Published in final edited form as:
J Biomed Inform. 2013 December ; 46(0): S40–S47. doi:10.1016/j.jbi.2013.11.001.

Towards generating a patient's timeline: Extracting temporal relationships from clinical notes

Azadeh Nikfarjam, **Ehsan Emadzadeh**, and **Graciela Gonzalez**
Department of Biomedical Informatics, Arizona State University, Tempe, USA

Azadeh Nikfarjam: anikfarj@asu.edu

Abstract

Clinical records include both coded and free-text fields that interact to reflect complicated patient stories. The information often covers not only the present medical condition and events experienced by the patient, but also refers to relevant events in the past (such as signs, symptoms, tests or treatments). In order to automatically construct a timeline of these events, we first need to extract the temporal relations between pairs of events or time expressions presented in the clinical notes. We designed separate extraction components for different types of temporal relations, utilizing a novel hybrid system that combines machine learning with a graph-based inference mechanism to extract the temporal links. The temporal graph is a directed graph based on parse tree dependencies of the simplified sentences and frequent pattern clues. We generalized the sentences in order to discover patterns that, given the complexities of natural language, might not be directly discoverable in the original sentences. The proposed hybrid system performance reached an F-measure of 0.63, with precision at 0.76 and recall at 0.54 on the 2012 i2b2 Natural Language Processing corpus for the temporal relation (TLink) extraction task, achieving the highest precision and third highest f-measure among participating teams in the TLink track.

Keywords

Temporal relation extraction; Clinical text mining; Automatic patient timeline; Natural Language Processing; Machine learning; Temporal graph

1. Introduction

The narrative sections of clinical records contain information about clinically relevant events happened to patients. Most of the events, such as the patient's illness progression, test results or the effect of a treatment are only meaningful in a specific timeline [1]. Questions such as “*How effective was the treatment?*” can only be answered and interpreted if the relative temporal relations between the events are considered. In general, temporal reasoning has applications in several tasks in the clinical domain such as information extraction [2,3], question answering [4,5], patient timeline visualization [6], clinical guideline development [7,8] and others. Automatic extraction of temporal information can facilitate processing of patient information in the narrative text, and this can contribute to the decision making process in fundamental patient care tasks such as prevention, diagnosis and forecasting the effects of the treatments [9,10]. Consider, for example, a situation where a patient with history of depression is brought to the emergency room. In order to make an informed decision about the gravity of the situation, the physician would need to go over previous

Correspondence to: Azadeh Nikfarjam, anikfarj@asu.edu.

visits and manually find and sort important events to determine suicidal intent. Alternatively, consider a child presented to the ER with trauma and possible fractures, reported as caused by a fall by the family member bringing her in. The attending physician has to quickly determine whether the injuries could be instead the result of abuse and flag the record for social services intervention, but key information can be hard to find on the spot. Automatic and reliable timeline generation in such cases and others like them can facilitate the decision-making process and potentially reduce medical errors. For a comprehensive review of the applications of automatic temporal reasoning in the clinical domain, we refer to Zhou et al.'s survey on the subject [9].

There are diverse and complex linguistic mechanisms for representing the temporal information in natural language that make it very challenging for Natural Language Processing (NLP) systems to extract such information. For example, in many of the temporal event descriptions, the associated time is not explicitly mentioned. Automatic extraction of such implicit information requires domain knowledge plus the utilization of sophisticated NLP techniques. Sun et al. [1] provide a thorough discussion about the challenges of automatic temporal reasoning from clinical text. To promote advances in this area, the Sixth Informatics for Integrating Biology and the Bedside (i2b2) Natural Language Processing Challenge for Clinical Records focused on temporal reasoning in clinical narratives [11]. The challenge included three tracks: Event/Timex, TLink and End-to-End. Here we focus on the TLink track and briefly explain about the provided 2012 i2b2 annotated corpus. For more information about the corpus and other tracks in the challenge, please refer to Sun et al.'s paper [11].

In the TLink track, participants were asked to develop systems to extract temporal relations (TLinks) of three types (*before*, *after* and *overlap*) between the given events and temporal expressions in the narrative portion of clinical records. The provided training data includes 310 de-identified discharge summaries in which the clinical events and temporal expressions are annotated. A total number of 190 notes were released for training and 120 notes were later released for testing. There are three different types of annotations in each note: events, temporal expressions (timex), and temporal relations (*TLinks*) between events and timex mentions. Events are clinically relevant situations such as treatments, problems, tests and other occurrences. Temporal expressions are mentions of date, time, duration and frequencies. The method that we present in this paper, a hybrid approach that features a combination of machine learning and graph-based inference, was developed for the TLink track. We propose an innovative way of utilizing parse dependencies for temporal information extraction by first simplifying a sentence and then generating a temporal graph based on the simplified version of the sentence. We introduce a method for generalizing sentences and extracting hidden frequent patterns, which was applied in the creation of temporal graphs. Moreover, novel classifier features are introduced to characterize the TLinks. The features significantly contributed to achieving high performance on the test data compared to other systems in the TLink track.

2. Related work

Extraction of temporal relations from non-clinical text has attracted a lot of attention in the text mining community. The two TempEval competitions [12,13] were designed for the goal of temporal information extraction, and greatly helped to advance the field. In these competitions, machine learning (ML) approaches [14,15] were more successful than the rule-based methods [13]. CU-TMP [16] is an SVM-based system for the temporal extraction problem and was the best performing system in TempEval 2007 [13]. Other ML-based systems presented there, utilize Markov logic networks [15] [17] and Conditional Random Field classifiers [18,19]. After comparing the performance of various classifiers, Min et al.

[20] report that SVM is the best performing ML method for temporal relation extraction. Learning from these non-clinical best performing systems, we used SVM as the ML classifier in our work.

Clinical text presents additional challenges for the extraction of temporal relations, and work in this area was recently motivated by the 2012 i2b2 task that focused on this problem. Eighteen teams from around the world participated in different tracks of the challenge, and utilized different ML, rule-based, or hybrid approaches [11]. We briefly outline the approaches taken by the top 2 teams in the TLink track. Cherry et al. [21], who attained the best f-measure (0.69), show that using an ensemble system, consisting of four components and targeting different possible TLinks in the notes, can successfully extract the TLinks. Tang et al. [22] submitted the next top ranked system, using heuristic rules to define the candidate link pairs instead of generating all the possible candidate links. They apply CRF++ and SVM for classification of the TLink candidates. In the official submission, our system achieved a precision of 0.76, a recall of 0.54 and F-measure of 0.63, placing 1st in precision and 3rd in F-measure among the competing systems in the TLink track. One of our unique contributions is the idea of temporal graph (Section 3.3) built based on frequent patterns and rules. Our method calls first for simplifying the sentences (Section 3.2), then parsing them and calculating novel grammatical features (Section 3.4.2). Our approach results in a high-precision system that can subsequently be refined to improve its recall.

3. Methods

We approached the problem of finding TLinks with three complementary methods: graph reasoning (Section 3.3), machine learning (SVM) (Section 3.4), and rule-based classification (Section 3.5). The overall approach is illustrated in Fig. 1. We first generated all possible TLinks and calculated the features that characterize them in the “TLink Candidate Builder” module (Section 3.1). The TLinks were divided into three categories: section time-event (sectime-event), within-sentence and between-sentence links. For each type of TLink, a different classification pipeline was used. Examples of the different types of TLinks are present in the following sentences:

i. The patient's chest tubes were removed on postoperative day Three.

ii. The patient was started on low dose Lasix which he tolerated well.

The TLink between “*The patient's chest tubes*” (event) and “*postoperative day three*” (timex) is an example of a within-sentence link with the related type “*before*”. The TLink that connects “*low dose Lasix*” (event) and “*postoperative day three*” is a between-sentence TLink with the link type “*after*”.

We trained an SVM classifier for the classification of the sectime-event candidates. The within-sentence candidates were first passed to the Temporal Graph Reasoning module. If the type of a candidate could not be determined, it was passed into the within-sentence SVM Classification module. Between-sentence candidates were processed solely with a set of heuristic rules (Section 3.5).

3.1. TLink Candidate Builder

The “TLink Candidate Builder” module created the possible TLink candidates (between-sentence, sectime-event and within-sentence TLinks) in a given clinical note. The later two candidates were used for training/testing the SVM classifiers or fed into the graph reasoning module for deciding about the link types. The candidates were categorized into two types:

- *Sectime-event*. Each clinical note has associated admission and discharge time, which are referred to as *section times*. Every event in the note can be *before, after* or *overlap* with either "admission" or "discharge" time; the choice of admission or discharge depends on the location of the event in the note. Each note in the corpus includes two main sections: *patient history* and *hospital course*. To comply with the guideline, we compared the time of the events presented in the *patient history* section with "admission" time, and the events in the *hospital course* section with "discharge" time. We created a candidate TLink connecting every event to its associated section time. For instance, the first example sentence is located in the *hospital course* section; therefore "The patient's chest tubes" and "postoperative day three" are both compared with "discharge" time and are *before* discharge.

- *Within-sentence*. For each sentence, we built a complete graph. The nodes were the events and time expressions in the sentence, and the edges were the link types (*before, after, overlap* and *unknown*). We added *unknown* as the fourth and the default type to consider all the possible temporal relations in a sentence. Other link types were set based on the corresponding annotated TLink in the training data. There were two different types of within-sentence TLinks:
- *Timex-event*. timex-events were the links that connected a timex node to all of the existing event nodes in the sentence. We generated a candidate TLink for each timex-event edge in the graph.
- *Event-event*. Similarly, for each possible link between every two events in a sentence, a corresponding candidate event-event TLink was created.

After generating the TLink candidates, a set of features was calculated for every candidate before passing them to the graph inference or the SVM module for classification. See Section 3.4.2 for feature details. We used simplified versions of the sentences in the corpus, in generating some of the classifier features, and in building the temporal graph. The sentence simplification approach is described in the following section.

3.2. Sentence simplification

Many of the events and time expressions in the sentences were expressed as descriptive phrases, such as "a video assisted thoracoscopic study of the right lung," which has 9 tokens as a single annotated event. The sentence simplification idea was motivated by observing that grammatical dependency relations provided useful information in finding the temporal relations between the mentions. However we did not need the whole content of the mentions to establish the dependencies, and the extra content could actually hinder the parser's determination. Therefore, we decided to first simplify the sentences and then parse them to get the dependency relations. By simplifying the sentences, we excluded the uninformative words (with regard to the syntax of the sentence) in each mention to make it simpler for analysis. As an example, consider the following sentence with the pre-tagged events in bold: "***The MRI scan*** *on admission revealed* ***an impending cord compression*** *at the level of T10*". This was simplified to "***Scan*** *on admission revealed* ***compression*** *at the level of T10*." Thus, to simplify a sentence, we replaced each identified event or timex with only one representative word of the event. The representative word of an event was simply the head word of the phrase. Choosing the representative word of a temporal expression depended on the type of the timex. Date and time expressions were replaced with their absolute normalized value (included in the provided input data). For instance, "*the morning of the first day of admission, August 16, 1998*" was replaced with 1998-08-16. Other types of temporal expressions (such as *duration*) were replaced with the first noun in the phrase. If there was no noun in the phrase, the last word was used as the representative word (e.g. "*the following three days*" was replaced with "*days*").

3.3. Temporal graph

As stated before, a novel aspect of our work is a graph-based approach to find the temporal relations in a sentence (within-sentence TLinks) based on the possible path between pairs of events and time expressions in the sentence. A temporal graph is a directed graph where the nodes (vertices) are a subset of the words in the sentence, and the edges are labeled with possible temporal relations. We generated a corresponding temporal graph for each candidate sentence. The possible link type between pairs of nodes was identified based on the calculated temporal signal. Fig. 2 presents a sample sentence with the corresponding temporal graph. Consider the two target events “*scan*” and “*compression*”. There is a path with {*overlap*, *after*} as the set of edge labels, illustrated with bolder arrows. Based on a set of rules (Section 3.3.2), and considering the label set in the path, we concluded that scan occurred “*after*” *compression*. Graph building details and reasoning are presented in the following sections.

3.3.1. Generating the temporal graph—Building the graph started with adding a node corresponding to every event or timex in the sentence. Next, we added the edges based on a set of temporal signals using two main approaches: pattern driven and rule driven. Considering every pair of nodes, if the activated signal indicates *before*/*after*/*overlap*, an edge with the corresponding label connected the first node to the second one and the reverse edge was added from the second to the first node (Fig. 3). Otherwise, no edge was added between the nodes.

3.3.1.1. Pattern driven signal type detection: A group of the training TLinks follows recurring patterns of words and POS between oraround the arguments of the link such as “**occurrence** on **date**” or “**date** prior to **occurrence**”. We built a simple conditional model over the observed pattern of tokens presented in the link. First, we generalized the sentence and then extracted the tokens located between the pairs of the link's arguments. Examples of the generalized sentences are presented in Table 1.

A between-token pattern of a given TLink in the test data could appear zero to many times in the training data. If we had never seen the pattern in the training data, no decision was made using this approach. However, in many cases, the pattern was observed in different links and sometimes with different link types. For assigning the signal type for a given test instance, with the observed pattern (p), we chose the link type that maximized the conditional probability in Eq. (1) – where x and y are TLink instances and L is the set of all TLinks in the training data. *Pattern*(x) is the between-mention pattern of x and *LinkType*(x) is the type of the TLink x. We applied two constraints to choose the link type: the maximum probability should be higher than the defined threshold parameter, α (α > 0.5), and the total number of patterns in the training data should be more than β instances, where β is set to be 2 in our experiments.

$$LinkType = \text{argmax}_t \frac{|\{x|x \in L, Pattern(x)=p, LinkType(x)=t\}|}{|\{y|y \in L, Pattern(y)=p\}|} \quad (1)$$

Eq. 1: The link type(t) that maximizes the conditional probability is the selected linkType

There are different approaches whereby a sentence can be generalized; following a similar approach to our previous proposed method [23], we replaced every mention in the sentence with the corresponding type such as *treatment*, *frequency* or *duration*. Other words (except verbs) were replaced with the related part of speech, while verbs remained intact.

3.3.1.2. Rule driven signal type detection: In this module, we first parsed the simplified sentences to get the dependency relations. A dependency relation is a triplet that shows the grammatical relationships between two words in a sentence, and is usually presented as "*Relation* (w_i,w_j)." *Relation* is the name of the dependency and w_i and w_j are referred to as the *governor* and the *dependent* words of the relation [24]. In the next step, for every dependency, a set of rules was checked and if satisfied, the corresponding temporal signal was used as the label of the edge that connected the governor and the dependent words. If the governoror the dependent were not in the initial set of graph nodes, we added the nodes and then connected them with the corresponding edge. The possible edge labels were *before, after* and *overlap*. Prepositions in a sentence play an important role in conveying the temporal signal type. We manually selected a subset of the Stanford dependency relations [24] for every link. For example, "*before*" signal was activated between the governor and the dependent nodes if the dependency relation belonged to *prep_prior_to*, *prep_until* or *prep_towards*. The "*after*" signal was activated if the relation name was *prep_after*. The "*overlap*" signal was activated by the relations such as *prep_at* or *prep_during*.

3.3.2. Inference based on temporal graph—

We identified the link type of a candidate TLink based on the path between the arguments of the TLink candidate in the temporal graph. The underlying assumption was that if there was a path between the two nodes, it was likely that there was a temporal relation between them; otherwise the link type could not be identified by the graph reasoning module. We used the following rules based on the edge labels in the path to assign one of the possible link types:

- *Overlap:* if an "*overlap*" edge was in the path and there was no edge with "*before*" or "*after*" label.
- *Before:* if a "*before*" edge was in the path and there was no edge with "*after*" label in the path.
- *After:* if an "*after*" edge was in the path and there was no edge with "*before*" label in the path.

If the graph inference module determined the type of the TLink, that type would be the final decision for the given candidate and the predicted class was then updated in the database. Conversely, if the graph inference module could not decide about the type of the link, the candidate was passed to the SVM classifier for the final decision.

3.4. TLink SVM classifiers

We used SVM classifiers for sectime-event and within-sentence TLink candidates. We trained an SVM model for sectime-event, and two separate SVM models for event-event and timex-event candidates. SVM was selected since it was shown to be effective in similar temporal link extraction tasks [16, 20]. There were four possible final classes (*before, after, overlap and unknown*) that every TLink candidate could be assigned by the classifier, therefore we utilized SVM$^{\text{Multiclass}}$ implementation [25] of the algorithm.

3.4.1. Expanding the within-sentence TLinks—

We evaluated the effectiveness of training the within-sentence SVM classifiers on the expanded set of original TLinks. Consider the three events (A, B and C): if A is before B and B is before C, then we can infer that A is before C. In most cases, there was no explicitly annotated TLink in the training data that connected A to C. As we explained in Section 3.1, we generated TLink candidates for every pair of mentions in a sentence and assigned the default type to "*unknown*". Therefore, the type of the TLink connecting event A to C initially was set to "*unknown*" while the inferred type was "*before*". By considering the transitivity of the temporal relations, we increased the number of training instances that had a link type other than *unknown*. We expanded the original TLinks based on the transitive rules in Table 2. We found that training the classifiers on the expanded set increased the recall of the system with the cost of having a decrease in the precision (see the results in Section 4).

3.4.2. TLink classification features—

We calculated the same set of features for event-event and timex-event candidates. The following list of features effectively contributed to achieving the highest precision among all the systems submitted to the TLink task.

3.4.2.1. TLink's arguments basic features: These features were mainly attributes of the events and temporal expressions which were provided as part of the annotations in the input i2b2 corpus. They were used for both of the participating mentions (timex or event) in a TLink. The basic features included: textual content, mention type (whether it was an event or a timex), event type (such as problem, test, treatment and others), event modality (e.g. factual, hypothetical), event polarity (negated or not), timextype (date, time, frequency or duration), timex modality (e.g. approximate).

3.4.2.2. TLinks' lexical features: These features were related to the ink arguments and the words between them. This includes the number of words in between, number of events and time expressions in between, and bigrams of the words that were between the TLink arguments. Bigrams were binary features that turned true based on the presence of the two consecutive words located n between the TLink arguments.

3.4.2.3. Dependency-based features: These were the syntactic features calculated from the parse dependency relations and part of speech (POS) tags of the TLink's sentence. In order to use dependencies as classifier features, they are usually transformed to the corresponding string "Relation-w_i-w_j". This way of representation s referred to as lexicalized dependency [26]. We used the Stanford parser [27] to parse a sentence and calculated the following features accordingly. Some of these features, such as POS, preposition, and related verb, have been previously shown to be effective in temporal link extraction [16].

- *Mention POS* was the part of speech of the link arguments. If a mention included more than one word then the sequence of the part of speeches were used as the value of this feature (e.g. ADJ-NN).
- *Related preposition* was the preposition (such as *for*, *on*, *at*) related to the mention, for instance "*at*" in "*at the hospital*".
- *Related verb* was the governor verb of the TLink arguments. Among Stanford dependencies, there are some dependencies that represent the relation of a verb with subject (*nsubj*, *nsubj-pass*), object (*dobj*) or complement (*cop*) of a sentence; in such relations, the governor word is the governor verb of the dependent. For more information about the dependency relations, please refer to Marneffe and Manning [24]. If we could not find any of the verb related dependency relations among the sentence's dependencies, we chose the nearest preceding verb to the mention as the related verb.
- *Verb auxiliaries* were the auxiliaries of the related verbs such as can, could and may.
- *Are verbs connected?* This was a binary feature that showed if the related verbs of the link arguments were connected in the dependency graph or not.
- *Lexicalized dependencies of the simplified sentence*. We included all the lexicalized dependencies of the simplified sentences. For instance, a subset of the lexicalized

features for the example sentence in Section 3.2 includes: nsubj-revealed-scan, prep_on-scan-admission, dobj-revealed-compression.

- *Are arguments directly connected?* This was a binary feature that turned true if the TLink arguments had a direct relation among the dependency relations of the simplified sentence.
- *Have common governors?* This feature was also a binary feature showing that whether the two TLink's arguments had a common governor in their dependency relations of the simplified sentence.

Note that all of the above features were used in training both event-event and timex-event candidates. Features used in sec-time-event SVM include: TLink's arguments basic features; the location of the event (“hospital course” or “history”) and the type of the target section time (admission or discharge).

3.5. Rule engine

For finding links between concepts in two different sentences, one approach is to create all possible links between mentions in the neighboring sentences and run an SVM classifier on them. This approach turned out not to be effective, since the number of negative instances became very large. To overcome this problem, a set of limited heuristic rules was used to create and label TLinks based on certain observations in the training data. The rules performed better than an SVM classifier, running over all possible links between neighbor mentions in different sentences. We defined the following rules for classifying between-sentence links (the first sentence denoted as s_1 and the second sentence denoted as s_2):

1. Create “overlap” link if s_2 has only one mention m_2:

   i. m_2 is TEST and m_1 is the first occurrence of *treatment*, *clinical_dept* or *test* before m_2.

   ii. m_2 is “*duration*” and m_1 is the first occurrence of “*treatment*” before the m_2.

   For example, consider the two following sentences: “ALT was 53.” and “AST was 89;” the assigned link type between “ALT” and “AST” is “overlap”.

2. Create “before” link if an event is repeating. The repetition is detected by “repeat” trigger word. For example: “White cell count” and “Repeat white cell count”.

These two simple rules detected some between-sentence TLinks and slightly helped to improve the overall recall. As we mentioned before, we did not focus on proposing a complete solution for between-sentence TLinks and further research is needed to solve this problem.

4. Results

We measured the performance of the system by evaluating it with the test data (120 notes) using the evaluation script provided by the i2b2 challenge organizers. It measured the overall performance of the system (considering all the expected TLinks in the discharge note). Precision, recall and F-measure were used as the evaluation metrics.

Table 3 shows the individual evaluation of the different modules in the system after the modifications presented here, compared against our submitted system performance (Original TLinks), listed in Table 4. Using the ground truth of the test data, we assigned the gold standard link types to the TLink candidates and measured the maximum possible obtained recall that each individual module (sectime-event, within-sentence (WS) and between-sentence) could achieve. We found that around 30% of the TLinks were between events and section times, 43% were within sentence, and 27% were between sentence links (listed as “max possible recall” in Table 3). The sectime-event module that used SVM to classify the TLinks, successfully extracted the majority of the sectime-event candidates and classified them with the precision of 0.92. The within-sentence module, trained on the expanded set of TLinks (Section 3.4.1), was the strongest module in our system, achieving a recall of 0.34 and a precision of 0.60 in extracting the within-sentence TLinks. In evaluating the within-sentence module, the achieved recall (using gold standard values) was 0.43, while our system's recall was 0.34. The graph inference method for the within-sentence module extracted a relatively low number of TLinks with high precision. When we used the hybrid method (WS (Hybrid)), we got a slight rise in the F-measure. Between-sentence (BS) rule-based method covered a small portion of the between-sentence TLinks and, as expected, contributed little to the overall performance.

Table 4 shows a comparison of the overall performance of the system when the within-sentence classifier is trained on the original TLinks versus when it is trained on the expanded set of TLinks (as explained in Section 3.4.1). When the system was trained on the expanded set, we got a noticeable rise in the recall. However, the precision dropped from 0.76 to 0.71, resulting in a 2% increase in the overall F-measure.

5. Discussion

We found that the combination of graph inference and ML-based classification is an effective approach for extracting temporal links from clinical notes. Temporal reasoning over clinical data is a challenging task for even human annotators, as demonstrated by an inter-annotator agreement for the TLink track of 0.79 [11]. This means that the best automated approaches, when trained on this corpus, are expected to eventually achieve a performance no higher than the reported agreement level.

As Table 4 shows, training the within-sentence SVMs on the expanded set of the original TLinks (Section 3.4.1) significantly increased the recall of the system. However, the precision decreased to 0.71 from 0.76. The expanded set of TLinks included more instances, and their arguments were located farther apart in a sentence. They also added more variety to the classifier features, but did not generalize as well as the original TLinks. Yet, the overall F-measure improved by expanding the training data.

Table 3 shows that sectime-event module alone reached a precision of 0.92 and a recall of 0.25. Considering that sectime-event TLinks constitute 30% of the total TLinks, this module successfully extracted 83% of the possible sectime-event TLinks with high precision. The errors of this module were mainly related to the candidate builder component (Section 3.1). If the candidate builder component could not find the “*hospital course*”, “*patient history*” or the section times (“*admission*” and “*discharge*”), it did not generate the correct sectime-event candidate.

When we only evaluated the within-sentence SVM classifiers and measured the overall performance, the system reached a recall of 0.33 and a precision of 0.60 (Table 3); while when using the graph inference the system reaches a recall of 0.14 and a precision of 0.70. However, when we combined the classifier and the temporal graph inference, the recall did not get as large an increase as one would have expected, reaching only 0.34. One possible explanation is that when we use the graph inference and SVM individually, many of the correctly classified TLinks are common to both.

In general, the proposed approach to generating the temporal graphs has inherent limitations. One of the limitations is that currently generating the graph is partially dependent on a limited number of manual rules as explained in Section 3.3.1.2. Investigating the effectiveness of automatically generating the temporal graph is an interesting future research direction that could address this limitation and potentially impact the performance. On the other hand, pattern driven temporal signal detection (Section 3.3.1.1) for adding the edges in the graph highly depends on the size of the training data. We designed an experiment, in which we evaluated the within-sentence module when only pattern-driven signal type detection was applied. We measured the overall F-measure while using different number of training sentences (10% of the training sentences to 100%). As Fig. 4 illustrates, when only 10% of the training sentences was used, the recall was very low (0.04). As we increased the size of the training data, a smooth increase in the recall was observed. Therefore, using more training instances is expected to result in a higher F-measure, since recall is expected to continue increasing. The precision remained roughly at the level of 0.8 for different training size options, which was reflective of what was expected given the inter-annotator agreement. In the future, we plan to add more flexibility when generating the patterns, and apply semi-supervised pattern learning methods to be less dependent to the training data.

Additionally, we used a very limited set of rules to handle between-sentence TLinks, which contributed a little to the overall performance. However, around 27% of the links in the corpus were between-sentence links. More research is needed to design an ML classifier for this type of TLinks. The most challenging part in using ML classifiers for between-sentence links is keeping a balance between the number of positive (before, after or overlap) and negative (*unknown*) classification candidates by generating fewer negative candidates. Furthermore, using a co-reference resolution component in the system can help detect many of the overlap links that connect the references to the same event in different sentences.

We also performed an analysis to determine the source of the major errors in the within-sentence module. We randomly selected 50 false negative TLinks with expected types (*before, after or overlap*) that were wrongly classified as *unknown* by the system. As we showed in the result section, most of the within-sentence TLinks were classified by the SVM model. SVM tries to find an optimal hyper-plane to achieve the best overall result. This means that some similar instances, which are close to the hyper-plane, can be misclassified in order to get more instances correctly classified. For example, consider the following:

- The patient was briefly admitted to **the ICU** for low hematocrit and hyponatremia immediately following the surgery but was then **transferred** back to the floor in stable condition.

The system could correctly extract most of the TLinks in the sentence. Examples of the correct TLinks are: the link between "*low hematocrit*" and "*the ICU*" (*before*), "*hyponatremia*" and "*the ICU*" (*before*) and "*stable condition*" and "*the floor*" (before). It is interesting to note that, despite the relatively large distance between "*admitted*" and "*transferred*", the system could correctly classify the TLink into *before*. At the same time, the link type between "*the ICU*" and "*transferred*" was incorrectly predicted as *unknown*, while the correct link type is *before*. This shows that two very similar instances can be classified differently by SVM. Defining more distinguishing classifier features can reduce such errors.

Furthermore, for many of the misclassified TLinks, there were temporal trigger phrases in the sentence (such as *history of, continued, repeat, consecutive, subsequently*). If modeled properly, they could act as a very distinguishing feature and eliminate the misclassification. For example, the following TLink between the bold events was misclassified to *unknown*, but the word "*continued*" could trigger the right type, *overlap*:

- "He was given **D50**, but continued to have **progressive respiratory failure**, was …"

Similarly "*transformation*" could trigger *before* between "his *CMML*" (problem) and "*acute myelogenous leukemia*" (problem) in this sentence:

- "A bone marrow biopsy revealed the transformation of **his CMML** to **acute myelogenous leukemia**,…"

If the system had the knowledge that transformation of event A to B indicates event A happened before B, then it could predict the right link type. Incorporating similar knowledge in ML systems requires creating and incorporating a comprehensive ontology of all trigger words that is an ongoing research problem.

6. Conclusion

We proposed a system for extracting the temporal relations from clinical notes. The system utilized machine-learning and graph-based inference to extract the links between events and temporal expressions in the clinical notes. Specialized modules were designed for different types of temporal links: sectime-event, within-sentence and between-sentence. We found that using SVM classifiers in conjunction with temporal graph inference can produce promising results, in comparison with other systems, placing us among the top performing systems in the 2012 i2b2 TLink extraction challenge. The idea of sentence simplification and the use of the frequent patterns/parse dependency relations in creating the temporal graph can serve as a base for further studies on temporal relation extraction.

The sentence simplification method and the pattern-driven signal type detection approach can easily be applied for similar relation extraction tasks (such as drug-drug or gene-disease interaction extraction) from biomedical literature. The graph inference method and the proposed features are domain-independent and can be applied in other contexts.

Acknowledgments

The authors would like to thank the organizing committee members of the 2012 i2b2 Challenge, enabling the development of the methods presented. GG acknowledges the example cases presented in the introduction for which automatic timelines could be useful were inspired by discussions with Dr. Sarada Panchanathan, a pediatrician at Maricopa Integrated Health Systems and Clinical Faculty at ASU BMI. Proofreading by fellow lab member Rachel Ginn is greatly appreciated. The i2b2 NLP challenge was supported by Informatics for Integrating Biology and the Bedside (i2b2), grant number NIH NLM 2U54LM008748, and Challenges in Natural Language Processing for Clinical Narratives, grant number NIH NLM 1R13LM011411-01. This work was partially supported by NIH National Library of Medicine under grant number NIH NLM 1R01LM011176. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NLM or NIH.

References

1. Sun W, Rumshisky A, Uzuner O. Temporal reasoning over clinical text: the state of the art. J Am Med Inform Assoc: JAMIA. 2013 May.:1–6.
2. Denny JC, Peterson JF, Choma NN, Xu H, Miller Ra, Bastarache L, et al. Extracting timing and status descriptors for colonoscopy testing from electronic medical records. J Am Med Inform Assoc: JAMIA. 2010; 17(4):383–8.
3. Liu M, Jiang M, Kawai VK, Stein CM, Roden DM, Denny JC, Xu H. Modeling drug exposure data in electronic medical records: an application to warfarin. AMIA Ann Symp Proc. 2011:815–23.
4. Zhou L, Parsons S, Hripcsak G. The evaluation of a temporal reasoning system in processing clinical discharge summaries. J Am Med Assoc. 2008; 15(1):99–106.
5. Tao C, Solbrig HR, Sharma DK, Wei WQ, Chute GKSCG. Time-oriented question answering from clinical narratives using semantic-web techniques. The Semantic Web – ISWC. 2010:241–56.

6. Jung, H.; Allen, J.; Blaylock, N. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. Proceedings of BioNLP 2011 workshop; 2011. p. 146-54.

7. Duftschmid G, Miksch S, Gall W. Verification of temporal scheduling constraints in clinical practice guidelines. Artif Intell Med. 2002; 25(2):93–121. [PubMed: 12031602]

8. Terenziani P, Montani S, Torchio M, Molino G, Anselma L. Temporal consistency checking in clinical guidelines acquisition and execution: the GLARE's approach. AMIA Ann Symp Proc. 2003:659–63.

9. Zhou L, Hripcsak G. Temporal reasoning with medical data – a review with emphasis on medical natural language processing. J Biomed Inform. 2007; 40(2):183–202. [PubMed: 17317332]

10. Augusto JC. Temporal reasoning for decision support in medicine. Artif Intell Med. 2005; 33(1):1–24. [PubMed: 15617978]

11. Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. J Am Med Inform Assoc: JAMIA. 2013:1–8.

12. Verhagen, M.; Sauri, R.; Caselli, T.; Pustejovsky, J. SemEval-2010 task 13: TempEval-2. Proceedings of the 5th international workshop on semantic, evaluation; July 2010; p. 57-62.

13. Verhangen, M.; Gaizauskas, R.; Schilder, F.; Hepple, M.; Graham, K.; Pustejovsky, J. SemEval-2007 Task 15 TempEval temporal relation identification. Proceeding of the 4th international workshop on semantic evaluations (SemEval-2007); 2007. p. 75-80.

14. Derczynski L, Gaizauskas RJ. USFD2: Annotating Temporal Expresions and TLINKs for TempEval-2. Workshop on Semantic Evaluations (SemEval) ACL 2010. 2010:337–40.

15. Ha, E.; Baikadi, A.; Licata, C.; Lester, J. NCSU: Modeling temporal relations with Markov Logic and lexical ontology. Proceedings of the 5th international workshop on semantic, evaluation; July 2010; p. 341-4.

16. Bethard, S.; Martin, J. CU-TMP: Temporal relation classification using syntactic and semantic features. Proceedings of the fourth international workshop on semantic evaluations SemEval2007; June 2007; p. 129-32.

17. UzZaman, N.; Allen, J. TRIPS and TRIOS system for TempEval-2: extracting temporal information from text. Proceedings of the 5th international workshop on semantic, evaluation; 2010. p. 276-83.

18. Kolya, AK.; Ekbal, A.; Bandyopadhyay, S. JU_CSE_TEMP: a first step towards evaluating events, time Ex-pressions and temporal relations. Proceedings of the 5th international workshop on semantic, evaluation; 2010. p. 345-50.

19. Llorens, H.; Saquete, E.; Navarro, B. Tipsem (English and Spanish): evaluating crfs and semantic roles in tempeval-2. Proceedings of the 5th international workshop on semantic, evaluation; July 2010; p. 284-91.

20. Min, C.; Srikanth, M.; Fowler, A. LCC-TE: a hybrid approach to temporal relation identification in news text. Proceedings of the 4th international workshop on semantic evaluations (SemEval-2007); June 2007; p. 219-22.

21. Cherry C, Zhu X, Martin J, de Bruijn B. A la Recherche du Temps Perdu: extracting temporal relations from medical text in the 2012 i2b2 NLP challenge. J Am Med Inform Assoc: JAMIA. 2013 Mar.:1–6.

22. Tang B, Wu Y, Jiang M, Chen Y, Denny JC, Xu H. A hybrid system for temporal information extraction from clinical text. J Am Med Inform Assoc: JAMIA. 2013

23. Nikfarjam A, Gonzalez G. Pattern mining for extraction of mentions of adverse drug reactions from user comments. AMIA Ann Symp Proc 2011. 2011 Jan.:1019–26.

24. De Marneffe, M.; Manning, CD. The Stanford typed dependencies representation. CrossParser'08 Coling 2008. Proceedings of the workshop on cross-framework and cross-domain parser evaluation; September 2008; p. 1-8.

25. Joachims T. Text categorization with support vector machines: learning with many relevant features. Proc 10th European conf machine learning. 1998:137–42.

26. Joshi, M.; Penstein-Rosé, C. Generalizing dependency features for opinion mining. Proceedings of the ACL-IJCNLP 2009 conference short papers; August 2009; p. 313-6.

27. Manning, CD.; Klein, D. Accurate unlexicalized parsing. Proceedings of the 41st meeting of the association for, computational linguistics; 2003. p. 423-30.

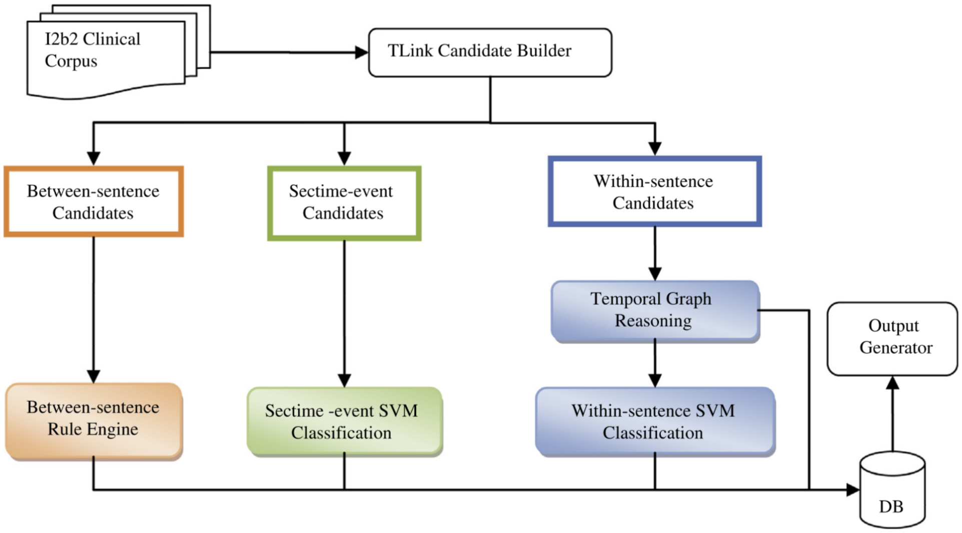
I2b2 Clinical Corpus
TLink Candidate Builder
Between-sentence Candidates
Sectime-event Candidates
Within-sentence Candidates
Temporal Graph Reasoning
Output Generator
Between-sentence Rule Engine
Sectime -event SVM Classification
Within-sentence SVM Classification
DB

Fig. 1.

Modules of the proposed system: between-sentence, sectime-event and within-sentence. Within-sentence candidates passed through the temporal graph reasoning module, and if the decision was not made, they were passed to the SVM for classification.

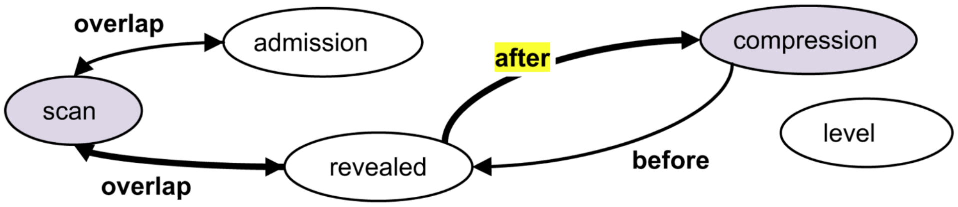

Fig. 2.

Sentence: “*The MRI scan on admission revealed an impending cord compression at the level of T10.*”; Simplified Sentence: “*scan on admission revealed compression at the level of T10.*” “*The MRI scan*” happened after “an impending cord compression”.

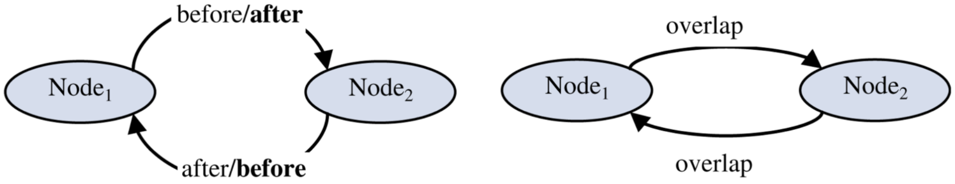

Fig. 3.

Corresponding edge labels in temporal graph; if the identified signal between Node$_1$ and Node$_2$ is *before* the reverse edge is also added (*after*) from Node$_2$ to Node$_1$.

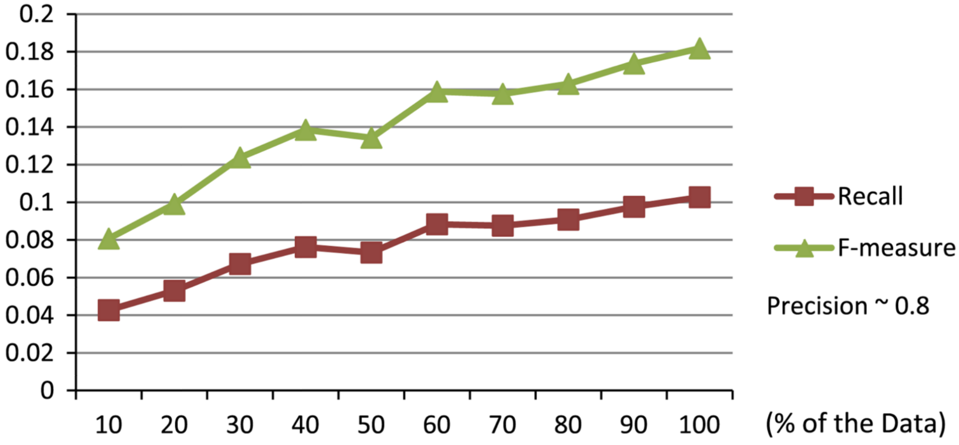

Fig. 4.
The effect of the training data size on the performance of the pattern-based approach.

Table 1

Examples of generalized sentences for pattern extraction.

Original sentence	Generalized sentence	Between tokens pattern
Dopamine and epinephrine given for cardiovascular support	[treatment_1] CC [treatment_2] given IN [treatment]	Treatment CC treatment
The patient's bilirubin level at 24 h of life was 4.6	[test_1] IN [date_2] IN NN was CD	Test IN date

Table 2

Extending temporal Links; A, B and C are events or temporal expressions.

If (A overlap B) *and* (B overlap C) *then* (A overlap C)
If (A before B) *and* (B [before| overlap] C) *then* (A before C)
If (A after B) *and* (B [after| overlap] C) *then* (A after C)

Table 3

The evaluation of the individual modules of the system (sectime-event, within-sentence (WS) and between-sentence).

Subtask	F-measure	Precision	Recall	Max possible recall
All	0.6412	0.7109	0.5839	~1
Sectime-event	0.3915	0.9221	0.2485	0.30
WS (SVM)	0.4256	0.6019	0.3292	0.43
WS (Graph)	0.2396	0.7044	0.1444	0.43
WS (Hybrid)	0.4291	0.5937	0.3360	0.43
Between-sentence	0.0395	0.5279	0.0205	0.27

Table 4

Comparison of the overall performance when the system was trained on the original vs. expanded TLinks.

Training data	F-measure	Precision	Recall
Original TLinks	0.6280	0.7569	0.5366
Expanded TLinks	0.6412	0.7109	0.5839