An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge

Yan Xu,[1,2] Yining Wang,[2,3] Tianren Liu,[2,3] Junichi Tsujii,[2] Eric I-Chao Chang[2]

▶ Additional material is published online only. To view please visit the journal online (http://dx.doi.org/10.1136/amiajnl-2012-001607).

[1]State Key Laboratory of Software Development Environment, Key Laboratory of Biomechanics and Mechanobiology of Ministry of Education, Beihang University, Beijing, China
[2]Microsoft Research Asia, Beijing, China
[3]Department of Computer Science, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

Correspondence to
Dr Eric I-Chao Chang, Microsoft Research Asia, T2-14463, No. 5 Danling Street, Haidian District, Beijing 100080, P.R. China; eric.chang@microsoft.com

Received 1 January 2013
Revised 12 February 2013
Accepted 17 February 2013
Published Online First 6 March 2013

To cite: Xu Y, Wang Y, Liu T, *et al*. *J Am Med Inform Assoc* 2013;**20**: 849–858.

ABSTRACT

Objective To create an end-to-end system to identify temporal relation in discharge summaries for the 2012 i2b2 challenge. The challenge includes event extraction, timex extraction, and temporal relation identification.

Design An end-to-end temporal relation system was developed. It includes three subsystems: an event extraction system (conditional random fields (CRF) name entity extraction and their corresponding attribute classifiers), a temporal extraction system (CRF name entity extraction, their corresponding attribute classifiers, and context-free grammar based normalization system), and a temporal relation system (10 multi-support vector machine (SVM) classifiers and a Markov logic networks inference system) using labeled sequential pattern mining, syntactic structures based on parse trees, and results from a coordination classifier. Micro-averaged precision (P), recall (R), averaged P&R (P&R), and F measure (F) were used to evaluate results.

Results For event extraction, the system achieved 0.9415 (P), 0.8930 (R), 0.9166 (P&R), and 0.9166 (F). The accuracies of their type, polarity, and modality were 0.8574, 0.8585, and 0.8560, respectively. For timex extraction, the system achieved 0.8818, 0.9489, 0.9141, and 0.9141, respectively. The accuracies of their type, value, and modifier were 0.8929, 0.7170, and 0.8907, respectively. For temporal relation, the system achieved 0.6589, 0.7129, 0.6767, and 0.6849, respectively. For end-to-end temporal relation, it achieved 0.5904, 0.5944, 0.5921, and 0.5924, respectively. With the F measure used for evaluation, we were ranked first out of 14 competing teams (event extraction), first out of 14 teams (timex extraction), third out of 12 teams (temporal relation), and second out of seven teams (end-to-end temporal relation).

Conclusions The system achieved encouraging results, demonstrating the feasibility of the tasks defined by the i2b2 organizers. The experiment result demonstrates that both global and local information is useful in the 2012 challenge.

INTRODUCTION

Temporal relation is a current popular topic in information retrieval. Its ultimate goal is to build a timeline for every event in a document. In the medical domain, such a timeline can be used to track the status of patients' diseases, determine the cause of patients' diseases, monitor the status of patients' operations, and confirm both the effectiveness and side-effects of drugs. In this task, tLink (temporal relation) is defined as the relation between two events, an event and a timex (temporal expression), two timexes, and an event and the section time of a document. In this paper, concepts are used to refer to both events and timexes.

In this i2b2 challenge,[1] event refers to anything that is relevant to the patient's clinical timeline in a medical record. An event is recorded in the form <EVENT id="EX" start="start_position" end="end_position" text="<event string>" modality="<modality category>" polarity="<polarity category>" type="<type category>" />, where modality category is denoted as <FACTUAL|CONDITIONAL|POSSIBLE|PROPOSED>, polarity category is denoted as <POSITIVE|NEGATIVE>, and type category is denoted as <PROBLEM|TREATMENT|TEST|OCCURRENCE|EVIDENTIAL|CLINICAL_DEPT>. Timex refers to phrases that contain time information. The format of a timex is <TIMEX3 id="TX" start="start_position" end="end_position" text="<timex string>" type="<type category>" val="<normalization value>" mod="<modality category>" />, where type category is denoted as <DATE|TIME|DURATION|FREQUENCY> and modality category is denoted as <MORE|MIDDLE|LESS|START|END|APPROX|NA>. tLink refers to the relation between event and event, event and timex, timex and timex, event and section time, or timex and section time. A tLink is in the form <TLINK id="TLX" fromID="EX" fromText="<concept string>" toID="EX" toText="<concept string>" type="<type category>"/>, where type category is denoted as <OVERLAP|BEFORE|AFTER>. It is noted that we define a new relation type 'none,' which means the temporal relation is unlikely to be determined. Section time is admission day and discharge day recorded in discharge summaries. Temporal relation exists either in the same sentences or in different sentences. In the 2012 i2b2 challenge, we accomplished three tasks including the event track, the timex track, and the tLink track. The event track is to extract events and to classify their attributes; the timex track is to extract timexes, to classify their attributes, and to infer their values; and the tLink track is to identify relation between event and event, timex and timex, event and timex, event and section time, and timex and section time.

In the timex extraction task, it is difficult to determine whether a word consisting of digits is a time expression or not. For example, in the sentence 'the pain became worse over the next few minutes to 07–12 and increased to 10–12 while he was in the ambulance,' '07–12' and '10–12' are not time expressions even if they are in correct date formats. They actually denote the range of the patient's pain. Since regular expressions can only extract phrases in some string patterns, new methods were adopted in order to solve the ambiguity problem. Therefore, we exploited rich semantic information and used a conditional random fields (CRF) model with some discriminative features to extract timexes.

In normalization, we observed that the values of some timexes cannot be directly computed and can only be inferred by contextual information. For example, to compute the value of the timex 'postoperative day five,' we must find out the exact operation date in the discharge summary. As a result, we adopted a two-stage algorithm to solve the problem. The first stage is to compute the values of temporal expressions using only local information by a context-free grammar. As for those expressions whose values cannot be determined directly, we converted them into intermediate representations. Then, the values of these intermediate representations can be inferred using several deduction rules in the second stage.

In the temporal relation task, two concepts (event, timex, or section time) in a relation may appear in different positions, which results in different properties in relation identification. (1) The relation between event and event in the same sentences depends on their syntactic structures and sentence patterns. (2) The relation between timex and event in the same sentences depends on prepositions prior to the timex and verbs in the same sentence. For example, in the sentence 'On **2 preceding days** noticed the stool to be **dark** and tarry,' the relation between '**2 preceding days**' and '**dark**' is 'overlap' due to the preposition 'on.' (3) The relation between event and event in the different sentences depends on their context and the time points in their own sentences. If the surface string of an event is admission or discharge, its value can be easily inferred. (4) The relation between event and timex in the different sentences is similar to the above relation. However, if the types of the timexes are DURATION or FREQUENCY, we still cannot capture the time point. (5) The relation between event and section time depends on the time point in the sentence of the event and section recognition (ie, determine whether an event belongs to the admission section or the discharge section). (6) The relation between timex and section time depends on section recognition and timex type, because a relation between some timexes and section time belongs to 'none' ('a β blocker was begun at 12.5 mg of Lopressor p.o. **b.i.d.** which was titrated to 25 mg at the time of discharge').

The very different properties of different types of relations require different resolvers. To this end, we designed 10 resolvers for different types of relations using multi -support vector machine (SVM) classifiers. In particular, we used 10 different multi-SVM classifiers to solve <event, event>, <event, timex>, and <timex, timex> in the same sentence; <event, event>, <event, timex>, and <timex, timex> in different sentences; and <event, admission time>, <timex, admission time>, <event, discharge time>, and <timex, discharge time>.

Furthermore, one key observation we made on the training set is that coordination information is very helpful in finding 'overlap' relations in the same sentences. For example, in the sentence 'Room air arterial blood gas showed **a pO2** of 56, **a pCO2** of 35, and **a pH** of 7.52,' there is a coordination structure among the three events and the relations between each two of them are indeed 'overlap.' Taking the above observation into consideration, we built a coordination classifier to decide whether two events in the same sentence are coordinated. The result of the coordination classifier was used as a feature in the SVM classifiers.

As the training set is labeled by different doctors, the relation <A=B>C> can be labeled in many different ways, such as <A=B, B>C>, <A=B, A>C>, or <A=B, B>C, A>C>. In addition, some relations are not even labeled and must be inferred via other relations. Therefore, a transmit closure was built.

The training set and the test set are provided by the i2b2 organizers and are collected from two hospitals. The training set includes 190 labeled discharge summaries with 34 204 tLinks, 16 619 events, and 2390 timexes. The test set consists of 120 discharge summaries with 27 736 tLinks, 13 593 events, and 1820 timexes.

Our contributions in this study are threefold. First, we proposed a two-stage normalization algorithm which can be generally applied in normalization tasks in any domain. We integrated a context-free grammar and deduction rules into the normalization algorithm, making it more effective than other normalization systems that only employ regular expressions. Second, the framework for task three is a successful one since a careful division of sub-tasks makes features homogeneous. Third, we furthermore proposed structured discriminative features such as syntactic patterns.

Related work

In the general domain, some work is related to temporal relation. Chambers *et al*[2] proposed an automatic two-stage machine learning architecture to learn temporal relations between events. The first stage is to learn attributes of single events; the second stage is to classify the relations based on an SVM model. Performance is 3% higher than the baseline method. Chambers *et al*[3] proposed an integer linear programming framework to resolve implicit global constraints of transitivity and time expression for temporal ordering. Performance is 3.6% higher than the baseline method. Yoshikawa[4] proposed a Markov logic model to jointly identify the relations of all three temporal relation types simultaneously, between events, between event and timex, and between event and document creation time. Performance is 3% higher on all three types of relations. Zhao *et al*[5] proposed a system that can perform extraction, normalization to time intervals, and relation identification. Their novel contribution is comparing two time intervals. Garrido *et al*[6] proposed a hierarchical topic model with pattern learning for relation extraction. Bunescu *et al*[7] proposed a shortest path dependency kernel for relation extraction. As we observed that usage of the above two approaches is similar to our task, we adopted similar methods for our task.

Furthermore, in recent years, several challenge tasks[8–10] on temporal relation in news articles have been devised. The workshops for TimeML have been held four times, where TimeML[8] is a special language for events and temporal expressions. The TempEval Challenges were held twice in 2007[9] and 2010.[10] In 2007, there were three tasks: temporal relation identification between events and timexes in the same sentences, temporal relation identification between events and the document creation time, and temporal relation identification between contiguous pairs of matrix verbs. The first ranked team used sentence-level syntactic tree generation. They also applied several heuristics and statistical data from the training set for inter-sentence temporal relation. In 2010, there were three main tasks: event identification, time expressions identification, and temporal relations identification. Temporal relations identification was further structured into four sub-tasks between events and time expressions within the same sentence, events and document creation time, main events in consecutive sentences, and events where one syntactically dominates the other. For extraction, HeidelTime[11] achieved the highest F measure based on regular expressions for extraction and knowledge resources and linguistic clues for normalization. The TRIPS and TRIOS system[12] approached a method based on a combination of deep semantic parsing, Markov logic networks (MLNs), and conditional random field classifiers; it achieved first place in nearly all relation identifications.

There is little existing work on temporal relation in the medical domain. Raghavan *et al*[13] drafted a new classification problem for temporal relation. The types are: before admission, admission, between admission and discharge, discharge, and after discharge. They used a CRF model combined with lexical, section-based, and temporal features from medical events in discharge summaries. The CRF model achieved an average precision and recall of 79.6% and 69.7%, while the Max-Ent model achieved 74.3% and 66.5%, respectively. In addition, there has been some work on temporal expressions. Denny *et al*[14] used regular expressions to extract temporal expressions and applied some rules to normalize the value of temporal expression. Zhou *et al*[15] proposed a system architecture for a pipeline integrated approach to perform temporal information extraction, representation, and reasoning in medical narrative reports. Although the system was comprehensive regarding the treatment of temporal expressions and inferences in the medical domain, the identification of temporal expressions was carried out by a rule-based system and regular expressions. Jung *et al*[16] proposed an end-to-end system to build timelines from unstructured narrative medical records. Their core procedure was also based on explicit rules for logical form pattern-based extraction, concept extraction, temporal expression extraction, event extraction, and timelines creation.

METHODS

In this section, we describe in detail our end-to-end system for the temporal relation challenge. Our overall approach comprises the following steps (see figure 1): event extraction with a CRF model,[17] event attribute classification, timex extraction, timex attribute classification, timex normalization, temporal relation extraction using 10 multi-SVM classifiers,[18] and a MLN[19] model for inference.

Event extraction

The extraction method is similar to that described by Xu *et al*.[20] We extracted concepts (problem, treatment, and test) using a dosage-unit dictionary to dynamically switch two models based on CRF. In addition, we also used various discriminative features including pattern matching, word normalization, N-character-prefix-and-suffix, and word clustering. For the three event categories that are not included in the 2010 i2b2 Shared Task (clinical_dept, occurrence, and evidential), we mainly used the first and last eight letters of each word as well as variants of some important words (like 'show,' 'present,' 'report,' etc) as features. The features and rules applied for polarity and modality classification are similar to those in our previous work[20] and we do not give more details here.

Timex extraction, normalization, and corresponding attribute classifiers

Our time extraction system includes temporal extraction based on a CRF model, normalization (compute the value of timexes) based on a context-free grammar algorithm (CFG), and several attributes' classifiers based on multi-SVM models.

1. *Timex extraction*: In the training set, we found that regular expressions cannot resolve some phrases which are in correct date formats but are actually not time expressions. We used a CRF model to extract temporal expressions. The features we used can be divided into the following groups: baseline word-level features, syntax features, and pattern features (see online supplementary table S1).
2. *Computation of the value of temporal expressions (normalization)*: There are two different kinds of time expressions. For some timexes, their values can be extracted directly from the text and the words surrounding them, like '2012-10-10.' However, there are some timexes whose values must be inferred from the whole document, like 'postoperative day five.' Therefore, our normalization system consists of two phases. First, values were computed using only local information. For those expressions whose values cannot be determined, we converted them into intermediate representations. Second, based on the intermediate representation, we used several deduction rules to infer the exact values of these time expressions. Figure 2 shows the flow of our normalization system.

Phase one

In phase one, we used a CFG to compute the values of timexes. Some of the symbols and rules of the context-free grammar are listed in online supplementary tables S2 and S3.

1. *Extract terminal symbols and non-terminal symbols:* Terminal symbols in our CFG grammar are those words or phrases inside or near a concept that are useful in determining the value of the concept, like cardinals, ordinals, or abbreviation of frequency expressions (see online supplementary table S2). In our system, we mainly used regular expressions to extract terminal symbols from the raw text. After the extraction step, all the other words in a concept will be simply discarded from succeeding processes.
2. *Parse the expression into a CFG parse tree by a CFG parser*: In this step, we attempted to parse the extracted terminal symbols into a CFG parse tree using our designed CFG (see online supplementary table S3 for some CFG rule examples). Some terminal or non-terminal symbols have extra properties containing more detailed information about a concept which can be used in the normalization task (see online supplementary table S3). The starting symbol is *S* associated with the only rule: $S \rightarrow Date|Time|Length|Frequency|RelativeDate$. If the starting symbol is parsed into *Date*, *Time*, *Length*, or *Frequency*, then the type and value of the concept can be determined directly by reading corresponding property values from the parse tree. Otherwise (ie, *S* is parsed into *RelativeDate*), the system must convert the concept into an intermediate representation which is used in phase two. A typical intermediate representation usually contains the original text of the concept, the anchor point (either key events like ADMISSION, OPERATION, or the time point of the current sentence, denoted by NOW) and the offset from the anchor point (eg, +2 days, −3 months). They can be easily captured from property values in the parse tree.
3. *Combinator*: Since the CFG is predetermined and cannot cover all patterns in real-world clinical records, it is possible that the CFG parser will fail when parsing some timex concepts. If the CFG parser fails, we will try to apply CFG rules in a bottom-up and greedy manner (ie, apply rules to combine symbols whenever possible). When no CFG rules can be applied, we search for the first of the five non-terminal symbols (*Date*, *Time*, *Length*, *Frequency*, and *RelativeDate*) and use it as the value or the intermediate representation of the timex concept.

Phase two

In this phase, we employed several deduction rules to infer the exact values of these time expressions, based on their

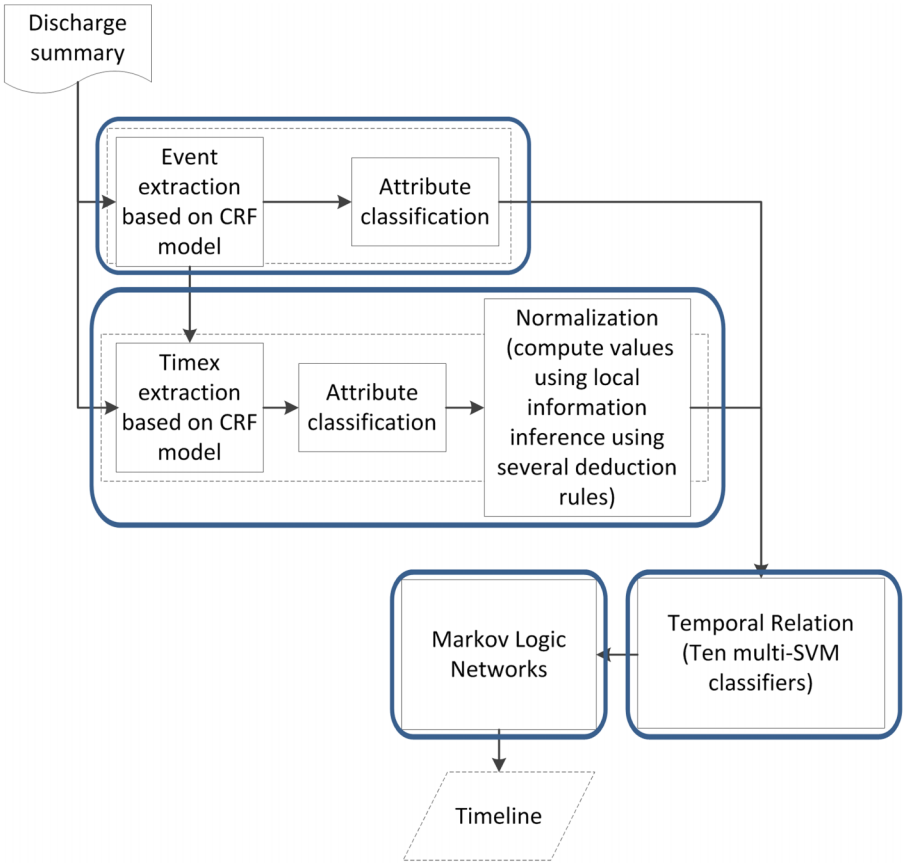

Figure 1 Flow diagram of our end-to-end system. CRF, conditional random fields; SVM, support vector machine.

intermediate representations. We used the following deduction rules repeatedly until no rules can be applied any more:

1. Determine the admission and discharge time according to the header of a discharge summary.
2. Determine the operation time: if part of a sentence matches a predetermined regular expression (eg, the sentence contains the phrase 'taken to the operation room'), then the date of the sentence is treated as the operation time.
3. Infer time using appositive concepts: if two timex concepts are appositives (eg, on postoperative day 7, 09/08), then the two timexes will be considered to be simultaneous and if the value of one timex is known, so is the other one.
4. Infer time using intermediate representations: in phase one, we have already obtained intermediate representations of several timexes. If the exact value of the anchor time is known, we can compute the exact time of the timex by simply adding the offset to the value of the anchor time. Similarly, if the exact time of the timex is known (possibly through other rules like appositive concepts), we can deduce the exact value of a key event (like the operation time).
5. Infer time using nearest timex concepts: if the value of a timex cannot be determined using the above four rules, the same value is given as the value of its nearest timex. Although this rule may result in inaccurate value estimations, it works quite well in most cases.

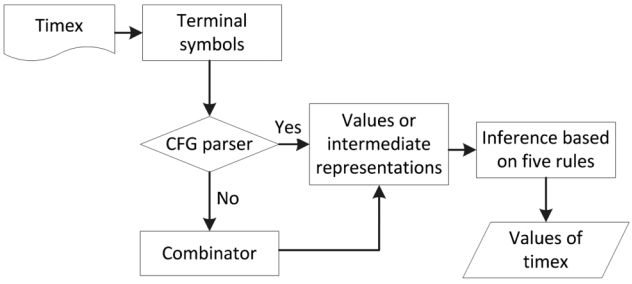

Figure 2 Flow diagram of our normalization system. CFG, context-free grammar algorithm.

One practical concern is that the five deduction rules above may lead to conflicting results (ie, different values of the same timex using different rules). In addition, due to the ambiguous nature of natural language, values inferred from some deduction rules may not be unique. For instance, the phrase 'POD #1' can mean either the operation day or the day after the operation day. As a result, we applied the following two rules regarding the above concerns:

1. The five deduction rules listed above should be carried out in a sequential manner, and when there is a conflict resulting from two different rules, the inferred value from the latter rule will be ignored.
2. For those rules whose inferred value is not unique, we simply enumerate all possible interpretations and select the one with the minimum number of conflicts. Take the phrase 'POD #1' as an example: if the operation day is February 2 and the phrase 'POD 3, February the fourth' occurs in the same discharge summary, then the value of 'POD #1' is inferred as February 2 because the other interpretation (ie, February 3) will present an extra conflict resulting from the third deduction rule (ie, inference using appositive concepts).

Temporal relation

In this task, the temporal relations can be divided into different categories: <event, event>, <event, timex>, <timex, timex>, <event, section time>, and <timex, section time>. The identification of <event, event> in the same sentences depends on syntactic structures generated from parse trees and sentence patterns. For <event, timex> in the same sentences, prepositions prior to timexes and verbs are cues to identify the relation. The identification of <event, event> and <times, event> in the different sentences are both dependent on contextual and sentence information. For <event, section time>, we captured time points of events and employed section recognition to identify this kind of relation. The relation <timex, section time> can be decided by both section recognition and timex types. Figure 3 shows the block diagram of the overall relation classification framework.

Creation of positive and negative instances

In the training set, we observed that the relation "A=B>C" may be labeled as <A=B, A>C>, <A=B, B>C>, <A=B, B>C, A>C>, <B=A, C, <B=A, C<B, C<A>, <B=A, C<A>, etc. Since not all relations are explicitly labeled by annotators, it is a formidable challenge to design a system that outputs consistent relation labels for the training set. For example, if a relation is labeled as <A=B, B>C> in a discharge summary, then the relation A>C will not be explicitly labeled. If we only use the explicitly labeled relations without inference, it is difficult to obtain the maximum margin of an SVM classifier. Consequently, when there is a C<B relation in the test set, the model may not be able to correctly obtain the result. In summary, there are two major problems in directly using the training set: (1) some transitive relations may be mislabeled as negative instances; and (2) nearly all reciprocal relations are labeled as negative ones. Therefore, we proposed a two-step method which is aimed at overcoming these two limitations. First, 'transitive closure' was used on the training set and thus all transitive relations will be labeled correctly. Second, if there is a relation A>C in the training set, we also consider C<A as positive instances. The method not only solves the above two problems, but also expands the number of relations in the training set.

Sentence normalization

To make the parsers work more accurately, we replaced each concept of type 'problem,' 'treatment,' 'test,' and 'clinical_dept' with a placeholder since the details of these four types have no impact on the result (see online supplementary table S4, 'Normalization'). After this, we used the Enju parser[21] and the Stanford parser[22] to parse the normalized text (see online supplementary table S4, 'Stemming').

Classifiers for relations in the same sentences

Compared to the other two types of relations, relations in the same sentences are easier to identify because there are more cues in sentences. In our method, multi-SVM classifiers were

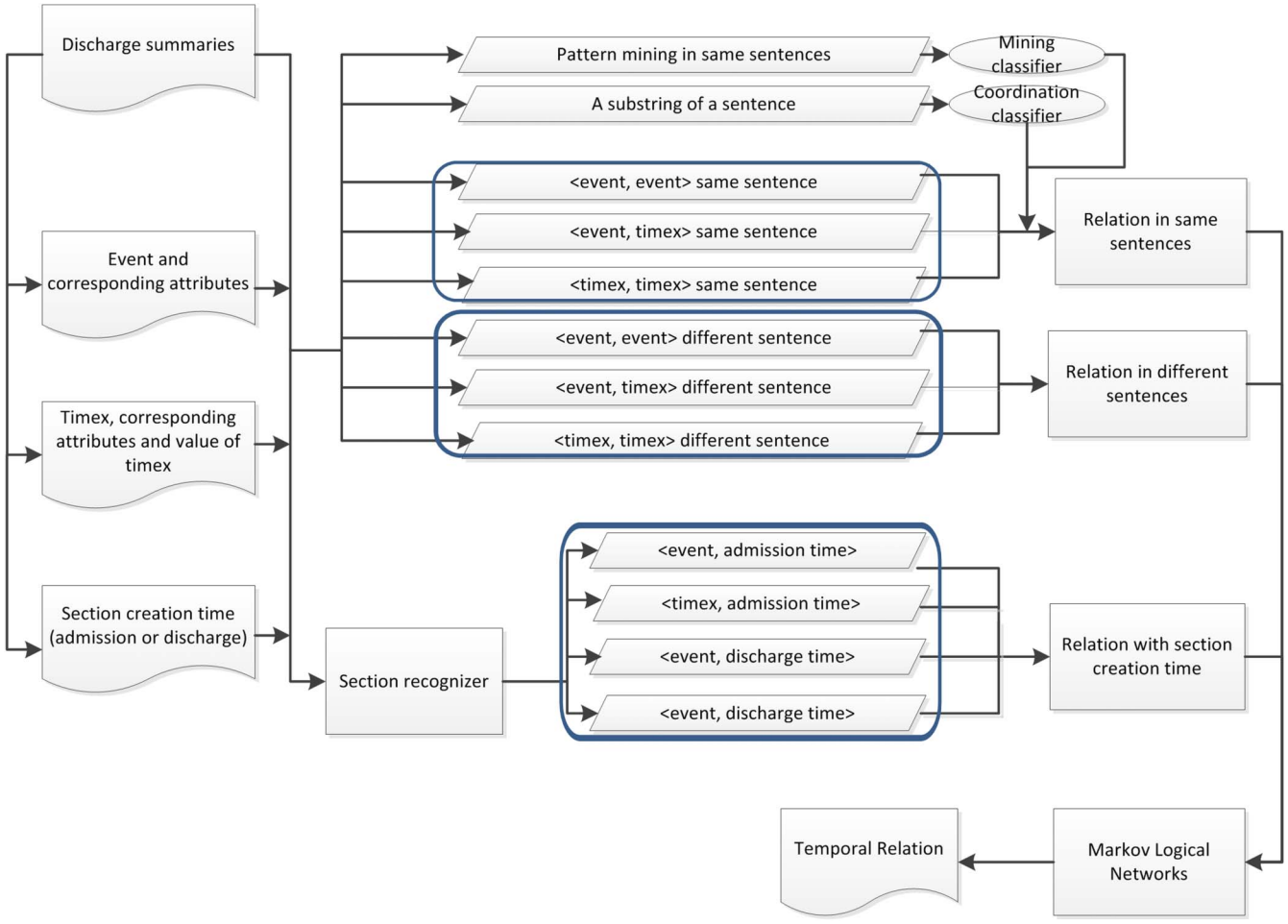

Figure 3 The overall temporal relation classification method.

used as classifiers to classify each pair of concepts into four possible relations: overlap, after, before, and none. We also used different features to incorporate local and global information into our system. Please see online supplementary table S5 for a list of the features we used. We also describe features that are specially designed for this task as follows.

Feature 1: Syntax features built on parse trees

Parse trees are crucial to our temporal relation extraction system, because they can reveal syntactical relations between a pair of concepts even if many words are in between. In our system, we mainly used two parsers: the Enju parser[21] and the Stanford parser.[22] Our syntax features can be divided into three categories:

- *Dependencies* (StanfordDependency, EnjuDependency, StanfordDependency_norm, EnjuDependency_norm): Dependencies reveal the syntactical relations between two concepts. In our system, the collapsed typed dependencies were used between a pair of concepts in both original sentences and normalized sentences as features.
- *Paths on parse trees* (StanfordTreePath, EnjuTreePath): For long-distance concept pairs, few direct dependencies are given by parsers. In such cases, we used the path linking the two nodes in the parse tree as features. We added the syntactic categories of each node on the path, and removed redundant syntactic categories, such as NX, VX, etc. We only used the normalized sentences as inputs for these features. Apart from syntactic categories, we also added head words that are prepositions (such as 'for,' 'in,' etc) because these are the most important words that imply the temporal relations of concepts. In the example sentence, the path between concepts 'the cardiac surgery recovery unit' and 'continuing care' is 'NN, PP (for), NP, NN,' using the Enju parser (see figure 4).
- *Dependency chains in dependency graphs* (EnjuDependencyChain, EnjuDependency): Using typed dependencies derived from parsers, we constructed a dependency graph whose nodes are words in the sentence and edges are dependency relations. We then found the shortest path between the two concepts in the dependency graph, and used this path as features.

Normally, only syntactical categories or part-of-speech tags on the nodes of the shortest path should be included to alleviate the sparseness of the feature. However, to make the feature more powerful, we made the following modifications: (1) one node was picked at each time to use its stemmed word instead of its syntactic categories or part-of-speech tags on the path. This does not make the feature too sparse, but by doing this we can capture the semantic information of those important words; (2) the direction of the dependency was also added for each path. For instance, if the verb 'transferred' is the first argument for the preposition 'on,' it should be written as 'transferred→on,' or 'on ←transferred.' This guarantees the consistency of the semantics of the word 'on'; (3) the polarity of each word was added. For instance, if the verb 'transferred' is negated (eg, he was not transferred… on hospital day two), the dependency path should be

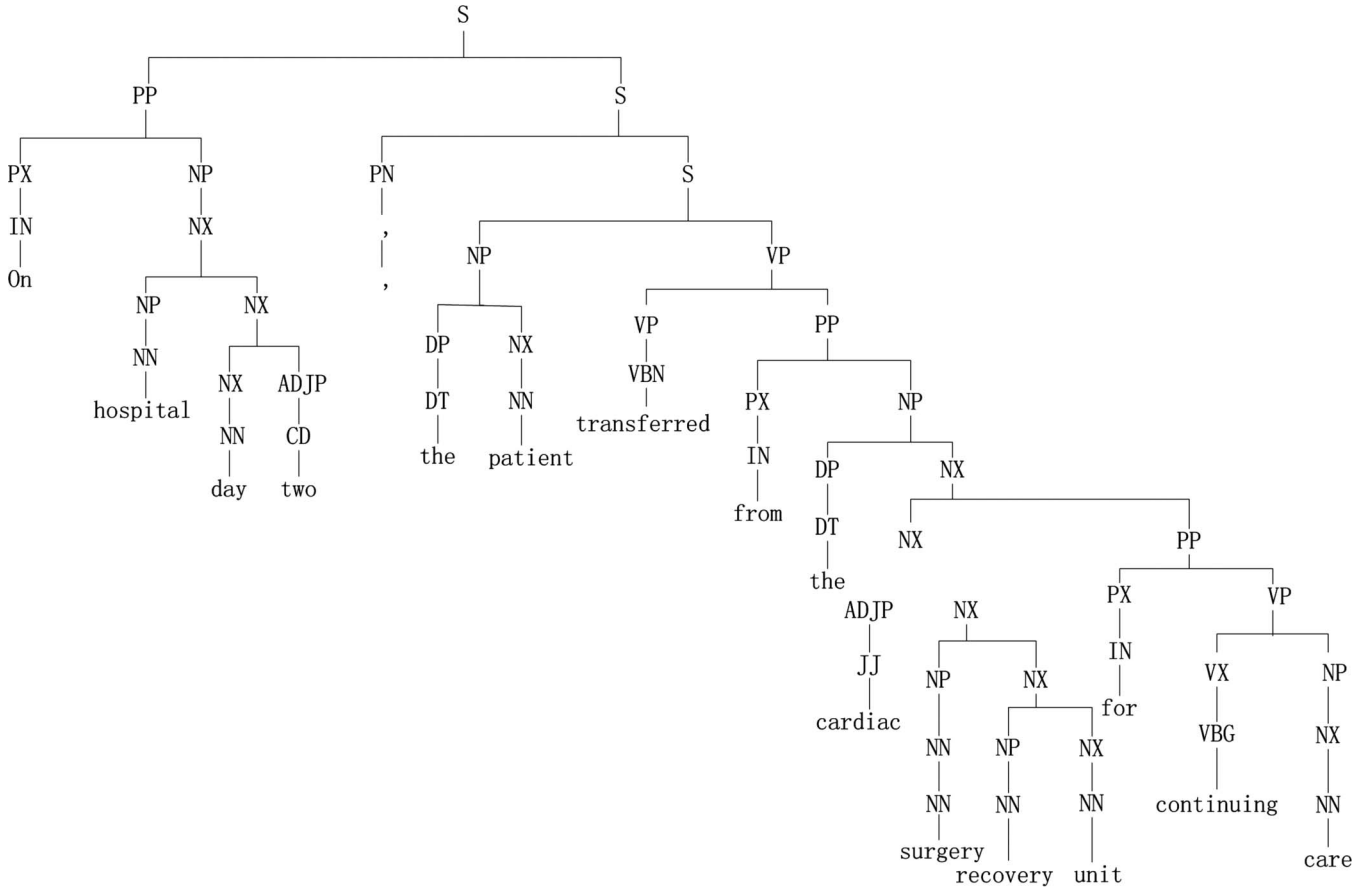

Figure 4 Enju parser results for 'On hospital day two, the patient was transferred from the cardiac surgery recovery unit for continuing care.'

written as '(-)transferred→on' or 'on←(-)transferred.' This guarantees the consistency of the verb 'transfer.' In the example sentence, possible dependency paths between 'hospital' and 'hospital day two' are < Hospital←from←VP→IN→Date, Hospital←IN←transfer→IN→Date, Hospital←IN←VP→on→Date> (see figure 4).

Feature 2: Features based on labeled sequential pattern mining

The structures of sentences assist the identification of relations. To exploit the structure of sentences, we employed labeled sequential pattern (LSP) mining, which was first proposed by Agrawal and Srikant[23] and has been widely used in various research fields such as question detection in question–answer pairs mining,[24] and erroneous sentences detection in writing English papers.[25] Therefore, we used normalized text to mine patterns in sentences (see online supplementary table S4). We then selected patterns with high frequency as features. The threshold for the final set of patterns was selected from cross-validation using the training set. In our system, 896 patterns were selected.

Feature 3: Coordination-class features

The identification of 'overlap' relations benefits from coordination-class features. In our system, a binary SVM classifier was trained to detect coordinated phrases in sentences. To train the coordination classifier, we created positive instances and negative instances in the following way. The positive instances were picked up from sentences containing overlap relations. We truncated these sentences into parts by first discarding all text before the first event. After this, we manually selected parts consistent with coordination syntax. For example, in the sentence 'Room air arterial blood gas showed **a pO2** of 56, **a pCO2** of 35, and **a pH** of 7.52,' the positive instance is '**a pO2** of 56, **a pCO2** of 35, and **a pH** of 7.52.' Negative instances were samples from the remaining sentences. The feature set for the classifier is given in online supplementary table S6. The high performance (F measure 0.8931) of this binary classifier makes the feature very reliable.

Classifiers for relations in the different sentences

In our method, multi-SVM classifiers were used as classifiers (overlap, after, before, none). Compared with the size of the training set in the same sentences, the positive size of the training set in the different sentences is smaller. Based on the fact that there are too many pairs between the different sentences which results in the large amount type 'none' contributes, we decided to only use concepts in neighbor sentences as pairs. Because concepts are in neighbor sentences, we introduced two kinds of features: sentence features and features of concept itself (see online supplementary table S7). Since type <timex, timex> is rare, we only used two features for this type: prepositions (before, after) and keywords (next, follow, post). We used three classifiers which are <event, event>, <timex, timex>, and <event, timex>.

Classifiers for relations between concept and section time

Guideline events must be linked to either the admission date or the discharge date. Hence, it is crucial and significant to recognize sections. After section recognition,[26] since every event is only linked with one section, we reduced the classifications from four (before, after, overlap, none) to three (before, after, overlap) in multi-SVM classifiers, and thus we successfully improved precision. In cross-validation using the training set, the F measure is improved from 0.8810 to 0.9102 (only <event, section time>) (see table 4).

We also observed that aside from the <event, section time> relation, there is also the <timex, section time> relation. However, many <timex, section time> pairs belong to relation 'none.' Therefore, we added two more classifiers, which are similar to those of events, with the only difference being on the features they used (see online supplementary table S8).

Inference using MLN

We observed that there is no obvious cue of either global information or local information between a few relations and they can only be inferred from other relations. Hence, an SVM classifier is unable to find such relations. Due to limited development time, we used MLN with the confidences of SVM as input to infer the relations. The MLN code was implemented from http://code.google.com/p/thebeast/19. The process is as follows. The confidences of SVM are considered as initialized values. If two existing relations can infer a new relation such as (<A=B, B<C> => A<C), the confidence of the new relation is decided by the confidences of the two existing relations: $f_{new} = \max(f1_{existing}, f2_{existing}) + C$, where C represents various relation types and the value of C is learned by the MLN model. $f1_{existing}$, $f1_{existing}$, and $f2_{existing}$ are the corresponding confidences. After sufficient relations were inferred between two events, the relation with the highest confidence is considered as the output.

Table 1 Submitted systems

Event extraction	
System 1	All features
Timex extraction	
System 1	Training data relabeled according to the latest annotation guideline
System 2	The original training set
System 3	All features plus regular expressions extracting concepts that are rare in the training set, such as ; and '97 (relabeled data)
tLink	
System 1	All features excluding the results of coordination classification and labeled sequential pattern mining without Markov logical networks
System 2	All features without Markov logical networks
System 3	All features with Markov logical networks
End-to-end tLink	
System 1	System 1 (Event Extraction)+System 1 (Timex Extraction)+System1 (tLink)

Table 2 Micro-averaged results for extraction and tLink in discharge summaries

Event	Precision (P)	Recall (R)	Average P&R	F measure	Type	Polarity	Modality	F measure (best)
System 1	0.9415	0.8930	0.9166	**0.9166**	0.8574	0.8585	0.8560	0.9166
Timex						Value	Modifier	
System 1	0.8814	0.9500	0.9144	**0.9144**	0.8929	0.7159	0.8918	
System 2	0.8818	0.9489	0.9141	0.9141	0.8929	0.7170	0.8907	
System 3	0.8908	0.9220	0.9061	0.9061	0.8703	0.7104	0.8670	0.9144
tLink								
System 1	0.5883	0.7184	0.6355	0.6468				
System 2	0.6589	0.7129	0.6767	**0.6849**				
System 3	0.6434	0.7027	0.6626	0.6718				0.6932
End-to-end tLink								
System1	0.5904	0.5944	0.5921	**0.5924**				
Our best	0.6710	0.6001	0.6245	**0.6336**				0.6278

Bold indicates the highest values.

RESULTS

Performance was measured by the standard measures from the i2b2 organizers. The evaluation measures for event extraction are precision (P), recall (R), average P&R (P&R), F measure (F), and the accuracy of type, polarity, and modality. The evaluation measures for timex extraction are also P, R, P&R, F, and the accuracy of type, value, and modifier. In temporal relation, the evaluation method is the TempEval method[10] and the span match method is overlap. The evaluation measures for temporal relation are P, R, P&R, and F.

We submitted one system for event extraction, three systems for timex extraction, three systems for tLink, and an end-to-end system (see table 1). Due to limited development time, we only submitted an end-to-end system for tLink. After the challenge, we ran another end-to-end system using System1 (Event Extraction)+System1 (Timex Extraction)+System2 (tLink) (our best; see table 2). For timex extraction, our best system, system 1, yielded 0.9144 (F), 0.8814 (P), 0.9500 (R), 0.8929 (type accuracy), 0.7159 (value accuracy), and 0.8918 (modifier accuracy). For temporal relation, our best system, system 2, yielded 0.6849 (F), 0.6589 (P), 0.7129 (R), and 0.6767 (P&R) (see table 2). The experiment result shows that the features of syntax from a parse tree, LSP, and the result of a coordination classifier are discriminative in temporal relation identification of same sentences. In the 2012 challenge , with the F measure used for evaluation, our systems achieved first place (event track), first place (timex track), third place (tLink track), and second place (end-to-end tLink track).

In addition, three experiments were conducted using cross-validation on the training set. The first experiment is a comparison between the raw training set and transitive closure in discharge summaries. Table 3 demonstrates that the performance of using transitive closure is better than that of using the raw training set. The second experiment is a comparison between all pairs and separated pairs in <concept, section time>. Table 4 shows that the <concept, section time> framework makes a major contribution shown by a 0.3% improvement. The third experiment is a comparison between 10 solvers and one solver. The F measure of one solver is 0.010 compared to the F measure of 0.6849 using 10 solvers.

DISCUSSION

Timex extraction

The size of the training set is limited for timex extraction (2390 timexes). Many patterns of time expressions have very few occurrences in the training set, and hence they cannot be learned correctly (eg, '97). Due to limited development time, the rule set is incomplete and may miss some expressions in the test set.

Another challenge is to testify whether a digital expression, which is in a correct date format, is actually a date expression, such as '12/8/5' in the sentence 'Patient was eventually placed on BiPAP with BiPAP settings of *12/8/5* liters of oxygen.' Most such cases can be solved using some simple rules, but there are still a few cases that cannot be correctly handled.

There are still some time expressions with ambiguous meanings that are not handled in the evaluation system, like 'POD #1,' which can mean either the operation day or the day after the operation day. In our normalization system, we used the number of conflicts among deduction rules to distinguish different meanings of the same time expression. However, there are some documents for which this method does not work, either because the clues are very hard to find or there are no clues at all.

For some other cases, computing correct values for time expressions requires understanding of the whole sentence. For instance, in the sentence 'given (drug name) p.o. on *Monday, Wednesday, and Friday*,' the expression 'Monday' no longer means 'every Monday' instead of a specific Monday. Hence, the expression 'Monday, Wednesday, and Friday' should be annotated together as 'RP2d,' because it means 'three times a week.'

Table 3 Micro-averaged results for a comparison between the raw training set and transitive closure in discharge summaries

Same sentences	Raw data F measure	Transitive closure F measure	Increased	Different sentences	Raw data F measure	Transitive closure F measure	Increased
<event, event>	0.6095	0.7269	0.1174	<event, event>	0.3816	0.4546	0.0730
<event, timex>	0.6408	0.6801	0.0393	<event, timex>	0.4803	0.5326	0.0523
<timex, timex>	0.3812	0.4625	0.0813	<timex, timex>	0.3212	0.3854	0.0642

Table 4 Micro-averaged results for a comparison between all pairs and separated pairs in <concept, section time>

Micro F measure	Separated pairs	All pairs
Admission		
<event, admission>	0.8221	
<timex, admission>	0.9576	
Discharge		
<event, discharge>	0.9502	
<time, discharge>	0.9876	
Micro F measure	0.9102	0.8810

Our current normalization system cannot handle these issues because it requires understanding of the whole sentence.

Some timex values rely on the time points of the sentences they belong to. In our normalization system, we used the value of the nearest date or time expression as the time point of the sentence. This is only a rough estimation and fails in many cases, such as when there are implicit expressions indicating time changes between the two expressions. For instance, consider the following sentences: '… on postoperative day two, … carried on a 2-day treatment, … at this time ….' Since '2-day' is a duration expression or not labeled at all because it may be located in a treatment concept, the anchor time point of 'that time' will be set to 'postoperative day two,' while the actual time point is postoperative day four.

The length of some duration expression relies on other time points. Consider the following two sentences: 'She has been on total parenteral nutrition and has been nothing by mouth *since then*.' and 'The patient did well in the *postoperative period*.' In both sentences, the values of duration expressions 'since then' and 'postoperative period' rely on two other anchoring time points: for the expression 'since then,' we need to know the time point of the current sentence and the time point anchored to 'then.' For the expression 'postoperative period,' we need to know the operation day and the time point of the current sentence. Our current normalization cannot handle such problems and will only give a rough estimation for these expressions.

Temporal relation

Relations in the same sentences

First, some errors arise from inconsistency of annotation. For example, in sentence 'Pathology was negative for tumor and **showed** *peritubal and periovarian adhesions*' the relation between 'showed' and the disease '*peritubal and periovarian adhesions*' is overlap, while in the sentence 'She has been worked up with barium enema in 09/97 which **showed** *multiple diverticula* throughout the colon, but mostly in the sigmoid,' the relation between 'showed' and the disease '*multiple diverticula*' should be after. Second, another problem is that the training set is sparse. For example, in 'She initially awoke after **surgery**, with **good hemodynamics**' and 'She initially awoke with **good hemodynamics** after **surgery**,' 'surgery' appears in both sentences denoting different time points.

Relations in the different sentences

As we are only able to identify the relations of concepts between adjacent sentences, pairs in non-adjacent sentences are missed, although temporal relations may exist in them. Another difficulty is that in two adjacent sentences, some pairs are mislabeled as none in the training set. For example, in sentences 'She described **the pain** as **a burning pain** which is positional, **worse** when she **walks** or does any type of **exercise**' and 'She has no **relief** from **antacids** or **H2 blockers**,' <the pain, relief> is labeled as 'before' but <a burning pain, relief> is labeled as 'none' even though <pain, a burning pain> is labeled as 'overlap' in the training set.

Relations between concepts and section time

We divided each discharge summary into an admission section and a discharge section, and matched only 'admission date' with concepts in the admission section and 'discharge date' with concepts in the discharge section. Thus, some relations are missed in our system since there are temporal relations linking the admission date to concepts in the discharge section and vice versa, as labeled in the training set.

Inference using MLN

Performance was degraded a bit when we used an MLN to infer tLinks. One possible reason is that we used the confidences of SVM as features in MLN inference. As a result, there is not enough information to infer temporal relations.

CONCLUSION

This paper describes an end-to-end system for identifying temporal relation in discharge summaries.

For timex extraction, we used contextual information to train a CRF model, which disambiguates expressions that are not time expressions. For normalization, the free-context grammar method which is flexible and can be applied to other domains, was utilized in order to avoid the use of regular expressions.

For relation identification, according to different relation types, we used 10 multi-SVM classifiers in order to make the features in each classifier homogeneous. The framework has the following advantages: (1) syntax features based on parse trees with global information improve the performance of temporal relation identification; (2) the use of LSP improves both the recall and precision; (3) a classifier of coordination was also introduced to correct the errors made by parsers; (4) only concepts in neighboring sentences were analyzed, which reduced errors caused by the sparseness of features, and (5) the use of section recognition. The above structure gives a satisfying result which demonstrates the success of our framework.

For event extraction, we used the gold standard from the i2b2 challenge in 2010 and 2011. However, since some event types (clinical_dept, evidential, and occurrence) were not included in the previous challenges, our future work is to further investigate these three types.

Acknowledgements We would like to thank the organizers of the i2b2 NLP challenge.

Contributors YX, YW, and TL wrote the code. All authors made valuable contributions to studying design and writing the article.

Funding This work was supported by Microsoft Research Asia (MSR Asia). The work was also supported by MSRA eHealth grant, Grant 61073077 from National Science Foundation of China and Grant SKLSDE-2011ZX-13 from State Key Laboratory of Software Development Environment in Beihang University in China. We would like to thank the organizers of the i2b2 NLP challenge. This project was supported by Informatics for Integrating Biology and the Bedside (i2b2) award number 2U54LM008748 from the NIH/National Library of Medicine (NLM), by the National Heart, Lung and Blood Institute (NHLBI), and by award number 1R13LM01141101 from the NIH NLM.

Competing interests None.

Provenance and peer review Not commissioned; externally peer reviewed.

REFERENCES

1 Sun W, Rumshisky A, Uzuner O, Evaluating Temporal Relations in Clinical Text: 2012 i2b2 Challenge Overview. *J Am Med Inform Assoc* 2013;20:806–13.
2 Chambers N, Wang S, Jurafsky D. Classifying temporal relations between events. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007). Prague, Czech Republic: Association for Computational Linguistics, 2007:173–6.
3 Chambers N, Jurafsky D. Jointly combining implicit constraints improves temporal ordering. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008). Waikiki, Honolulu, Hawaii: Association for Computational Linguistics, 2008:698–706.
4 Yoshikawa K, Riedel S, Asahara M, *et al*. Jointly identifying temporal relations with markov logic. Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJNLP 2009). Singapore: Association for Computational Linguistics, 2009:405–13.
5 Zhao R, Do QX, Roth D. A robust shallow temporal reasoning system. Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2012). Montreal: Association for Computational Linguistics, 2012:29–32.
6 Garrido G, del Rosal J. Pattern Learning for Relation Extraction with a Hierarchical Topic Model. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012). Jeju, Korea: Association for Computational Linguistics, 2012:54–9.
7 Bunescu RC, Mooney RJ. A shortest path dependency kernel for relation extraction. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMLP 2005). Austin, USA: Association for Computational Linguistics, 2005:724–31.
8 Mani I, Verhagen M, Wellner B, *et al*. Machine learning of temporal relations. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL 2006). Sydney, Australia: Association for Computational Linguistics, 2006:753–60.
9 Tempeval 2007. http://labs.thomsonreuters.com/files/studiessurveys/tempeval.pdf (accessed 2012).
10 Tempeval 2010. http://timeml.org/tempeval2/ (accessed 2012).
11 Strotgen J, Gertz M. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. Proceedings of the 5th International Workshop on Semantic Evaluation. Uppsala, Swden: Association for Computational Linguistics, 2010:321–4.
12 UzZaman N, Allen JF. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. Proceedings of the 5th International Workshop on Semantic Evaluation. Uppsala, Swden: Association for Computational Linguistics, 2010:276–83.
13 Raghavan P, Fosler-Lussier E, Lai AM. Learning to temporally order medical events in clinical text. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012). Jeju, Korea: Association for Computational Linguistics, 2012:70–4.
14 Denny JC, Peterson JF, Choma NN, *et al*. Extracting timing and status descriptors for colonoscopy testing from electronic medical records. *J Am Med Inform Assoc* 2010;17:383–8.
15 Zhou L, Friedman C, Parsons S, *et al*. System architecture for temporal information extraction, representation and reasoning in clinical narrative reports. Proc AMIA Symp 2005. Washington, DC, USA, 2005:869.
16 Jung H, Allen J, Blaylock N, *et al*. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011). Portland, Oregon, USA, 2011: 146–54.
17 CRF++. http://crfpp.googlecode.com/svn/trunk/doc/index.html (accessed 2012).
18 multi-SVM. http://svmlight.joachims.org/svm_multiclass.html (accessed 2012).
19 markov logic networks. http://code.google.com/p/thebeast/ (accessed 2012).
20 Xu Y, Hong K, Tsujii J, *et al*. Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. *J Am Med Inform Assoc* 2012;19:824–32.
21 Enju parser. http://www.nactem.ac.uk/enju/demo.html (accessed 2012).
22 Stanford parser. http://nlp.stanford.edu:8080/parser/ (accessed 2012).
23 Agrawal R, Srikant R. Mining sequential patterns. Proceedings of the 11th International Conference on Data Engineering (ICDE 1995). Taipei, Taiwan: International Conference on Data Engineering, 1995:3–14.
24 Cong G, Wang L, Lin CY, *et al*. Finding question-answer pairs from online forums. Proceedings of the 31st ACM Special Interest Group on Information Retrieval (SIGIR 2008). Singapore: ACM Special Interest Group on Information Retrieval, 2008:467–74.
25 Sun G, Cong G, Liu X, *et al*. Mining sequential patterns and tree patterns to detect erroneous sentences. Proceedings of the 22nd Conference on Artificial Intelligence (AAAI 2007). Vancouver, British Columbia, Canada: Association for the Advancement of Artificial Intelligence, 2007:925–30.
26 Denny JC, Spickard A III, Johnson KB, *et al*. Evaluation of a method to identify and categorize section headers in clinical documents. *J Am Med Inform Assoc* 2009;16:806–15.