A lightweight, pattern-based approach to identification and formalisation of TimeML expressions in clinical narratives

Phil Gooch[1]
[1] **Centre for Health Informatics, School of Informatics, City University London, UK**

Abstract

General Architecture for Text Engineering (GATE) components for identifying clinical events and temporal expressions are developed and evaluated against a corpus of 120 discharge summaries.

Introduction

TimeML[1] is an emerging standard for capturing and reasoning with temporal expressions occurring in narrative text. It has representation primitives for the annotation of events (`<EVENT>`), temporal expressions (`<TIMEX3>`), and the relationships between events or between an event and a temporal expression (`<TLINK>`). TimeML defines events as 'tensed or untensed verbs, nominalizations, adjectives, predicative clauses, or prepositional phrases'[1]. From a clinical perspective, a TimeML `<EVENT>` can be a clinical concept (e.g. '*type 1 diabetes mellitus*', '*renal function*'), a verb or verb group representing a process (e.g. '*should be referred*', '*will be discharged*'), or a concept or process modifier (e.g. '*decreasing*', '*severe*'). A `<TIMEX3>` annotation captures specific, relative or approximate dates, durations and frequencies. For example, the approximate duration '*for at least 3 days*' would be specified in TimeML as

```
<TIMEX3 tid="t0" type="DURATION" value="P3D" mod="EQUAL_OR_MORE">3 days</TIMEX3>
```

Informatics for Integrating Biology & the Bedside (i2b2) have recently made available a training corpus of 190 discharge summaries manually annotated with TimeML `<EVENT>`, `<TIMEX3>`, and `<TLINK>` annotations. In this paper, we report the development of a system for identifying and classifying events and temporal expressions in clinical notes, and structuring these expressing into formalised TimeML `<EVENT>` and `<TIMEX3>` expressions.

Methods

Components for identifying clinical concepts, events and temporal expressions were developed as resources in the General Architecture for Text Engineering (GATE) framework[2] using the GATE Java API and the Java Annotation Pattern Engine (JAPE) rules engine for writing lexico-syntactic patterns over existing annotations and features. These were assembled in a processing pipeline following generic GATE components for tokenization, sentence splitting, part of speech (POS) tagging, noun phrase (NP) and verb group (VG) chunking.

A lightweight clinical concept recogniser was created by extracting lists of distinct SNOMED CT terms for the McCray et al.[3] UMLS semantic type groupings of `Anatomy`, `Disorders`, `Procedures` and `Chemicals & Drugs` by running SPARQL queries against the NCBO BioPortal for biomedical ontologies[1]. For example, the following SPARQL query will extract *Neoplastic Process* concepts (part of the `Disorders` group):

```
SELECT DISTINCT *
 FROM <http://bioportal.bioontology.org/ontologies/SNOMEDCT>
 FROM <http://bioportal.bioontology.org/ontologies/globals>
WHERE {
      ?termURI a owl:Class ;
      skos:prefLabel ?prefLabel ;
      <http://bioportal.bioontology.org/ontologies/umls/hasSTY>
        <http://bioportal.bioontology.org/ontologies/umls/sty/T191> .
}  ORDER BY ?prefLabel
```

[1] http://sparql.bioontology.org/

Terms were then decomposed into distinct morphemes (tokens and neoclassical prefixes, roots and suffixes), which reduced the size of the data set dramatically, from ∼28 MB to ∼32 KB. Each morpheme was classified according to its POS; noun phrases and prepositional phrases composed of these morphemes are then identified as candidate terms and mapped to the `Problem`, `Test` and `Treatment` types used in the i2b2 corpus. To handle abbreviated clinical expressions, a lookup list of expanded medical abbreviations was extracted from Wikipedia[2] and manually grouped according to their semantic type (`Problem`, `Test` or `Treatment`), as detailed elsewhere[4].

As noted in the Introduction, TimeML events can be clinical concepts that describe a patient state, such as a symptom, disorder or disease; a procedure, such as a test or treatment; or some process or occurrence that affects the patient, such as admission, discharge, referral etc, and their corresponding verb forms. Therefore in addition to `Problem`, `Test` or `Treatment` concepts, we also consider verb groups to be potential events, labelled separately as `Occurrence`.

The well-known NegEx algorithm[5] for detecting negated clinical findings and ConText[6] for identifying their temporal, hypothetical and experiencer (i.e. patient or family member) contexts. NegEx uses a list of 270 'trigger terms' that may appear before or after a clinical concept (usually a disease, finding or symptom) that indicate whether the concept is possible (e.g. '*may not be ruled out*') or negative ('*no evidence of*', '*was ruled out*'), and has a reported precision and recall of 84.5% and 77.8% against a test set of 1235 concepts in 1000 sentences extracted from discharge summaries. While NegEx is available as a separate GATE framework component, it was developed for an earlier version (4.x), it requires a number of sub-components to be installed and instantiated in a specific order in the pipeline, and this author was unable to get it to work with the latest version of GATE (7.x). Therefore, a separate negation and possibility component was developed.

The ANNIE Verb Group (VG) chunker identifies sequences of verbs, including negations and modals. For example the phrase '*may not be appropriate to prescribe*' will be identified as two VGs: '*may not be*' with negated ('not'), modal ('may') features and '*to prescribe*' with an infinitive tense feature. For verb-group type events, these features were used to identify possibility and negation. Rather than recreate NegEx and use its lists of specific expressions, lexical patterns written in JAPE were used to generalise the identification of these features. For example, a concept preceded by a negating verb group (e.g. '*was not found*') or word (e.g. *no*, *absence* etc) within a certain window (e.g. between 0 and 3 intervening words) suggests that that concept is negated.

```
(
    (
        {VG.neg == "yes"}
        (
            TOKEN_WINDOW
            CONCEPT
        )[1, 5]
    ) |
    (
        CONCEPT
        {VG.neg == "yes"}
    )
):m
```

To ensure double negatives or negative possibility is not captured (e.g. '*does not exclude*'), negating phrases are only matched if they do not begin with 'not', as shown in the pattern below:

[2] http://en.wikipedia.org/wiki/List_of_medical_abbreviations

```
!["not"]
(
    (
        ["no|nor|any|deny|denie(s|d)|without|
        absen(t|ce)|exclude(d|s)|negative"]
    ) |
    (
        ["rule(s|d)?"]
        TOKEN_WINDOW
        ["out"]
    )
    (
        TOKEN_WINDOW
        CONCEPT
    )[1, 5]
)
```

where TOKEN_WINDOW is a flexible window of intervening words, and CONCEPT is a clinical term or event identified by a concept recognition pipeline step. A similar pattern can be expressed to represent a negating expression following the concept. We use a similar approach to identify possibility via the presence of 'hedge cues' [7] – words that indicate uncertainty or speculation:

```
(
    "possib(le|ility)|potential(ly)?|presum(e|ed|able|ably)|
    question(ed|able|ably)?|consistent|indicate(s|d)?|
    suggest(s|ed|ive)?|risk(s|ed)?"]
    (
        TOKEN_WINDOW
        CONCEPT
    )[1, 5]
)
```

The GATE Tagger_Numbers[3] component was used to identify and normalise spelt-out numbers and roman numerals to their arabic equivalents. This component does not handle ordinal numbers (21st, fourth etc), so a separate Gazetteer of ordinals from 1-31 (for day of the month identification) was created, e.g.

```
...
24th;val=24
twenty-fourth;val=24
twenty fourth;val=24
...
```

Gazetteer lists of units of measurement, their abbreviations and modifiers (in symbolic and text form e.g. '*less than*', '*at least*', <, >=) were created, and the output of these lookups were combined with JAPE patterns to identify clinical relevant measurement concepts such as values and ranges of weight, volume (e.g. for drug dosages), length (e.g. for tumour sizes) and pressure. Similarly, concepts of age, duration, frequency, and date/time were identified with JAPE string patterns combined with Gazetteers of month and day names, and relevant temporal units and their abbreviations. For example:

[3] http://gate.ac.uk/sale/tao/splitch21.html#sec:misc-creole:numbers

```
(
  {Number}
  {Lookup.majorType == time, Lookup.minorType == duration}
):expr
-->
  :expr.Duration = {value=:expr.Number.value, unit=:expr.Lookup.unit,
  period=:expr.Lookup.period, prefix=:expr.Lookup.prefix}

(
  {Duration}
  ("of")
  ["age"]
):expr
-->
  :expr.Age = {}
```

where entries for units of duration and their features are identified from the Gazetteer:

```
...
day;unit=H;period=24;prefix=P
days;unit=D;period=1;prefix=P
wk;unit=D;period=7;prefix=P
wks;unit=D;period=7;prefix=P
week;unit=D;period=7;prefix=P
weeks;unit=D;period=7;prefix=P
...
```

This allows duration values to be formalised according to the TimeML standard[1]. For example, '*for three weeks*' has a TimeML value of `P21D`, generated by multiplying the `value` and `period` features extracted by the above patterns and prepending and appending the `unit` and `prefix` features.

Frequency concepts are identified by similar patterns and gazetteers (e.g. *daily*, *weekly*, *once*, *twice*), where the TimeML value is calculated by dividing the the `period` by the `value` features. Expressing singular 'day' concepts in hours allows frequency values to be calculated more accurately, e.g. '`three times a day`' → value=3, period=24, frequency value=24/3=8, TimeML value=`RP8H`; and '`twice daily`' → value=2, period=24, frequency value=24/2=12, TimeML value=`RP12H`.

TimeML defines a generic `TIMEX3` tag for duration, date, time and frequency concepts where each is distinguished by a '`type`' feature. The distinct annotations created for each in the first pass through the document are converted to them to a single `TIMEX3` annotation in a second pass (e.g. `Duration` → `TIMEX3.type=Duration`). In this second pass, number ranges or numbers preceded by a modifier in temporal expressions were given a '`mod`' feature as per the TimeML standard (e.g. '*no more than 3 days*' → `Duration.value="3", unit="D", mod="EQUAL_OR_LESS"`). These mappings were set up as follows: each Gazetteer entry has three features: positive context, one for negation, and pre-modifier 'or', for example:

```
more;pos=MORE_THAN;neg=LESS_THAN;or=EQUAL_OR_MORE
earlier;pos=LESS_THAN;neg=MORE_THAN;or=EQUAL_OR_LESS
```

These are matched by the corresponding patterns:

```
(
  {Lookup.majorType == value_modifier}
  {Number}
):mod
-->
  :mod.NumberModifier = {mod=:mod.Lookup.pos}

(
  ("no|not|never")
  {Lookup.majorType == value_modifier}
  ("than")?
  {Number}
):mod
-->
  :mod.NumberModifier = {mod=:mod.Lookup.neg}

(
  {Number}
  ("or")
  {Lookup.majorType == value_modifier}
):mod
-->
  :mod.NumberModifier = {mod=:mod.Lookup.or}
```

Although GATE includes a component for identifying and normalising date values, it does not identify abbreviated dates as typically occur in clinical notes such as ‘on 8/26’ (i.e. 26 August) or handled relative dates such as ‘*on the third post-operative day*’ or ‘*on the day before discharge*’, a separate component was developed for this purpose, again using JAPE expressions. For example, for UK dates:

```
(
  (DAYOFMONTH):day
  (DATESEP)
  (MONTH):month
  (DATESEP)
  (YEAR):year
):dt
-->
  :dt.Date = {day=:day.Token.string,
  month=:month.Token.string, year=:year.Token.string}
```

For US dates:

```
(
  (MONTH):month
  (DATESEP)
  (DAYOFMONTH):day
  (DATESEP)
  (YEAR):year
):dt
```

where DATESEP = ”/” or ”-” and the following regular expressions identify month, day and year expressions:

```
MONTH = (0?[1-9]) | (1[0-2])
DAYOFMONTH = (0?[1-9]) | (1[0-9]) | (2[0-9]) | (3[0-1])
YEAR = ([1-2][0-9]{3}) | ([0-9]{2})
```

Shorter date expressions, on their own, are ambiguous (11-12 could represent a value range, 11 December or 12 November, depending on locale), so patterns for matching these require a preceding prepositions and fixed locale. For example, to match US mm/yy (e.g. 8/92) or US mm/dd or mm-dd (e.g 09-26):

```
("on|in|from|before|after|during")
(
  (MONTH):month
  (DATESEP)
  (YEAR):year
):dt
```

```
("on|in|from|before|after|during")
(
  (MONTH):month
  (DATESEP)
  (DAYOFMONTH):day
):dt
```

For relative dates, such as '*2 days before admission*':

```
(
    ({Duration}):dur
    ({TemporalRelation})?:rel
    ({Event}):evt
):dt
-->
    :dt.Date-Rel = {rel=:rel.TemporalRelation.type, mod=:dur.Duration.mod,
    period=:dur.Duration.period, value=:dur.Duration.value,
    interval=:dur.Duration.unit, event=:evt.Event@string}
```

and for dates relative to a nonspecific event (e.g. '*three days ago*'):

```
(
    ({Duration}):dur
    ({TemporalRelation}):rel
):dt
-->
    :dt.Date-Rel = {rel=:rel.TemporalRelation.type, mod=:dur.Duration.mod,
    period=:dur.Duration.period, value=:dur.Duration.value,
    interval=:dur.Duration.unit}
```

In the i2b2 discharge summaries, admission and discharge dates are explicitly identified in separate headings in the text. In GATE, we store these as document-level features, to allow normalisation of relative and abbreviated dates. For example, if we know that the admission date was 2011-12-16 and the discharge date was 2012-01-18, then short dates such as 12/26 can be normalised to 2011-12-26.

Similarly, given an expression such as 'three weeks after discharge', which generates a `Duration` concept with value `P21D` (see above), methods from the Java `Calendar` class can be used to generate a date 21 days after 2012-01-18, i.e. 2012-02-08.

Anaphoric date and duration expressions, such as '*on that date*' and '*during that time*' are linked back to the most recent, fully specified date or duration earlier in the document.

The performance of event boundary detection, type (e.g. `Treatment`, `Problem`, `Occurrence`), polarity (negation: either `POS` for positive or `NEG` for negated events), modality (possibility: either `FACTUAL` for events determined be true, `POSSIBLE` for events that may or may not be true, `PROPOSED` for planned events), was evaluated against a withheld test corpus of 120 discharge summaries provided by i2b2 for their 2012 Natural Language Processing Challenge on temporal relations[4]. This test data was provided by i2b2 in UTF-8 XML format and a Python script provided by the challenge organisers to evaluate the system output against the manually annotated test data.

The accuracy of temporal concept boundary detection, type, TimeML value and modifier was also evaluated against this test corpus. As the data originate from US healthcare providers, dates were in US format, so when running the pipeline against this data, the pattern for identifying UK dates (dd-mm-yyyy) was disabled.

Results

Table 1 shows the document macro-averaged and corpus micro-averaged precision, recall and F_1-measure scores for system-generated `EVENT` extents (boundary detection), and `Type` (`Problem`, `Test`, `Treatment`, or `Occurrence`[5]), `Polarity` (negative or positive) and `Modality` (possibility) F_1-measure scores for feature assignment, for the 120 discharge documents from the 2012 i2b2 corpus.

As shown in the table, corpus micro-averaged F_1-measure scores for `Type`, `Polarity` and `Modality` are significantly lower than the document macro-averaged scores as a result of the different way each is calculated. The macro scores show the average score per document for these features, given system events whose extents match the gold standard, whereas the micro scores show the average score over the whole corpus, taking into account false positives and false negatives. In other words, for a given system-generated `EVENT` annotation that matches or overlaps a gold-standard `EVENT` annotation, the accuracy (as measured by F_1) of negation and possibility assignment is 93% and 94% respectively, whereas these are reduced to 57% and 58% as a result of the number of false negatives (recall was only 62%, so 38% of all events were missed) and false positives (precision was 82%, so 18% of events were falsely identified as such).

Method	Precision	Recall	F_1	Type	Polarity	Modality
Macro	0.82	0.63	0.71	0.84	0.93	0.94
Micro	0.82	0.62	0.70	0.51	0.57	0.58

Table 1: Identification of events, negation and possibility: macro- and micro-averaged metrics over 120 discharge summaries.

Table 2 shows the document macro-averaged and corpus micro-averaged precision, recall and F_1-measure scores for system-generated TIMEX3 extents (boundary detection), and `Type`, `Val` and `Mod` F_1-measure scores for feature assignment, for the 120 discharge documents from the 2012 i2b2 corpus. The `Type` represents TIMEX3 type assignment accuracy (i.e. `Duration`, `Date`, `Time`, or `Frequency`) . The `Val` score represents accuracy of TimeML value calculation. This takes into account unit conversion, e.g. a value of `P36H` in the system output will score as a match against a value of `P1.5D` in the gold-standard output (provided the concept extents and `Type` also match). The score for the `Mod` feature represents accuracy of the TimeML modifier (`NA` for specific values, `LESS`, `MORE`, `APPROX`, `START`, `END` and `MIDDLE` for concepts preceded by an appropriate modifier).

As shown in the table, corpus micro-averaged F_1-measure scores for `Type`, `Val` and `Mod` are significantly lower than the document macro-averaged scores as discussed above. These differences provide that, for a given document, there accuracy of type and normalised value will be 90% and 78%, respectively, for a given temporal annotation identified by

[4] https://www.i2b2.org/NLP/TemporalRelations/Main.php

[5] a verb group

the system that matches the same annotation in the gold standard, whereas over the corpus as a whole, these accuracies are reduced to 68% and 59%, due to the effect of false negative and false positive system TIMEX3 extents over the corpus as a whole.

Method	Precision	Recall	F_1	Type	Val	Mod
Macro	0.85	0.80	0.81	0.90	0.78	0.88
Micro	0.83	0.77	0.80	0.68	0.59	0.68

Table 2: Temporal concept identification: macro- and micro-averaged metrics over 120 discharge summaries.

Discussion

The low recall in event detection was largely a result of abbreviations in the discharge summaries (e.g. '*benzos*', '*the Rita*'), trade names for drugs ('*Klonopin*') not picked up by the concept recogniser. Reduced precision resulted from identifying generic verb groups as event occurrences. If we take the macro scores in order to consider the performance of negation and possibility assignment separately, then the simple lexical patterns described here appear to have performed well, giving F_1 measures of 0.93 and 0.94 for negation and possibility assignment respectively. This suggests that, on the evaluation corpus at least, simple patterns, rather than specific, hard-coded expressions as used by NegEx, perform well – Chapman et al[5] cited precision and recall of 84.5% and 77.8% for negation detection, giving an F_1 measure of 0.81, although they used a different corpus of discharge summaries than the one used here for evaluation. A simplification of NegEx based on the presence of negating words within a flexible window of the target term was also proposed by Koeling et al.[8], although only negating words preceding the term were considered, whereas here we have used negating expressions both preceding and following the target term.

Instances where the simple negation detection patterns fail, however, include expressions such as '*gram negative bacteria*, and '*she didn't know why she is HIV positive*'. Future work might look at making the simple patterns a little more sophisticated without having to create a fully specified list of complete expressions.

In terms of automatically identifying temporal concepts and formalising them into TimeML expressions, there has been little previously reported research on evaluation of methods to achieve this, particularly in the clinical domain. Chang and Manning[9] have recently reported on SUTime, a rule-based system that is part of the Stanford CoreNLP framework[6] that using similar temporal morphemes and composition rules as those presented here as JAPE expressions. They reported precision, recall and F-measure scores of 0.88, 0.96 and 0.92 for extents and TimeML `Type` and `Value` F-measures of 0.96 and 0.82 against a corpus of general newswire texts derived from the TimeBank corpus[10]. Although nominally superior to the results presented here for the clinical discharge summaries, their evaluation corpora are not comparable to the discharge summaries used in this work. Also, it is not clear whether the authors report micro- or macro-averaged figures.

Also, as previously noted, working with patient notes presents particular difficulties for NLP applications, such as use of non-standard acronyms and abbreviated expressions. Although the performance of SUTime has not been formally evaluated against the i2b2 corpus, running the online demo[7] of SUTime against a small selection of discharge summaries suggests that it does not recognise abbreviated date expressions such as '*on 10/19*', abbreviated durations such as '*15 min*' nor dosage frequency abbreviations such as '*t.i.d*', and also annotates `Age` concepts (e.g. '*a 48 year old man*') as `Duration`.

Strötgen and Gertz[11] recently reported on HeidelTime, another rule-based temporal tagger developed to identify temporal expressions in four different domains: narrative, colloquial, news, and biomedical. In the biomedical corpus that they created for evaluation, they reported precision, recall and F-measure scores of 0.95, 0.66 and 0.78 for extents and TimeML `Value` F-measure of 0.70. Again, their evaluation corpus is not comparable to the one used in the present work, although it is at least in the same domain. Unlike SUTime, HeidelTime is available as a standalone

[6] http://nlp.stanford.edu/software/corenlp.shtml

[7] http://nlp.stanford.edu:8080/sutime/process

Java component, which should allow it to be integrated into the current pipeline via the GATE API. Future work could compare the performance of HeidelTime against that of the current component on the i2b2 corpus.

To identify the causes of the errors made by the GATE temporal expression component developed here, 20 documents with the lowest scores in precision, recall or `Value` F-measure were selected from the corpus and the discrepancies between the system output and gold standard analysed. Errors are summarised in Table 3, in the examples given, the left-hand-side expressions are from the gold standard.

Error type	Examples	n
Incorrect relative date `Value` calculation	postoperative day number two: 2009-08-26 *vs* 2009-08-19; hospital day # 1: 1992-09-21 vs 1992-09-22; the Sunday prior to admission: 2016-03-13 *vs* `<null>`	50
Missing abbreviated event date/duration or other temporal abbreviation	POD#6; last couple of days; on the 16th; stent [05-26]$_{\text{Date}}$; day of life #1; through [12-21]$_{\text{Date}}$	38
Missing 'orphaned' frequency, duration or time expressions	[five]$_{\text{Frequency}}$ grafts; days #5 and [6]$_{\text{Date}}$; [1]$_{\text{Time}}$ and 5 minutes; [four]$_{\text{Frequency}}$ past hospitalizations;	5
Incorrect type	for [ten days]$_{\text{Duration}}$ after discharge *vs* [ten days after discharge]$_{\text{Date}}$; for [4 days]$_{\text{Duration}}$ prior to discharge *vs* [4 days prior to discharge]$_{\text{Date}}$; [5 hours of life]$_{\text{Time}}$ *vs* [5 hours]$_{\text{Duration}}$ of life; [three days ago]$_{\text{Date}}$ *vs* [three days]$_{\text{Duration}}$ ago; [the three days]$_{\text{Duration}}$ prior to admission *vs* the [three days prior to admission]$_{\text{Date}}$	29

Table 3: Analysis of errors in temporal expression identification and formalisation.

As shown in Table 3, incorrect `Value` calculation of relative dates ($n = 50$) and abbreviated event dates and durations ($n = 38$) form the bulk of the errors identified in the 20 documents sampled. Problems with relative date value calculation stem from incorrect identification of the antecedent source date. Such dates are often not explicit in the document. For example, calculation of the correct value for '*postoperative day number two*' requires correct identification both of the surgical event (which may be expressed in many ways, such as '*went for surgery*', '*was transferred to the operating room*' etc), the date that this event occurred, and then recognition that '*postoperative*' refers to a time after this date.

In the present component, calculation of relative date values is limited to dates relative to the date of admission or discharge in the current component. For example, '*the next day*' in the absence of a prepositional attachment to an admission or discharge event, will, by default, be calculated as the day following admission. However, if the text has '*pt was sent to the ICU 20/12/2002. He received a dose of furosemide and was transferred to the ward the next day.*' then the relative date calculated will be incorrect. One potential solution would be to simply link relative dates back to the most recently mentioned date. However, if the most recently mentioned date is a historical episode, such as '*pt was diagnosed with CHF in 8/04*' then a later mention of '*the next day*' is more likely to be relative to some other date or period in the current episode than the immediately preceding, historical one. Clearly, identification and calculation of relative date values in clinical notes required more sophisticated handling than linking them to either the fixed dates (admission and discharge) or the most recently mentioned date – although this is also a weakness shared by other, general temporal expression parsers such as SUTime[9].

Difficulty distinguishing relative dates from durations was also a common problem ($n = 29$ in the 20 documents sampled). This was partly a result of inconsistent annotation in the gold standard: for example, instances of '*several months ago*' being annotated as an approximate `Date` (with features val="2004-06" mod="APPROX") vs '*three days ago*' being annotated as `Duration`. There may be some mileage making use of the preceding preposition to disambiguate the two, e.g. '*over the three days prior to admission*' would be a `Duration` but the same expression without the preceding preposition, '*three days prior to admission*', would be a `Date`.

A less frequently encountered error ($n = 5$) was the failure to pick up 'orphaned' temporal expressions, i.e. those linked by a conjunction to an earlier or later, more fully specified expression: for example '5' in '*day 4 and 5*' or a quantification of an event functioning as the frequency that that event occurred (e.g. 'five' in '*five previous operations*').

Conclusion

Lightweight components for identifying temporal expressions, clinical concepts, events, negation and possibility have been developed and evaluated against a corpus of discharge summaries. While the simple patterns for polarity and modality assignment perform well at the individual annotation level, the components perform poorly in terms of concept boundary detection and temporal value normalisation against the evaluation corpus. Development of these components was a useful exercise, but future work should focus on integrating existing tools for identification of clinical concepts, events and temporal expressions into the chosen information extraction framework, modifying the output of these for specific tasks on clinical notes.

References

1. Pustejovsky J, Castaño JM, Ingria R, Sauri R, Gaizauskas RJ, Setzer A et al. TimeML: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*, pages 28–34. AAAI Press, 2003.

2. 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'02)*, Philadelphia, 2002.

3. McCray AT, Burgun A and Bodenreider O. Aggregating UMLS semantic types for reducing conceptual complexity. *Stud Health Technol Inform*, 84(Pt 1):216–220, 2001.

4. Gooch P and Roudsari A. Lexical patterns, features and knowledge resources for coreference resolution in clinical notes. *J Biomed Inform*, Mar 2012.

5. 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(5):301–310, 2001.

6. Harkema H, Dowling JN, Thornblade T and Chapman WW. ConText: an algorithm for determining negation, experiencer, and temporal status from clinical reports. *J Biomed Inform*, 42(5):839–851, Oct 2009.

7. Agarwal S and Yu H. Detecting hedge cues and their scope in biomedical text with conditional random fields. *J Biomed Inform*, 43(6):953–961, Dec 2010.

8. Koeling R, Tate AR and Carroll JA. Automatically estimating the incidence of symptoms recorded in gp free text notes. In *Proceedings of the first international workshop on Managing interoperability and complexity in health systems*, MIXHS '11, pages 43–50, New York, NY, USA, 2011. ACM.

9. Chang AX and Manning C. Sutime: A library for recognizing and normalizing time expressions. In Calzolari Nicoletta, Choukri Khalid, Declerck Thierry, Doan Mehmet Uur, Maegaard Bente, Mariani Joseph et al., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

10. Pustejovsky J, Hanks P, Saurí R, See A, Gaizauskas R, Setzer A et al. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, Lancaster, March 2003.

11. Strötgen J and Gertz M. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eigth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753. ELRA, 2012. ISBN 978-2-9517408-7-7.