Temporal expressions in clinical text: Event recognition and time expressions

Ivelina Nikolova, MA[1], Svetla Boytcheva, PhD[2],
Galia Angelova, PhD[1], K. Bretonnel Cohen, PhD[3]
[1] **Bulgarian Academy of Sciences**
[2] **American University in Bulgaria**
[3] **University of Colorado School of Medicine**

Abstract

Our approach to the i2b2 2012 task on temporality used a combination of machine learning and rule-based approaches. Regular expressions were used to recognize TIMEX3 expressions, and a hybrid system of machine learning and rule-based post-processing were used to recognize and classify events. We achieved an F-measure of 0.79 for event recognition on the original submission and 0.82 in subsequent experiments.

1 Introduction

Temporality, or the reflection of time, is an object of interest in natural language in general and in clinical notes in particular. A straightforward example of the challenge is to differentiate between mentions of past conditions (e.g. *Patient has a history of...*) versus present ones. However, the challenges can be much greater, and a sophisticated analysis of not just the semantics of temporal expressions, but the structure of clinical documents, will be necessary to get the temporal relations between events and time expressions right in this textual genre.

The issue of temporality has received a fair amount of attention in text mining. A primary issue is how to represent temporal relations, with theories on this ranging from first-order predicate calculus, which unfortunately fails for common verb combinations, to the reference point theory of[1]. One of the most important foci of recent research in temporality has been the two Tempeval shared tasks (a third one is planned for 2013). The most relevant aspect of the Tempeval shared tasks are Task 2 of the 2007 competition and Task D of the 2010 competition, in which the task was to label the temporal relation between pre-labeled events in a document and the Document Creation Time. In 2007, the six participants took varying approaches. These included machine learning systems, rule-based systems, and hybrid learning/rule-based systems—but none achieved results particularly different from the others, with the average accuracy being 0.76[2]. The task was repeated as Task D of Tempeval2; this time one system achieved spectacularly poor results (no details are given on the methodology), but the other systems mostly scored very similarly to each other, with an average accuracy of 0.78[3].[4] described a system for determining temporally coherent segments of clinical text and the relations between them, tackling only a very restricted ternary set of temporal relations, using machine learning. The primary advantage of their approach is that it outputs a global ordering, versus Tempeval-inspired approaches, which output only localized relationships.[4] achieved an F-measure of 0.83 on the temporal segmentation task and an accuracy of 78.3% on classifying the three sets of relations. This work was very ambitious in attempting temporal segmentation, on which it achieved very promising results, as described.

"Events" are a broad category with varying definitions. In previous i2b2 shared tasks, comparable tasks have included medical concept extraction, drug event recognition, health condition recognition, etc., although with a simpler representation. According to[5], in the 2010 i2b2 shared task, the most effective concept extraction systems used conditional random fields. In some of the approaches, the authors split the problem into subtasks, such as *(i)* detecting concept boundaries and *(ii)* concept class detection[6]. Other groups train CRFs on textual features enhanced with the output of a rule-based entity recognition system[7]. Others such as[8],[9] make use of CRFs combining existing named entity recognition systems and chunkers or similar algorithms, based on output from knowledge-rich sources. There are also approaches where semi-supervised CRFs utilize distributional semantics features. The best system from the challenge in 2010[10] achieves 0.852 exact F-measure. It uses an algorithm similar to CRFs—discriminative semi-Markov HMM, trained using passive-aggressive online update. The authors extract a large number of features, among which are token, context, sentence, section, document features, and features gathered from external tools such as cTakes, MetaMap, ConText, statistical parsers, etc.

Our approach involved a combination of machine learning and rule-based approaches. An extensive set of regular expressions was used for recognition of TIMEX3 expressions, and a machine learning approach was used for event recognition, combined with MetaMap [11] and rule-based post-processing.

2 Materials

This study is performed on the data provided for the i2b2 temporal relation extraction task. The documents are medical records in free text of patients diagnosed with various problems. The training documents are in XML format, containing the original text and 4 types of annotations—EVENT, TIMEX3, TLINK, and SECTIME. A detailed description of each type of annotation, its structure, its coverage, and its attributes was provided by the competition organizers. The records are organized in 3 sections: administrative data—(*i*) admission and discharge date; (*ii*) history of present illness; and (*iii*) hospital course. They are comparatively well segmented. Most of the documents contain about 40-60 sentences. The tables below provide descriptive statistics about the corpus.

Dataset	**Docs**	**Words**	**Sentences**	**EVENT**	**Unique EVENT**	**TIMEX3**	**Unique TIMEX3**
Train data	187	96 053	11 543	16 468	9 129	2 366	1 598
Test data	120	80 429	9 339	13 594	7 873	1 820	1 223

Table 1: Descriptive statistics for the training and test datasets

3 Methods

Time expression recognition

For the time expression recognition task, we use a rule-based approach. In the preprocessing phase, all phrases and keywords from the "Text" tag in TIMEX3 are collected. The next step includes clustering of the phrases and keywords. After analyses of some of the clusters, supplementary related keywords not present in the training set are added. For example, if the name of a month is present, the names of all months are added. Other examples are days of the week, and numbers in spelled-out form.

The order in which data for different TYPE attributes of TIMEX3 markers are extracted is important. First we begin with TIME data extraction, then we process DATE, and then we recognize DURATION and FREQUENCY information. In addition, for each of these types of TIMEX3 markers the order of the relevant rules is important—recognition progresses from more specific to more general rules. Each rule is represented as a regular expression, and an evaluation procedure was developed to find the corresponding attributes and to store the temporal information in ISO 8601 [12] standard normalized form.

We applied a processing pipeline performing the following steps:

- Sentence splitting
- SECTIME marker recognition: Because some of the TIMEX3 markers correspond to relative time, two important dates should be recognized first—the admission and the discharge dates.
- Time expression recognition: For this task 20 groups of regular expressions were developed. Some of the more important rules contain templates for different time format (complete or incomplete) information extraction about hours, minutes and seconds, and "a.m." or "p.m.", including 24-hour and 12-hour formats. These rules can process both digital and word form number (e.g. *3* or *three*). There are also some rules for relative time, such as "morning", "evening", "afternoon", "noon", "midnight", "night", "the end of the day", and "breakfast", "lunch", and "dinner". We use default values for them for time in normalized form. Some additional phrases like "which time", "that time", etc. are processed, but their normalization is ambiguous, because they can be classified both as TIME and DATE types. At this step we do not process any date information. The VAL attribute contains only data for the time in ISO 8601 standard normalized form. The date information extraction

and normalization is delayed until the end of the process of primitive time expression extraction. This allows us to treat equally both absolute and relative date information.

- Date expression recognition: We use 30 rules for "special" dates—public holidays in the United States. Additional data for Federal holidays and some religious, traditional, and informal holidays is provided. We also take into consideration variation both for holidays' names and date of celebration in different states and dates depending on the year. For instance, "Washington's Birthday" is known officially as "President's Day" in some states, and we recognize both names. "Good Friday" is celebrated the Friday before (western) Easter, but the Easter date varies from year to year. This means that we need first to calculate the date for Easter in the corresponding year and later to find the date on the Friday preceding this date.

  We use 35 rules for absolute and relative general date information extraction. For absolute date information, several regular expressions for different date formats (complete and incomplete) recognition are used. These rules can process both digital and word form numbers. The month information can be also presented as month number, full month name, month name abbreviation, or by using Roman numerals. For incomplete date format containing years only, we need to exclude that information describing the patient age, because some general greedy regular expressions can match it incorrectly. To keep the date information extraction process as simple as possible, we try to avoid usage of context information. This can cause incorrect classification of some lab test data as incomplete date representations. To cope with this problem we use current date procedure (calendar date between admission and discharge dates) and recent date procedure (we assume that patient age is a maximum of 100 years and the dates more than 100 years in the past are skipped). We test whether the extracted data corresponds to a recent date and exclude those that are outside of the range. We use as keywords "day", "month", "year", and "week" in incomplete date information extraction. For relative date information extraction, key word phrases like "post operative day", "day of life", "hospital day", "admission", "discharge", "same", "this", "that", "next", "following", "previous", etc. are used and combined with numerical representation of the time before or after. For some relative dates, "couple", "several", "many", "few", etc. are also used. For them we set a default value for time period.

  Some relative time information is represented by weekdays instead of dates. This module is implemented in C#. The DateTime class in C#, which also provides information for the weekday and allows automatic arithmetic with dates according to the current year month calendar data, is used. To cope with the problem of relative time information extraction, we use some temporary markers like "PREVIOUS" and "DELAY", "PA" (prior admission) and "PD" (prior discharge) for the VAL attribute during the primitive expression recognition for dates, duration, and frequency. Additional information is temporarily set to the TYPE attribute instead of DATE value to indicate the measure of the period, such as "D" for days, "M" for months, etc. The delayed normalization is further processed after primitive events extraction and sorting.

- Duration expression recognition: Another 25 rules are used for the duration expression extraction task. Keywords like "year", "month", "hour", "week", "minute", "second", "day", and some of their abbreviations are used for period measure recognition. The duration period is represented either using digital or word form numbers. For relative time, default periods for "few", "several", "couple", "many", "recent", "final", etc. are set.

  For correct type association, additional keywords such as "last", "prior", "next", "first" , "past", etc. are used, because some phrases such as "three weeks" can be classified either as DATE if they are immediately followed by keywords such as "ago" or "prior", or as DURATION in the case where they are preceded by "last" or "prior". Other keywords used for duration specifiy repetitions like "daily". We also process time periods like "the next three to seven days", which requires conversion of numerical word form numbers and calculation of the average time for this period. For "weeks" periods like "26 and 2/7 weeks", "32 and 03-05 week" and "35-1/7 week", additional rules for conversion between weeks and days for the time period are used. Other periods such as "45 minutes to 1 hour" also require conversion to "minutes" of the period in order to calculate the average period and then conversion back to hours. Some additional calculations are also necessary for periods represented like dates in the form "1/20-2/1/07". Because this case requires date information recognition, its recognition is not part of the primitive duration data recognition and its processing is delayed until the next steps.

- Frequency expression recognition: For this task, 20 rules are defined. They include recognition of key phrases

like “b.i.d”, “t.i.d.”, “q.i.d.”, “ad lib”, “p.r.n.”, “regular basis”, “several times”, and “multiple episodes”, which are associated with default values, and keywords like “per”, “every”, “times”, “q.d.”, “occasions”, etc. combined with numerical expressions. The repletion period is also specified by measures like “day”, “hour”, “minute”, “week”, etc. More complicated rules are used to process repetitions described as weekdays. For instance, “Tue / Th / Sat” requires calculation of the time period in days between listed weekdays. Unfixed repetitions, as in “x 3-4 per night”, also require calculation of the average time of repetitions and assignment of the “APPROX” value of the MOD attribute.

- TIMEX3 marker sorting: In this step, all primitive time events are sorted according to their START attribute value. This step is required for the next step of evaluation of postponed data concerning relative time information and also for combining data for time-date and durations including date information.
- Important clinical event time recognition: To evaluate some time markers, such as “post operative day”, “day of life”, “hospital day”, and other phrases marking important clinical events, time metadata should be extracted in advance—for instance, surgery date, infant’s delivery date, clinical procedure dates, etc. Additional keywords and patterns are used to extract the “Clinical History date” and “Hospital Course date”. For “post operative day” we look for absolute dates mentioned for a “surgery” procedure. In the case where there is no explicit information available, keywords such as “ultimately”, “urgent”, and “immediate” are used to indicate the admission date; an “operative report” date is used, if there is one.
- TIMEX3 marker filtering: For this step, 15 additional rules are used for processing postponed information. For temporary markers “PREVIOUS”, “PA”, “PD” and “DELAY”, the information temporarily stored into the TYPE attribute to present the period measure is used (see Example 1). Again using the DateTime class in C#, the date corresponding to the period is decreased automatically by the specified number indicating the period. The important clinical events time is used to find in normalized form the relative time data (see Example 2). For TIME and DURATION types, rules combining information from the consecutive TIMEX3 markers are used (see Example 3). After these processes, the TIMEX3 markers representing time information that is overlapped in the text are filtered. At the end of this step, unique ID numbers are associated to the final set of TIMEX3 markers.

Example 1:
The phrase “The next day” was extracted from the “Hospital Course” section of the discharge letter. Initially the following TIMEX3 tag is generated by the Date expressions recognition step:
<TIMEX3 id=“T0” start=“1571” end=“1583” text=“The next day” type=“DATE” val=“D1” mod=“PREVIOUS” />
Later the DATE type tag preceding this phrase in the discharge letter text is recognized from the TIMEX3 markers filtering step. Because the TIMEX3 tags are sorted, we select the last TIMEX3 tag with DATE type attribute immediately preceding the current one. In our example it is set to admission date value 10/15/1999, because “The patient was directly admitted”. Thus the final version of “The next day” normalization is:
<TIMEX3 id=“T8” start=“1571” end=“1583” text=“The next day” type=“DATE” val=“1999-10-16” mod=“NA” />

Example 2:
The phrase “postoperative day three” is initially stored by the Date expressions recognition step as:
<TIMEX3 id=“T0” start=“1631” end=“1654” text=“postoperative day three” type=“DATE” val=“3” mod=”DELAY”
textgreater
Later the operation date from the phrase “The patient was taken to the operating room on March 11 , 2002” is recognized from the TIMEX3 markers filtering step and after the TIMEX3 markers filtering step is stored in the following normalization:
<TIMEX3 id=“T6” start=“1631” end=“1654” text=“postoperative day three” type=“DATE” val=“2002-03-14” mod=“NA” />

Example 3:
The text “1/20-2/1/07” is initially recognized as the following two TIMEX3 tags with DATE type attributes:
<TIMEX3 id=“T0” start=“1093” end=“1097” text=“1/20” type=“DATE” val=“2019-01-20” mod=“NA” />
<TIMEX3 id=“T0” start=“1098” end=“1104” text=“2/1/07” type=“DATE” val=“2007-02-01” mod=“NA” />
Later from the TIMEX3 markers filtering step these two tags are merged into one TIMEX3 tag with DURATION type attributes:
<TIMEX3 id=“T7” start=“1093” end=“1104” text=“1/20-2/1/07” type=“DURATION” val=“P13D” mod=“NA” />

Event recognition

The event recognition task is approached based on supervised machine learning for event detection, with rule-based post-processing to filter false positives and to determine the attributes of the event.

Preprocessing

The first phase in the processing of the training data is the extraction of gazetteers from the annotated text. All phrases which are marked as EVENT are extracted. Duplicates are removed. The list formed in this way is referred to as the event gazetteer. This gazetteer is used as a reference for feature generation during the feature extraction phase. The procedure is repeated for annotations of type TIMEX3 and a time expression gazetteer is prepared.

In the second step, MetaMap[11] is run on the training data. The result is a list of mentions of medical concepts occurring in the text. MetaMap extracts matches of text phrases to medical concepts available in UMLS, their corresponding concept, semantic types, and a flag for negation if the term is negated. In this study MetaMap2011v2 is applied.

In the third preprocessing step, the training data is passed through a set of tools for part-of-speech tagging, chunking, and stemming within the GATE platform[13]. The integrated ANNIE[14] tools are used for tokenization and sentence splitting. The OpenNLP[15] tagger and chunker are applied afterwards, as well as the Snowball stemmer[16] .

Feature selection

The model for event recognition is trained with a conditional random field algorithm using five groups of features:

1. surface features
2. gazetteer-based features
3. contextual features
4. lexical and morphosyntactic features
5. semantic features

These are extracted for each token of the original documents. Punctuation is not excluded from the list of tokens—it is processed in all phases.

The first group of features is *surface features*. These are collected by applying patterns or simple manipulations to each of the tokens. These are: *token_length* (discrete value); *token_in_lowercase* (string); *contains_digits* (boolean); *is_alphanumeric* (boolean); and *token_orthography* (string).

The second group of features is obtained by using the extracted gazetteers as references. Although the aim at this stage is to recognize only the events, both gazetteers (the event gazetteer and the TIMEX3 expression gazetteer) extracted in the preprocessing phase are used. The rationale for including the TIMEX3 gazetteer at this step is that it will act as a discriminative feature between EVENT and TIMEX3 terms and thus help to recognize events. Each token receives one of the following values for each gazetteer feature:

- *IS_TERM* if the gazetteer contains an entry which equals the given token
- *IS_PART_OF_TERM* if the token is a substring of a gazetteer entry
- *NO* if the token is not a substring of any gazetteer entry

The third group of features is the contextual features. Word bigrams to the left and to the right are used for this feature.

The fourth group of features is the lexical and morphosyntactic features. They are extracted with a GATE[13] pipeline. They include:

- a *POS tag* for each token
- a *chunk marker* given by the chunker
- the *token stem*

The fifth type of features are the semantic ones. These are the semantically typed medical concepts extracted by MetaMap. The results obtained from MetaMap are stored in the internal representation of the document as MetaMap annotations. Each of these annotations has the following features: *matched text*; corresponding *concept label*; *negation flag*; and *semantic types*. If a token is contained in such a MetaMap annotation, the semantic type (or union of semantic types) of the annotation is copied to the *semantic type* feature of the current token. This feature has the role of filtering the tokens given their semantic type, when present. In the case where the feature is not present, this is an indication that the current token may not be part of a medical term. The events in the training data often contain recognizable medical terms, but time expressions are also often recognized by MetaMap as a medical term with temporal meaning.

Training/Labeling

The recognition models are trained with the Mallet[17] CRF implementation with its default settings. Several models were trained with different feature sets. The token sequences are labelled with BIO-annotation (Begin/Inside/Outside) as follows:
B-EVENT—token at the beginning of an event;
I-EVENT—token inside an event;
OTHER—negative example, i.e. token which is not part of an event.

10-fold cross-validation is used—the models are trained on 90% of the training data and tested on the remaining 10%. The successful models are retrained on the entire training data set and tested with the test data set. The results are presented in Table 4.

Postprocessing

The span which an event occupies is determined by the Mallet CRF labeling. A sequence of tokens is considered to be an event if the corresponding token labels are as follows:
B-EVENT [I-EVENT [I-EVENT [I-EVENT [...]]]] or
I-EVENT [I-EVENT [...]]
where the bracketed elements are optional and where each such sequence is preceded and followed by negative labels or by other event sequences.

The modality, polarity and type of the events are assigned by employing a set of rules and by using the NegEx software[18]. The modality is determined by analyzing the training set for triggering words. The modality attribute can accept 4 possible values: FACTUAL, CONDITIONAL, POSSIBLE, and PROPOSED. The events with modality FACTUAL are dominant (96% of all events) and the other events are about 1%–1.5% per value. Thus the value FACTUAL is considered to be the default one, and gazetteer based recognition is applied for the other three categories. A gazetteer with triggering words is created for each modality category by analyzing the context of the events in the training set. For example, the gazetteer of triggering words signalling events with modality POSSIBLE and their respective frequencies in the training corpus are as follows: *question* (24), *possible* (23), *likely* (21), *most likely* (15), *possibly* (9), *presumed* (6), *questionable* (5), *probable* (2), *suspected* (2), *apparent* (1). The gazetteers are provided to the NegEx software instead of a list of negation triggering terms. NegEx is run without changing its rules, e.g. using rules for negation. Even though the approach is rather naive and not tailored for this specific task, the results show quite high precision.

For determining polarity, the system uses NegEx output. The list of triggering terms is augmented with corpus-specific expressions.

The event type is determined on the basis of semantic type filters and gazetteers extracted from the training corpus. The filters are created by analyzing MetaMap semantic type annotations and event type dependencies. The type of an event is dependent on the semantic types of the tokens contained in the event expression span. In addition to the training data gazetteers, lists of ICD-9 drugs are used for recognition of events of type TREATMENT.

4 Experiments and results

Results of original submission

TIMEX3 recognition

The evaluation results for the time recognition task for the test set are listed in the table below. They show that even the Type and Mod attributes are identified with relatively high precision 0.94 and 0.93, respectively. The Val attribute precision is only 0.81. The main reason for this is the incorrect normalization of some cases of relative time events. Other problems are caused by ambiguous cases between different types of attributes. Some phrases can be classified both as DURATION and FREQUENCY type, for instance, depending on the context. For both cases we use different types and values are normalized in different format. This leads to incorrect values for Val and Type. The overall precision for the time recognition task is 0.86 and the recall is 0.77. The relatively small recall is due to the filtering step, because some TIMEX3 tags generated from the previous steps of primitive time type recognition are merged incorrectly. There are also some incomplete dates skipped by the rules and wrongly recognized as lab results.

System	Time Recognition Module
Gold standard	1 820
System output	1 664 (91,43%)
Recall	0,776783
Precision	0,865717
Average P&R	0,808497
F-measure	0,808392
Val match score :	0,818149
Type match score :	0,945658
Mod match score :	0,931255833

Table 2: Time recognition evaluation on the test set

[19,20,21,22,23,24] all present relevant work, primarily on the TIMEX corpus and TempEval shared tasks. Our results are comparable with the presented systems even with slightly lower results both for the precision and recall, and for Type and Val attributes as well, although tests are performed on different corpora.

Event recognition

The task of event recognition is approached with a combination of machine learning and rule-based approaches. The tokens are labelled by a model trained with a supervised machine learning algorithm, but the extraction of events from the resulting sequences of labels and assigning values to the event attributes is done in rule based settings (as explained in Section 3).

The baseline model for this study is trained only on a limited feature set: contextual features (left and right bigrams), surface features, and gazetteers; it is referred to here as *System I*. *System II* is the model which was used to generate our shared task submission. In addition to the *System I* feature set, it included semantic types extracted from MetaMap. The results are shown in Table 3 along with all system results on event recognition. The results which are shown in this table are produced by the automatic evaluation scripts provided by the i2b2 shared task organizers.

Results of experiments after submission

After submitting our results to the i2b2 shared task, we fixed some bugs and performed additional experiments on event extraction. A comparison of the methods' performance is shown in Table 3.

Event recognition

For convenience, we review the features of a baseline system, the system that produced our submitted results, and two systems that we used for post-submission experiments.

1. System I: This is the baseline system that we used in our initial experiments. It included the surface features, gazetteers, and surrounding bigrams.
2. System II: This is the system that produced the results of our original submission. It included the features of System I plus MetaMap conceptual features.
3. System III: This system included the features of System II, plus part-of-speech tags, chunking, orthographic features, and stems.
4. System IV included the features of System III plus a modification of the post-processing procedures. In *System IV* the labels are predicted by the same model as in *System III*, but the proceedure for selecting relevant subsequences as events is different. In Systems I-III the selection is done as described in section 3; in System IV it is taken into account that multiple events could be contiguous.
   E.g. *B–EVENT [I–EVENT [I–EVENT]] B–EVENT [I–EVENT [I–EVENT]]*
   In the latter version both sequences of 3 tokens in the example above are extracted as events.

The results for the baseline system (System I), the system that produced our submission (System II), and the two systems used in post-submission experiments are described in Table 3.

Comparing the four systems, we find that the relative number of recognized events increases with each successive elaboration of the system, from 69.52% in the submitted system up to 83.26% in the last and best *System IV*. There is a significant increase in recall from 65% initially, up to 76% in *System IV* (in total 11 percentage points). The tradeoff is a small decrease in precision of only about 2 percentage points. In sum the F-measure increases by about 6 percentage points and reaches 82%.

System	System I	System II	System III	System IV
Gold standard events	13 594	13 594	13 594	13 594
System output events	9 450 (69,52%)	10917 (80,31)	10 959 (80,62%)	11 319 (83,26%)
Recall	0,651020	0,718686	0,735365	0,757872
Precision	0,925303	0,885493	0,909995	0,908312
Average P&R	0,760393	0,788838	0,808540	0,821267
F-measure	0,760382	0,788813	0,808490	0,821223
Modality	0,933152	0,931658	0,930180	0,932005
Polarity	0,954637	0,950857	0,951373	0,954857
Type	0,712071	0,705148	0,705877	0,702273

Table 3: Event recognition evaluation on the test set

5 Discussion and error analysis

Our results show that a hybrid machine learning/rule-based system can achieve relatively high F-measures on the event detection task, probably comparable to typical human agreement on an annotation task, and can be achieved with relatively simple featural representations. MetaMap and NegEx were the most semantically advanced feature

extractors that we used. This is not to say that they are simple feature sources—indeed, both are semantically quite sophisticated. However, they are publicly available and are well within the reach of any system builder.

Our results on the TIMEX3 event detection show that regular expressions and rule- and knowledge-based approaches remain robust natural language processing tools. They also reveal a previously uninvestigated finding in cross-lingual temporal expression extraction for clinical texts—many of our successful regular expressions for finding TIMEX3 expressions were translated from a previously built system for text mining from Bulgarian clinical texts.

6 Contribution and error analysis of MetaMap

MetaMap and the semantic types that it returns for each concept have a central role for the event feature generation, model training, and event type detection. Some details of the contribution of MetaMap to the evaluation results are given here. In our error evaluation, we manually mapped the event annotations in the training data and the concept annotations extracted from MetaMap. It was found that 12 566 MetaMap concepts do not overlap with any event in the training data (this is about 76% overgeneration) and only 158 events have no overlapping MetaMap concept (0.95%). In many cases MetaMap annotations cover mentions of the patient, such as *male, female, patient*, and *Mr.* Since these are never marked as events in the training corpus, such results do not harm the model learning. On the other hand, often the dates in the text are not recognized correctly by MetaMap and are given the wrong semantic type. This fact affects the event detection, because the semantic type *Temporal concept* is a sign for time expression and and an implicit sign that the concrete term is not an event. In such cases, the wrong MetaMap semantic types cause noise in the description of the negative examples. Despite these shortcomings, the results in Table 3 show that the semantic type contributes to both precision and recall in comparison with the system based on surface features only.

7 Conclusion

The results reported here show that even relatively simple systems can achieve respectable scores on the event detection and TIMEX3 detection for clinical text. They also suggest that the annotation effort was well-done and that the data is reliable. Furthermore, they suggest that temporality in clinical text is a tractable object of research in biomedical natural language processing. Nonetheless, the results also reveal considerable room for improvement, and suggest that investigation should continue into this interesting task. The authors recommend a repeat of this task in the following year, so that the community can see just how much room for improvement there is and improve on the performance of the extant systems that were built in response to the shared task.

References

1. Reichenbach H. *Elements of symbolic logic*. The Macmillan Company, 1947.
2. Verhagen M, Gaizauskas R, Schilder F, Hepple M, Moszkowicz J and Pustejovsky J. Semeval 2007 task 15: Tempeval temporal relation identification. 2007.
3. Verhagen M, Sauri R, Caselli T and Pustejovsky J. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, 2010.
4. Bramsen P, Deshpande P, Lee YK and Barzilay R. Finding temporal order in discharge summaries.
5. Uzuner O, South BR, Shen S and DuVall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *JAMIA*, pages i116–i124, 2011. doi: doi:10.1136/amiajnl-2011-000203.
6. Roberts K, Rink B and Harabagiu S. Extraction of medical concepts, assertions, and relations from discharge summaries for the fourth i2b2/VA shared task. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.
7. Gurulingappa H, Hofmann-Apitius M and Fluck J. Concept identification and assertion classification in patient health records. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.

8. Jiang M, Chen Y, Liu M, Rosenbloom ST, Mani S, Denny JC et al. Hybrid approaches to concept extraction and assertion classification - Vanderbilt's systems for 2010 I2B2 NLP Challenge. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.

9. Kang N, Barendse R, Afzal Z, Singh B, Schuemie MJ, van Mulligen EM et al. Erasmus MC approaches to the i2b2 Challenge. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.

10. de Bruijn B, Cherry C, Kiritchenko S, Martin J and Zhu X. NRC at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.

11. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of AMIA Symposium*, pages 17–21, 2001.

12. Data elements and interchange formats – information interchange – representation of dates and times.

13. Cunningham H, Maynard D, Bontcheva K, Tablan V, Aswani N, Roberts I et al. *Text Processing with GATE (Version 6)*. 2011. ISBN 978-0956599315. URL `http://tinyurl.com/gatebook`.

14. Cunningham H, Maynard D, Bontcheva K and Tablan V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002)*, 2002.

15. Apache OpenNLP. URL `http://opennlp.apache.org/`.

16. Snowball stemmer. URL `http://snowball.tartarus.org/`.

17. Machine learning for language toolkit (mallet). URL `http://mallet.cs.umass.edu/`.

18. Chapman WW, Bridewell W, Hanbury P, Cooper GF and Buchanan BG. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34:301–310, 2001.

19. Pustejovsky J, Hanks P, Sauri R, See A, R Gaizauskas, Setzer A et al. The TIMEBANK Corpus. In *Proc. of Corpus Linguistics 2003*, page 647656, 2003.

20. Pustejovsky J, Castano J, Ingria R, Sauri R, Gaizauskas R, Setzer A et al. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proc. of the Fifth Int. Workshop on Computational Semantics (IWCS-5)*, pages 1–11, 2003.

21. MR Boland, Tu SW, Carini S., Sim I. and Weng C. EliXR-TIME: A Temporal Knowledge Representation for Clinical Research Eligibility Criteria. In *Proc. of AMIA 2012 Clinical Research Informatics Summit*, pages 71–80, 2012.

22. Chang AX and Manning CD. SUTIME: A Library for Recognizing and Normalizing Time Expressions. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, 2012.

23. Strotgen J and Gertz M. HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions. In *Proceedings of the HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions*, 2010.

24. UzZaman N and Allen J. Trips and trios system for tempeval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283, 2010.