

Contents lists available at [ScienceDirect](http://www.sciencedirect.com)

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Temporal relation discovery between events and temporal expressions identified in clinical narrative

Yao Cheng^{*}, Peter Anick, Pengyu Hong, Nianwen Xue

Computer Science Department, Brandeis University, United States

ARTICLE INFO

Article history:

Available online 27 September 2013

Keywords:

Temporal relation
Clinical narrative
i2b2

ABSTRACT

The automatic detection of temporal relations between events in electronic medical records has the potential to greatly augment the value of such records for understanding disease progression and patients' responses to treatments. We present a three-step methodology for labeling temporal relations using machine learning and deterministic rules over an annotated corpus provided by the 2012 i2b2 Shared Challenge. We first create an expanded training network of relations by computing the transitive closure over the annotated data; we then apply hand-written rules and machine learning with a feature set that casts a wide net across potentially relevant lexical and syntactic information; finally, we employ a voting mechanism to resolve global contradictions between the local predictions made by the learned classifier. Results over the testing data illustrate the contributions of initial prediction and conflict resolution.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

In order to study development of diseases and effectiveness of potential treatments, it is very important to understand the temporal relations among all the clinically relevant events such as patient's symptoms (pre- and post-admission), medical test results, doctor's diagnoses, administered drugs and patient's responses. Starting from a patient's admission and continuing until after discharge, these events together with related temporal expressions, such as dates and times, are recorded in various clinical records. However, rather than employ a chronicle order and structured format, these records are usually written freely in natural language according to each doctor's writing styles. Therefore, they need to be transformed into structured data in which the events within each patient's history can be assigned to a "time line". This time line captures the essential clinical experience of a patient during the visit. With enough such structured data, systematic analysis can be performed and information about disease progression and treatment effectiveness can be obtained. However, given the sheer number of clinical records available, it is hard to imagine any individual or group can manually process these records and turn them into structured data, even for a particular type of disease. Therefore, an automated system is desired. This automated system should be able to examine natural language text in clinical records, identify clinically relevant events and temporal expressions (to-

gether, we call them entities of interest), then discover the temporal relations between these entities and reconstruct the time line.

The rest of the paper is organized as follows. In Section 2, we provide background information and briefly describe the temporally annotated data provided by the organizers of the 2012 i2b2 Shared Challenge. In Section 3, our three-step method for temporal relation discovery is presented. Results are summarized in Section 4. We conclude in Section 5 with some discussion.

2. Background

2.1. i2b2 Shared Challenges

The 2012 i2b2 Shared Challenge focused on promoting the development of temporal relation discovery systems for clinical records. As has been the tradition for the i2b2 series of shared challenges, each year the challenge organizers decide on one or several tasks, provide manually annotated clinical records as ground truth data, and evaluate the final output generated by participants' systems. This paper focuses on the subtask of temporally linking pre-identified events and time expressions within clinical records. In the rest of this section, we briefly describe previous work and the mark up language used to express temporal information in ground truth data.

2.2. Temporal relation discovery in clinical narratives

SPRUS [1] (later version MPLUS [2]), MedSyndikate [3] and the Medical Language Extraction & Encoding System (MedLEE) [4] are

^{*} Corresponding author. Address: 415 South Street, MS018, Waltham, MA 02453, United States.

E-mail address: yaocheng@brandeis.edu (Y. Cheng).

a few early information extraction systems developed for the clinical domain. Zhou et al. proposed an architecture for an integrated approach to processing temporal information [5] that uses MedLEE as its NLP component. Description and evaluation of their system (TimeText) can be found in [6]. Still, while representation of and reasoning about temporal information in text has generated growing interest within the general computational linguistics community [7], not much work has been done specifically in the domain of clinical narrative. Savova et al. note in their 2009 review [8] that most efforts in temporal reasoning have focused on processing structured data [9,10]. Since then, more attempts have been made to develop systems supporting entity extraction and relation discovery directly from un-structured clinical records. cTAKES [11,12] is a software system built on top of the UIMA framework and OpenNLP toolkit which is specifically adapted to the clinical domain. It provides capabilities of sentence boundary detection, tokenization, part-of-speech tagging, dependency parsing and named entity recognition. CNTRO [13] is an attempt to build a Semantic Web ontology so that powerful reasoning tools developed for Semantic Web can be applied to un-structured clinical narratives. There are also frameworks developed based on cTAKES and CNTRO which enable time-oriented question answering in this domain [14]. Jung et al. have made one such attempt to develop an end-to-end system based on deep natural language understanding (NLU) [15] that extracts entities and reconstructs their time line. Their system is an adaptation of the general purpose NLP system developed by Allen et al. [16].

2.3. TimeML and data annotation

With little or no enforced structure, clinical records are written primarily using natural language. In order to turn temporal information in free text into structured data, researchers from the NLP community have developed a markup language, TimeML [17], and an annotation schema for expressing events and temporal relations [18,19]. Data used in the 2012 challenge was annotated using a subset of tags defined in the TimeML framework. These tags include EVENT, TIMEX3, and TLINK, as well as an additional SECTIME tag for “section creation time”. According to their definition, EVENTS are situations that happen or occur. An event has “modality”, “polarity” and “type” as its attributes. Most events are noun phrases, although some verbs qualify as well. For example, in the sentence

EKG showed sinus tachycardia at 100 and echo revealed pericardial effusion, a 10 mm pulsus paradoxus was noted, and no evidence of tamponade.

the nouns *EKG* and *echo* are annotated as events of type “test”, *sinus tachycardia*, *pericardial effusion*, *pulsus paradoxus*, and *tamponade* are events of type “problem”, and the verbs *revealed* and *noted* are events of type “evidential”. All have modality labeled “factual” and all have polarity “pos(itive)” except for *tamponade*, which has polarity “neg(ative)”.

TIMEX3s are explicit temporal expressions with “type” and “value” attributes. In the sentence below, the expression *06/16* is a TIMEX3 with type “date” whose value is the canonicalized date, “1991-06-16”.

She was noted, on 06/16, to have numerous erythematous maculopapules on her back and chest.

A TLINK or Temporal Link represents the temporal relation between pairs of events or temporal expressions, or between an event and a temporal expression. The target entities involved in a TLINK are referenced by the attributes “fromID” and “toID”, while the “type” attribute represents the kind of the temporal relation.

As originally defined in TimeML, a TLINK type can be any of *simultaneous*, *before*, *immediately before*, *after*, *immediately after*, *includes*, *being included*, *during*, *during inverse*, *begins*, *begun by*, *ends*, *ended by* and *identical*. However, for the sake of simplicity and in order to increase inter-annotator agreement, these fourteen fine-grained types are reduced to three general relation types in this challenge: *before*, *after* and *overlap*. SECTIME represents a special kind of temporal expression that captures the creation times for the two primary sections in a discharge summary, which are *History of Present Illness* (HPI) and *Hospital Course* (HC). These two times are usually identical with patient’s admission and discharge time. Besides the attributes mentioned above, each tag also contains an “id” and “text” (both “fromText” and “toText” in the case of TLINK tags) attributes.

In our first example sentence above, the TLINK between *revealed* and *echo* is labeled “*overlap*”, since it is assumed that the revealing takes place during the test administration. The TLINK between *pericardial effusion* and *echo*, however, is labeled as “*before*”, since it is assumed that the problem occurred before the test was done. Not all possible TLINKs are explicitly annotated in the training data, however. For example, the relationship between *revealed* and *pericardial effusion* is omitted. This does not mean no temporal relationship exists; rather it implies that the relationship can be inferred from other relationships that are explicitly annotated. In this case, if *revealed* and *echo* overlap and *pericardial effusion* occurred before *echo*, and then one can infer that *pericardial effusion* also occurred before the *revealed* event.

Besides connecting entities within the same sentence, TLINKs appearing across different sentences are also provided in the training annotations (we call them between-sentence TLINKs). Most of these tend to relate co-referential entities in neighboring sentences, or link events to their corresponding section creation time.

3. Method

In the 2012 challenge, we were provided with 190 annotated clinical records as ground truth data and 120 records partially annotated as test data. Within both sets of records, events and temporal expressions were identified and labeled, but temporal relations (TLINKs) were only provided in the training data. (After the challenge ended, TLINKs in testing data were released for evaluation purpose.) The task was to build a system to automatically label all pairs of events and time expressions with their TLINK types (*before*, *after* or *overlap*). Given the success of machine learning methods for similar problems within the medical record domain, such as co-reference resolution and relation type classification, we first developed a machine-learning-only approach (MaxEnt) for labeling TLINKs both within and between sentences. This performed poorly on identifying between-sentence TLINKs, however, as many potential TLINKs were not annotated in the training data. For this reason, we moved to a rule-based approach for detection of between-sentence TLINKs, which gave us more control over candidate selection and labeling.

3.1. Training data pre-processing

The set of annotated TLINKs provided in the training data does not constitute a complete accounting of the temporal relationships among entities. Even within a sentence, some entity pairs may be unlabeled. Such pairs may be left unlabeled either because the temporal relationship cannot be determined from a reading of the text or else because the relationship can be inferred using other relationships between entities that have been labeled. While this saves work for the annotators, it diminishes the size of the training set for machine learning. In order to remedy this problem, we

calculated the transitive closure of all the annotated TLINKS within a sentence. For example, given the set of TLINKS {[A, B, *after*], [B, C, *after*]}, the inferable relation between A and C, namely [A, C, *after*], would be added. SputLink [20] was used to generate this transitive closure.

We initially tried to expand between-sentence TLINKS using the same process (while we were using a machine-learning based approach only). However, a large portion of the resulting TLINKS had no lexical, syntactic or contextual evidence to support them. So we resorted to a rule-based approach for between-sentence TLINK resolution, for which the transitive expansion process was no longer needed.

Next we normalized the representations of individual entity pair relationships by ordering each pair of entities according to their locations in the text, so that for every labeled pair [A, B, label], the entities A and B were put in their textual order and the temporal label was adjusted, as necessary, to be consistent with this order. So, for example, an initially out-of-order TLINK [B, A, *before*] would be replaced with the adjusted TLINK [A, B, *after*].

3.2. Candidate generation

As noted above, many of the between-sentence TLINKS are unlabeled in the training data. Initially, we attempted to account for these omissions by adding a “none” label category. This label was assigned to every pair of expressions that remained without a {*before*, *overlap*, *after*} label after applying transitive closure. The machine learned classifier suffered from extremely unbalanced data; due to the large number of possible entity pairs and relatively small number of annotations, even after expansion the training data was dominated by “none” labels, which in turn produced a classifier that was biased in favor of the “none” label. A two-step process where we first trained a classifier to separate “none” from valid labels and then used a second model to classify among valid labels also performed poorly due to error propagation from the first step.

Given that most annotated between-sentence TLINKS are either co-referential expressions or special relationships between an event and its corresponding section creation time, we abandoned our approach of enumerating all possible entity pairs in a document and instead relied on a set of rules to generate between-sentence TLINKS and their labels. For example, for two co-referential entities (identified heuristically using shared head terms), the link type would be “*overlap*”; for a relationship involving section creation time, rules would choose between the link types “*before*” or “*after*”.

For within-sentence TLINKS, we continued to use transitive closure to generate the complete set of inferable entity pair combinations as TLINK candidates for each sentence. A model was trained from the expanded within-sentence data.

3.3. Feature creation for within-sentence TLINKS

It is clear from manual inspection of the training data that many overt and subtle lexical and syntactic factors contribute to the recognition of temporal relations among the entities. In Example 1 below, the preposition *after* explicitly conveys the temporal relationship between *resolved* and *surgery*. In Example 2, it is the preposition *for* that implies that *pain* preceded the *operation*. In Example 3, the use of the adverbs *intraoperatively* and *postoperatively* distinguishes the time frames for their respective symptoms. In Example 4, the conjunction *and* implies a temporal sequence in this context, while in Example 5, it is the prepositional phrase *at the time of* that relates the condition with *discharge*.

Examples:

1. Her pain resolved after surgery.
2. This operation was performed for 2 months of increased rest pain.
3. She had a strong popliteal doppler pulse intraoperatively and good PVR on the right postoperatively.
4. The area of erythema on her left leg enlarged slightly and the patient was placed on Vancomycin for several days after which time her erythema again began to decrease.
5. The patient was afebrile with stable vital signs at the time of discharge.

In lieu of creating special purpose dictionaries and linguistic rules to capture such cases, we sought to design a feature set that would cast a wide net over the space of lexical and syntactic constructions. In order to extract such linguistic features, we made use of the general-purpose clinical record analysis toolkit cTAKES [11,12]. cTAKES’ output includes lexical information such as the lemma and part-of-speech tag for each token as well as syntactic information in the form of a dependency parse for each sentence. Combining information from cTAKES and entity annotation (EVENT and TIMEX3 tags in each record), we generated features of the following general types.

- Features indicating **positional** information about the entities, such as token distance between two entities, clinical section the sentence occurs in and whether two entities are adjacent to each other or consecutive entities within the sentence.
- Features associated with the lexical **context** around the entities, such as tokens occurring between them or to the immediate left or right of the entities.
- Features derived from the **dependency** parse of the sentence. As described below, the path from each entity to its closest shared parent node in the parse tree is computed. Any verb, adverb, preposition, conjunction, or noun found along one of these paths is captured in a feature, tagged with the entity along whose path it is found.
- Features capturing the **tense** of any entity whose head term is a verb.
- Features capturing general **properties** of entities, such as their semantic type, modality, part-of-speech (of head term); whether the two entities are identical or share the same head term; whether an entity name includes the prefixes *post*, *pre*, or *inter*.

Note that the contextual features and dependency features are both designed to discover diagnostic lexical items relating the two entities. In the case of contextual features, we use simple positional relationships, such as words to the left and right of entities and words appearing between the entities when the entities are relatively close (within three words) of each other. For the features based on the dependency parse tree, we employed the very general notion of a *path* between target entities. The path is the sequence of parent nodes up the dependency parse tree from the syntactic head of one entity to the first node encountered which is in the path of parent nodes above the syntactic head of the other entity. We hypothesized that using these syntactic paths would facilitate the capture of some temporally sensitive lexical relations not readily detectable using the simple positional relationships of the contextual features. An example dependency tree generated from cTAKES can be viewed in Fig. 1, where in our opinion, token “then” should depend on “may” instead of “surgery”. The complete feature list can be found in Table 1.

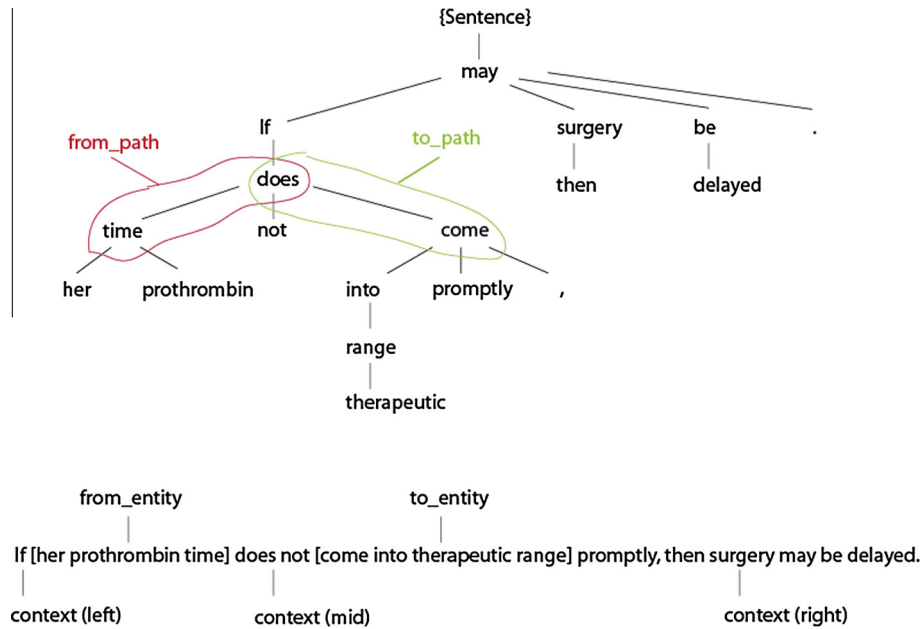


Fig. 1. Example of a sentence and its dependency tree generated by cTAKES.

Table 1 Complete feature list in groups.

Positional features	<ul style="list-style-type: none"> • Token distance between entities • If entities are adjacent to each other • If entities are consecutive in the sentence • Section number of the sentence
Contextual features	<ul style="list-style-type: none"> • Concatenation of tokens between entities (if less than 3) • Token to the immediate left of from_entity and its POS • Token to the immediate right of to_entity and its POS • Tokens of certain POS (VB, RBR, RBS, JJR, JJS, IN, TO, CC, WP, WRB) between entities • POS of verbs between entities • Prefixes (pre, post, inter) contained in the tokens between entities, to the left of from_entity and to the right of to_entity • If from_entity/to_entity is the object of “to” • Preposition if from_entity/to_entity is the object of one
Dependency features	<ul style="list-style-type: none"> • Head term of from_entity if from_entity heads to_entity and the term is a verb • Head term of to_entity if to_entity heads from_entity and the term is a verb • If from_entity/to_entity heads to_entity/from_entity • If some token heads both entities • Dependency root of both entities and root's POS • Tokens of certain POS (VB, IN, RB, NN, CC) in from_path • Tokens of certain POS (VB, IN, RB, NN, CC) in to_path
Tense features	<ul style="list-style-type: none"> • Tense of the head term of from_entity (could be non-verb or unknown) • Tense of the head term of to_entity • Tense pair of both entities
Property features	<ul style="list-style-type: none"> • Type, signature of from_entity/to_entity • Head term of from_entity/to_entity and its POS • If head term of from_entity/to_entity is a verb • If head term of both entities are identical • If both entities' string are identical • If from_entity/to_entity is a location • Prefixes (pre, post, inter) contained in from_entity/to_entity

3.4. Training

We used MALLET [21] with default parameters to train a maximum entropy classifier to label within-sentence TLINK candidates with one of the labels {before, overlap, after}.

3.5. Conflict resolution within sentences

Since our classifier (MaxEnt) assigns TLINK types to each edge independently, contradictions in the resulting temporal network may arise. For example, the set of TLINKs {[A, B, before], [B, C,

before], [A, C, after]} represents a situation of conflict. The semantics of temporal relations makes it possible to detect the existence of globally contradictory labels. In our case, the conflict resolution is only needed for within-sentence TLINKs, since the rules we use to generate between-sentence TLINKs are guaranteed to introduce no conflict across sentences. In order to resolve these within-sentence contradictions, we applied a network refinement strategy to the classification result.

For every three entities in a sentence, a TLINK triangle can be formed in which the edges represent the temporal relationships between the respective pairs of entities. Each TLINK triangle

therefore can be determined to either contain a temporal conflict in it or not. For those triangles containing a conflict, we identify the TLINK with the least confidence according to the classifier (weakest edge). For example, consider that TLINKs A, B, C form a “conflicting triangle” T1. A is classified as *before* with (MaxEnt) probability of 0.7, B is classified as *before* with probability of 0.8 and C is classified as *overlap* with probability of 0.6. In this case, the edge with the least confidence in T1 is C. In order to resolve the conflict, then, C’s label needs to be changed. However, C can also appear in another conflicting triangle, T2, in the same sentence. And in T2, C may not be the least confident edge, so its label needs to be kept. We must inspect all the triangles in the sentence in order to accumulate “change” and “keep” votes for each TLINK. The TLINK that gets the most “change” votes will be changed in order to resolve the conflict in this sentence. We have experimented with several variations of this algorithm in order to find the best result (for example, different weights can be assigned to a “change” or “keep” vote, “weakest edge” can be decided by different measures, etc.), which will be presented in the next section.

4. Results

4.1. Initial prediction

In this section, we first present individual results for between and within-sentence TLINK tasks. Then the combined overall performance of our system is reported. Using 98 testing documents, we evaluated our system with the evaluation script provided by the challenge organizer, which reports Precision, Recall, Average of Precision and Recall, and *F*-measure.

After examining several rule combinations, we decided on a set of three rules to generate between-sentence TLINKs, and they are listed below in from high priority to low priority. When multiple rules are applied to the same TLINK instance, rules with higher priority overwrite the result of low priority rules.

- Co-referential EVENTS are linked as “*overlap*”.
- EVENTS in the HPI section are always “*before*” admission temporal expression (TIMEX3).
- EVENTS in the HC section are always “*before*” discharge temporal expression (TIMEX3).

The result for between-sentence TLINK resolution is summarized in Table 2.

To assess the contribution made by different classes of features on the machine-learned model for within-sentence TLINKs, we conducted an ablation study in which models were built lacking various feature subsets. Table 3 shows the results of this study. (Note that each line reflects the removal of the specified feature subset only, not a cumulative removal of all subsets listed above it.) In order to determine whether the difference is statistically significant, we resampled the predicted TLINKs 25 times (with replacement), calculated *F*-measure from each sample, and then performed *t*-test between the samples lacking certain feature set and the samples with all features. As a result, except for the model trained lacking Tense features ($p < 0.1$), we were able to get a *p* value less than 0.05 for all the other feature set combinations, suggesting the differences are statistically significant. The effect of removing property features reflects the capacity for the semantic types of the entities (e.g., treatment, problem, type, department) to convey temporal clues. For example, a treatment usually temporally follows a problem. Likewise, two entities with the same name are likely co-referential, indicating a case of temporal overlap. Contextual lexical features also make a significant contribution, as expected, since the semantics of specific verbs and prepositions

Table 2

Result for between-sentence TLINK task.

	Precision	Recall	Average P&R	<i>F</i> -measure
between	0.7908	0.6905	0.7367	0.7372

Table 3

Ablation study showing the effects of removing sets of features for within-sentence TLINK task.

	Precision	Recall	Average P&R	<i>F</i> -measure
All features	0.4839	0.7695	0.5842	0.5942
w/o properties	0.4622	0.7419	0.5596	0.5696
w/o contextual	0.4681	0.7511	0.5662	0.5768
w/o dependency	0.4769	0.7629	0.5771	0.5869
w/o dep + contx	0.4402	0.7193	0.5351	0.5461
w/o positional	0.4757	0.7661	0.5774	0.5870
w/o tense	0.4792	0.7670	0.5797	0.5899

Table 4

Result for combined between and within-sentence TLINK task.

	Precision	Recall	Average P&R	<i>F</i> -measure
within + between	0.6028	0.6625	0.6275	0.6312

co-occurring with the entities carries a lot of temporal information. The dependency scores suggest that, in spite of some redundancy with contextual features, using the parse trees enabled the capture of lexical cues not available from the sequential context alone. Further finer-grained exploitation of parse tree features is probably warranted. Tense information for verb-headed entities, as well as positional features each proved moderately helpful.

Combining two sub-components, we report our overall system performance in Table 4.

4.2. Conflict resolution

The final step in our TLINK labeling process was the resolution of conflicting predictions within a sentence. We tested our strategy on a subset of 50 training documents and 50 testing documents, and calculated how many more correct TLINKs we achieved after the process. The total number of TLINKs in all 50 testing documents was 5645 and the system made 7180 initial predictions due to candidate generation strategy. As mentioned in the previous section, multiple variations of the main strategy were applied. The best result we achieved increased the number of correct TLINKs for 63 cases. Among the 7180 predictions, 3757 were correct predictions, 1882 incorrect, and 1541 were TLINKs that did not appear in the annotation (6 are missing from the predications because there are TLINKs between two identical entities annotated in the testing data, which are examples of violation of the annotation guideline). There are several reasons why conflict resolution did not boost the performance as we expected. First, we discovered that the “conflicting TLINK triangle” is not always a good indicator for the location of incorrect TLINKs. Among 1882 incorrect TLINKs, there were 1002 appearing in these conflicting triangles. This fact sets an upper bound for this strategy. Second, while the algorithm does successfully correct 465 incorrect TLINKs, it also alters 402 correct TLINKs’ labels. Third, the rest of the corrections were made to “false positive” TLINKs which did not even exist in the annotation. Fourth, the accuracy of the initial predictions (MaxEnt probabilities) affected the performance of the resolution. With a higher baseline of correct predictions, and stronger probabilities for those predictions, we would expect the strategy could be much more effective; conversely, if accuracy were lower than 50% to begin

with, then this approach to conflict resolution could result in an increase to the error rate.

5. Conclusions

We have presented an automated system for temporal relation discovery from clinical narratives consisting of three components: data expansion, rule-based candidate generation plus supervised machine learning, and conflict resolution. The temporal classification task has proved to be a difficult one because of the variety of ways that temporal information can be linguistically expressed (see previous section for the effect of contextual features). For example, in the sentence “*She again tolerated [the procedure] well and [did well] postoperatively*”, “the procedure” is classified as *overlap* with “did well” due to the conjunction “and”. But in fact, this is a *before* relationship hinted by the adverb “postoperatively”, where the part “-operative-” is referring to “the procedure”. In some other cases, decisions may rely on pragmatic domain knowledge not directly accessible through syntax. For example, in the sentence “*She is admitted now for a neuro-interventional radiology procedure to decrease the likelihood of [epistaxis] on [Coumadin]*”, “epistaxis” is classified as *overlap* with “Coumadin” because of the preposition “on” and possibly the entity type being PROBLEM and TREATMENT. However, if the model understood “epistaxis” is a problem caused by applying “Coumadin”, it should know this is actually an *after* relationship. Yet, the positive contribution of lexical features generated from a simple application of dependency parse trees suggests that syntactic structure can be further exploited to supplement lexical features relying on flat sequential relationships. A second problem, especially with respect to between-sentence relationships, is what to do when annotations fail to capture many temporal relationships. The transitive nature of temporal relations may in some cases make up for such missing data but in others it is more likely to propagate incorrect information through the network, making both learning and evaluation difficult. While we have concentrated our efforts on generic features and strategies, many opportunities exist for applying more domain-specific lexical and pragmatic knowledge to this task, such as the sequential relationships implicit between certain problems, treatments, and even a patient’s location and progress within the hospital setting. We look forward to tuning the feature set, further exploring conflict resolution strategies, and evaluating the system more thoroughly.

6. Funding

This project was funded by NIH NLM 2U54LM008748: Informatics for Integrating Biology and the Bedside (i2b2) PI: Isaac Koh-

ane and NIH NLM 1R13LM011411-01: Challenges in Natural Language Processing for Clinical Narratives PI: Ozlem Uzuner.

References

- [1] Haug PJ, Ranum DL, Frederick PR. Computerized extraction of coded findings from free-text radiologic reports. *Radiology* 1990;174(2):543–8.
- [2] Christensen LM, Haug PJ, Fiszman M. MPLUS: a probabilistic medical language understanding system. In: Proc workshop on NLP in the biomedical domain; 2002. p. 29–36.
- [3] Hahn U, Romacker M, Schulz S. MedSyndikate – a natural language system for the extraction of medical information from findings reports. *Int J Med Inform* 2002;67(1–3):63–74.
- [4] Friedman C. A broad coverage natural language processing system. *Proc AMIA Symp* 2000:270–4.
- [5] Zhou L, Friedman C, Parsons S, Hripcsak G. System architecture for temporal information extraction, representation and reasoning in clinical narrative reports. *Proc AMIA Symp* 2005:869–73.
- [6] Zhou L, Parsons S, Hripcsak G. The evaluation of a temporal reasoning system in processing clinical discharge summaries. *JAMIA* 2008;15(1):99–106.
- [7] Zhou L, Hripcsak G. Temporal reasoning with medical data – a review with emphasis on medical natural language processing. *JBI* 2007;40:183–202.
- [8] Savova G, Bethard S, Styler W, Martin J, Palmer M, Masanz J, Ward W. Towards temporal relation discovery from the clinical narrative. *AMIA Annu Symp Proc*. 2009;2009(1):568–72.
- [9] Augusto JC. Temporal reasoning for decision support in medicine. *Artif Intell Med* 2005;33(1):1–24.
- [10] Zhou L, Melton GB, Parsons S, Hripcsak G. A temporal constraint structure for extracting temporal information from clinical narrative. *JBI* 2006;39(4):424–39.
- [11] Savova G, Kipper-Schuler K, Buntrock JD, Chute CG. UIMA-based clinical information extraction system. *LREC* 2008: towards enhanced interoperability for large HLT systems: UIMA for NLP; 2008.
- [12] Savova G, Masanz J, Ogren P, Zheng J, Sohn S, Kipper-Schuler K, et al. Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *JAMIA* 2010;17:507–13.
- [13] Tao C, Wei WQ, Solbrig HR. CNTRO: a semantic web ontology for temporal relation inferencing in clinical narratives. In: AMIA annual, symposium, 2010. p. 787–91.
- [14] Tao C, Sobrig HR, Sharma DK, Wei WQ, Savova G, Chute CG. Time-oriented question answering from clinical narratives using semantic-web techniques. In: The 9th international semantic web conference, 2010.
- [15] Hyuckchul J, James A, Nate B, Will B, Lucian G, Mary S. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. *BioNLP* 2011 workshop.
- [16] Allen J, Byron D, Dzikovska M, Ferguson G, Gulescu L, Stent A, et al. An architecture for a generic dialogue shell. *J Nat Language Eng* 2000;6(3):1–16.
- [17] James P, José C, Robert I, Roser S, Robert G, Andrea S, et al. TimeML: robust specification of event and temporal expressions in text; 2003.
- [18] Pustejovsky J, Hanks P, Saur R, See A, Gaizauskas R, Setzer A, et al. The timebank corpus. In: *Corpus Linguistics*, 2003. p. 647–56.
- [19] Sauri R, Littman J, Knippen B, Gaizauskas R, Setzer A, Pustejovsky J. TimeML annotation guidelines; 2006.
- [20] Verhagen M, Mani I, Sauri R, Knippen R, Jang SB, Littman J, et al. Automating temporal annotation with TARSQI. *ACL* 2005 on interactive poster and demonstration sessions. p. 81–4.
- [21] McCallum AK. MALLET: A machine learning for language toolkit. 2002. <<http://mallet.cs.umass.edu>>.