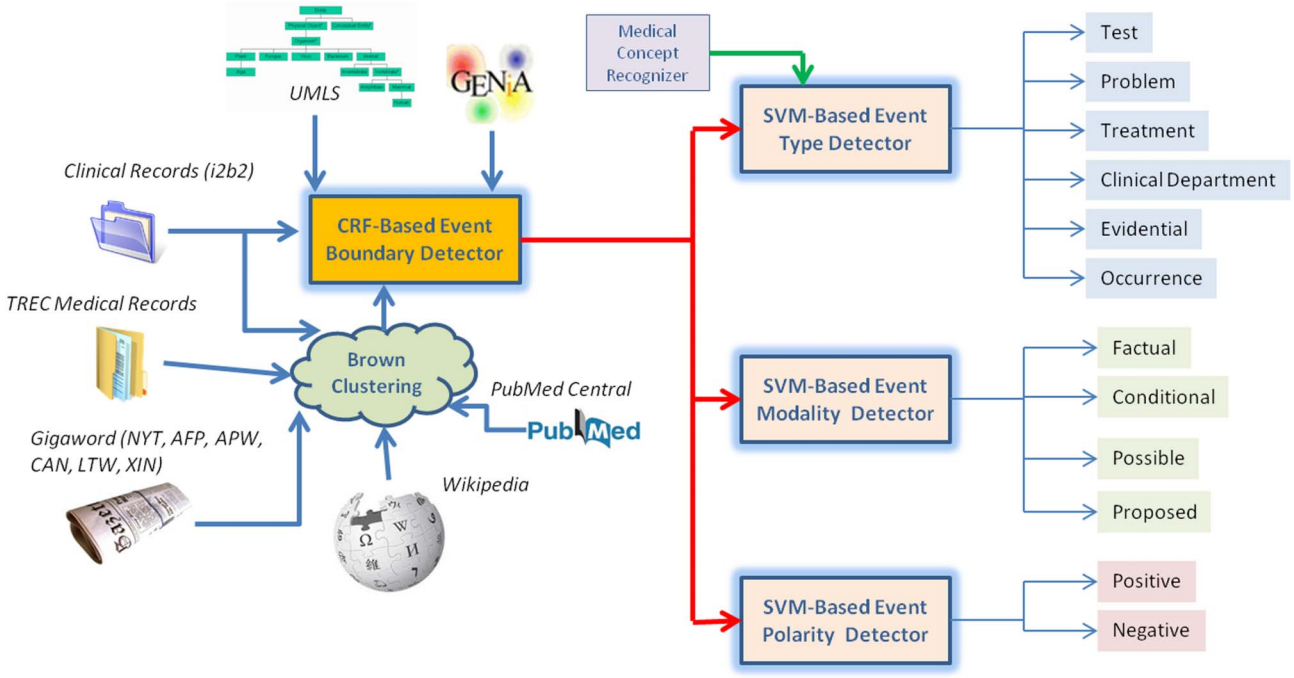

Figure 1 Event Recognition Architecture.

are likely to be useful for medical words, while the two Gigaword sub-corpora cluster more general words.

For type classification, the feature selector chose a diverse set of feature types, indicating the wide variety of ways in which Event types may be discovered. The lexical features (E2.1 and E2.2) capture the common Events from the training data (eg, *noted*, *administered*), while features E2.3 and E2.4 suggest the importance of an Event's context. Feature E2.6, the previous Event's type, was chosen because often Events of the same type occur together, such as in a list. While the UMLS feature (E2.10) is clearly useful, as UMLS is organized into a taxonomy compatible with the event types, it is interesting that several Brown cluster features (E2.11–13) were also chosen. These features indicate that the Brown clusters are able to differentiate between event types, justifying the distributional hypothesis.

For polarity and modality classification, the feature selector chose much smaller sets of features. Both include a feature indicating the previous Event's classification. Note again that these features were chosen automatically. Although it may seem logical, the decision to use the previous event's type/polarity/

Table 1 Event features

Boundary features	Type features
(E1.1) GENIA previous token lemma	(E2.1) Event text (uncased)
(E1.2) GENIA next token part-of-speech	(E2.2) Event last token (uncased)
(E1.3) GENIA next token phrase chunk IOB	(E2.3) Previous token
(E1.4) GENIA 1-token lemma context	(E2.4) Previous bigram (uncased)
(E1.5) UMLS category IOB	(E2.5) Contains punctuation
(E1.6) Brown cluster (TREC-1000)	(E2.6) Previous Event type
(E1.7) Brown cluster (i2b2–100)	(E2.7) i2b2 Concept type (exact)
(E1.8) Brown cluster prefix (XIN-100, 2)	(E2.8) i2b2 Concept type (overlap)
(E1.9) Brown cluster prefix (CNA-1000, 6)	(E2.9) TARSQI event polarity
	(E2.10) UMLS category prefix (5)
	(E2.11) Brown cluster prefix (TREC-100, 4)
	(E2.12) Brown cluster (TREC-1000)
	(E2.13) Brown cluster (Pubmed-100)
	(E2.14) Unigram PMI>0
	(E2.15) Sentence Unigram PMI strongest type (sum)
Polarity features	**Modality features**
(E3.1) Event unigrams	(E4.1) Previous token lemma
(E3.2) Previous Event polarity	(E4.2) Previous previous token lemma
(E3.3) Indexed previous token lemmas	(E4.3) Previous Event modality

PMI, pointwise mutual information.

Table 2 TIMEX3 features

Boundary features	Type features
(T1.1) Current token (numbers replaced with '0')	(T2.1) TIMEX3 unigrams (uncased)
(T1.2) Next token	(T2.2) TIMEX3 text (uncased)
(T1.3) 6-character token prefix	(T2.3) Previous token
(T1.4) 4-character token suffix	(T2.4) GENIA coarse part-of-speech
(T1.5) Quantitative pattern IOB	
(T1.6) GENIA next token lemma	
(T1.7) GENIA previous token lemma	**Mod features**
(T1.8) GENIA 1-token part-of-speech context	(T3.1) TIMEX3 unigrams (uncased)
(T1.9) GENIA 4-token lemma context	(T3.2) Previous token (uncased)
(T1.10) GENIA 4-token phrase chunk context	(T3.3) Contains uppercase
(T1.11) GENIA 6-token phrase chunk context	(T3.4) GENIA coarse part-of-speech
(T1.12) Third-party TIMEX3 IOB (all systems)	(T3.5) GENIA part-of-speech trigrams (replace verbs)
(T1.13) Third-party TIMEX3 IOB (HeidelTime)	
(T1.14) Third-party TIMEX3 IOB (TERNIP)	

modality for each of the respective sub-tasks is based entirely on the data (via feature selection) without any manual intervention. For instance, if the previous event's polarity had been useful for modality classification, it would have been included by the feature selector. Thus this automatic method still produces a logical feature set for each task.

Temporal expression recognition

The architecture of our temporal expression recognition method is shown in figure 2. Similar to EVENT annotation, a CRF classifier recognizes TIMEX3 boundaries and two SVM classifiers determine the type and modifier. A rule-based method then normalizes the TIMEX3. The chosen features for all TIMEX3 classifiers are shown in table 2.

For boundary detection, the feature selector chose a combination of lexical and syntactic features, along with utilizing third-party TIMEX3 systems. The basic text feature (T1.1) generalizes the text by replacing all non-zero digits with a zero. For instance, both *10:05 am* and *11:14 am* become *00:00 am*. The prefix (T1.3) and suffix (T1.4) features provide a functionality similar to lemmatization. The quantitative patterns (T1.5) contain regular expressions for times and dates, so it is not surprising they prove useful for detecting TIMEX3s. Unlike EVENTs, the third-party methods were quite useful for detecting TIMEX3s. HeidelTime (T1.13) and TERNIP (T1.14) in particular were chosen in addition to the combination of all systems (T1.12).

For type and modifier classification, the feature selector largely chose lexical features, including a bag-of-words (T2.1, T3.1), a complete TIMEX3 span (T2.2), and a previous token (T2.3, T3.2) feature. Lexical cues are good indicators of the type of TIMEX3, words like *am* indicates TIME, and *daily* indicates FREQUENCY. Beyond the lexical features, the coarse-grained part-of-speech (T2.4, T3.4) feature was chosen to generalize the TIMEX3's syntactic category. Feature T3.3 indicates whether the TIMEX3 contains any uppercase characters. The final feature (T3.5) is a part-of-speech trigram replace feature, returning the parts of speech of all three-word sequences within the TIMEX3, but replacing verbal parts of speech with the verb itself.

Our TIMEX3 normalization (val) method is rule based, combining custom-built rules with TIMEN,[33] an open-source temporal normalizer. In combination with TIMEN, we use two types of rules: (1) pre-TIMEN rules that fully normalize temporal