GATE-Time: Extraction of Temporal Expressions and Events

Leon Derczynski,[♡] **Jannik Strötgen,**[♠] **Diana Maynard,**[♡] **Mark A. Greenwood,**[♡] **Manuel Jung**[♣]

[♡] University of Sheffield; S1 4DP, UK

[♠] Max-Planck-Institut für Informatik; 66123 Saarbrücken, Germany

[♣] Universität Heidelberg; INF 348, 69120 Heidelberg, Germany

Abstract

GATE is a widely used open-source solution for text processing with a large user community. It contains components for several natural language processing tasks. However, temporal information extraction functionality within GATE has been rather limited so far, despite being a prerequisite for many application scenarios in the areas of natural language processing and information retrieval. This paper presents an integrated approach to temporal information processing. We take state-of-the-art tools in temporal expression and event recognition and bring them together to form an openly-available resource within the GATE infrastructure. GATE-Time provides annotation in the form of TimeML events and temporal expressions complying with this mature ISO standard for temporal semantic annotation of documents. Major advantages of GATE-Time are (i) that it relies on HeidelTime for temporal tagging, so that temporal expressions can be extracted and normalized in multiple languages and across different domains, (ii) it includes a modern, fast event recognition and classification tool, and (iii) that it can be combined with different linguistic pre-processing annotations, and is thus not bound to license restricted preprocessing components.

Keywords: Temporal tagging, Event extraction, GATE

1. Introduction

Time is critical to the meaning of language. Without it, we cannot discuss past events, change, or plans. Therefore, the annotation and extraction of temporal information from language is very important.

Much of the information and many of the assertions made in a text are bounded in time. For example, the sky was not always blue; George W. Bush's presidency was confined to an eight-year interval. An understanding of time in natural language text is critical to effective communication and analysis and must be accounted for in natural language processing and understanding.

Over recent years, research has been published in the area of temporal annotation resulting in several publicly available tools for temporal information extraction. Unfortunately, many of the tools rely on license-restricted linguistic preprocessing components.

In this paper, we present GATE-Time, an ISO-TimeML (Pustejovsky et al., 2010) approach to temporal information extraction in GATE (Cunningham et al., 2002) which brings together a number of these existing tools, as well as an upgraded approach to event annotation, within GATE. We use HeidelTime, a state-of-the-art, actively maintained, multilingual and domain-sensitive temporal expression tagger, breaking it out of its prepackaged pipeline and making the core component available in GATE. We present a novel event annotation tool, based on the earlier Evita system, using statistical learning with uneven margins; this yields state-of-the-art results.

As GATE-Time can make use of whatever linguistic preprocessing annotations are provided in GATE, it is not bound to license restricted preprocessing components. The overall approach to integration thus adds flexibility to existing and novel temporal information extraction tools and makes temporal annotation accessible to the large community of GATE users, which includes thousands of both academic and industrial sites worldwide.

2. Background & TimeML

In TimeML, representations of times in language are divided into two distinct categories: temporal expressions (TIMEX3s) and events (EVENTs). TimeML events are the lexicalisations of events and eventualities. Its temporal expressions represent time periods, to which events and other temporal expressions, or timexes, may be anchored. This paper focuses on the extraction of temporal entities, i.e., timexes and events (as opposed to relation extraction).

The TimeML temporal annotation problem can be thought of as the recognition of nodes and links in a temporal graph. Nodes correspond to mentions of events and times. Links between them can be different kinds of relations: temporal relations drawn from Allen's interval logic; subordination and modality; and aspectual links. This makes for a cross-lingual annotation scheme that can support reasoning, inference and information extraction. The scheme has been a great success, ported into many languages and used for a variety of applications (Derczynski et al., 2013).

Major progress in temporal annotation has been stimulated by the TempEval shared task. This comprises a subset of decomposed TimeML annotation tasks – for example, typing the temporal relations between events in the same sentence, or identifying temporal expression boundaries. The three main tasks have all attracted attention, with several temporal annotation tasks at SemEval'15.

With regard to integrated systems for temporal annotation: the earliest are Tango & Callisto (Verhagen et al., 2006); later, there was Tarsqi (Verhagen and Pustejovsky, 2012). Time and event annotation technology has moved on significantly since the last iteration of these, and so an update is required. Further, prior efforts are often standalone and sometimes closed-source tools, unlike GATE, an established open-source community tool maintained by a team of active developers. Our approach to extract events builds on the earlier approaches to replicating Evita in a statistical learning environment (Demidova et al., 2013).

3. Temporal Tagging

In this section, we discuss some existing tools for temporal tagging, explain the importance of performing domain-sensitive temporal tagging, motivate our choice of adding HeidelTime to GATE-Time, and present the new HeidelTime GATE wrapper.

3.1. Popular Tools for Temporal Tagging

For the task of temporal tagging, two popular, publicly available tools are SUTime (Chang and Manning, 2012) and HeidelTime (Strötgen and Gertz, 2013). At TempEval-3, SUTime achieved the best results for the extraction of temporal expressions in English, while HeidelTime won the full task of temporal tagging (correct extraction and normalization of temporal expressions) for English and Spanish (UzZaman et al., 2013). When processing news (and news-style) documents, SUTime and HeidelTime perform similarly. However, two major differences between them are that HeidelTime is multilingual, and that it applies different normalization strategies depending on the domain of the documents that are to be processed. This is paramount when processing narrative-style documents such as Wikipedia articles (Strötgen and Gertz, 2012), as will be detailed in the following section.

3.2. Domain-sensitive Temporal Tagging

In general, date and time expressions can be either explicit (e.g., *January 2016*), implicit (e.g., *Saint Patrick's Day 2016*), underspecified (e.g., *November*), or relative (e.g., *two weeks ago*). The difficulty of their normalization depends on the occurrence types of the expressions and the domain of the text in which they occur (Strötgen, 2015). Explicit expressions can be directly normalized with standard temporal knowledge (e.g., *2016-01* for *January 2016*) and implicit expressions with non-standard temporal knowledge such as information about holidays (e.g., *2016-03-17* for *Saint Patrick's Day 2016*). In contrast, the normalization of relative and underspecified expressions requires a reference time for the normalization, and a relation to the reference time for underspecified expressions, additionally.

A major difference between news- and narrative-style documents is that for the former, the document creation time plays an important role and can often be used as the reference time. In contrast, the latter are independent of the document creation time, i.e., it is typically not used as reference time for underspecified and relative expressions so that all temporal expressions can be interpreted correctly without considering the document creation time. Thus, depending on the type of the documents that are to be processed, a temporal tagger should apply different normalization strategies.

For instance, assuming the temporal expression *November* in a news-style document, it is quite likely that it refers to the November before or after the document creation time. In contrast, the reference time required to normalize the expression in a narrative-style document has to be usually determined in the document's text (e.g., a previously mentioned expression). A promising approach to determine the relation to the reference time in news-style documents is to use tense information, while a chronology assumption between the reference time and an underspecified expression in a narrative text is often valid (Strötgen, 2015).

	relaxed extr.			extr. & norm.
	p	r	f1	value f1
TE-3 platinum (news documents)				
SUTime*	89.4	91.3	90.3	67.4
HeidelTime 1.8*	93.1	87.7	90.3	77.6
HeidelTime 2.0	93.1	88.4	90.7	78.1
WikiWars (narrative Wikipedia documents)				
SUTime (new)	94.5	88.0	91.1	50.4
HeidelTime 2.0	98.3	86.1	91.8	83.1

Table 1: Comparing HeidelTime and SUTime performance on TempEval-3 platinum and WikiWars. On WikiWars, HeidelTime is used with its narrative-style normalization strategy.
* official results reported by (UzZaman et al., 2013).

3.3. Temporal Tagger Selection

To compare SUTime and HeidelTime, we performed an evaluation on two publicly available corpora, the news corpus TempEval-3 platinum (UzZaman et al., 2013) and the narrative-style corpus WikiWars (Mazur and Dale, 2010), which contains Wikipedia articles about important wars in history. The evaluation results shown in Table 1 demonstrate the importance of applying domain-dependent normalization strategies. HeidelTime uses domain-dependent strategies and achieves high-quality normalization results on both corpora, while SUTime applies the same normalization strategy independent of the domain of the documents and shows a significant decrease in the normalization quality on the narrative corpus.

Note that the WikiWars corpus contains annotations in TIMEX2 format and that both HeidelTime and SUTime extract temporal expressions following TimeML's TIMEX3 tags. However, a meaningful comparison is nevertheless possible as (i) many TIMEX values are identical in TIMEX2 and TIMEX3, (ii) some simple mappings from TIMEX3 to TIMEX2 are possible and we applied them for SUTime's and HeidelTime's output, and (iii) the other differences affect HeidelTime's and SUTime's evaluation results in the same manner. In addition to HeidelTime's domain-sensitivity and thus its higher normalization quality on narrative documents, a further advantage is that it is a multilingual tool. Thus, its integration into GATE is more valuable due to a higher number of application scenarios than integrating SUTime.

A further temporal tagger distinguishing between news- and narrative-style documents is DANTE (Mazur and Dale, 2009). However, DANTE extracts temporal expressions following the older TIDES TIMEX2 annotation guidelines, and modern efforts and evaluation tasks have focused on the different TIMEX3 standard for many years. Importantly, TIMEX2 is a standard for annotating temporal expressions only, whereas TimeML – which relies on TIMEX3 – allows annotation of many other temporal phenomena, including events, temporal signals, temporal relations, aspectual links and so on. Since GATE-Time should be suitable for the full task of temporal entity annotation, TimeML annotations are preferred, requiring TIMEX3. In addition, DANTE is a monolingual tool for processing English only. Thus, it is less suitable than HeidelTime.

One option for later work is to use a flexible, compositional temporal expression annotation system, i.e., separate tools for the extraction and the normalization of temporal expressions. This can prove especially useful for cases where rule-based interpretations cannot cover the given expression. Examples of such systems use context-free grammars (Bethard, 2013) or language-independent latent parses (Angeli and Uszkoreit, 2013). This line of research has begun to reconsider the TIMEX annotation standards (Bethard and Parker, 2016). However, the available systems – while achieving superior performance on some of the common news-style test sets – can be either slow or restricted to just English. So, for the scope of GATE-Time, we prefer a high-accuracy, high-speed temporal tagger that works across multiple languages and domains.

3.4. HeidelTime GATE Wrapper

The multilingual and domain-sensitive temporal tagger HeidelTime (Strötgen and Gertz, 2013) was initially developed within the UIMA framework (Ferrucci and Lally, 2004). It is already publicly available as a UIMA component and as a standalone Java version.[1] However, while some aspects of UIMA and GATE are similar – in particular the underlying pipeline principle – and indeed, elements of UIMA are available within GATE and vice versa, in practice combining elements from each framework is not always desirable or practical.

Since HeidelTime requires sentence, token, and part-of-speech annotations, the UIMA heideltime-kit contains components wrapping some preprocessing tools. Using the standalone version, linguistic preprocessing is performed directly. Thus, a simple solution to develop a HeidelTime GATE wrapper would have been to build a black box wrapper that calls HeidelTime with all its preprocessing capabilities (sentence splitting, tokenization, part-of-speech tagging). While this would result in HeidelTime annotations within a GATE pipeline, this would clearly harm the idea behind GATE's pipeline principle. Neither replacements of linguistic preprocessing components nor the use of gold standard annotations for sentence, token, and part-of-speech information would be possible. Thus, a solution allowing for flexible linguistic preprocessing is required. In addition, a further important requirement is that newly released HeidelTime versions can be easily used with the GATE wrapper.

Due to these requirements, the architecture of HeidelTime's GATE wrapper is as depicted in Figure 1. It can be used either with or without performing linguistic preprocessing, in the following way. If sentence, token, and part-of-speech annotations are available (i.e., if other tools such as GATE ANNIE are used in the pipeline to create them, or if gold standard annotations have been provided), the user has to specify the annotation names. Then, HeidelTime uses the provided annotations and just adds TIMEX3 annotations. Otherwise, if the user does not set these parameters, the pre-processing tasks are performed within the HeidelTime wrapper component of the GATE pipeline.

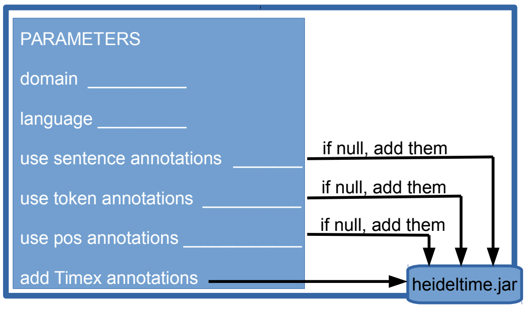


Figure 1: Architecture of the HeidelTime GATE wrapper

In addition to sentence, token, and part-of-speech annotations, HeidelTime requires further information to successfully extract and normalize temporal expressions: the language, the text type (e.g., news-style documents vs. narrative-style documents such as Wikipedia articles), and the date of publication if the domain is set to "news". When processing document collections – as is typical with GATE – all documents of a corpus are likely to be of the same language and domain type so that this information can be set as parameters in the pipeline. In contrast, the documents of document collection may have varying document creation times. As HeidelTime requires a document's creation time to normalize underspecified and relative expressions in news-style documents correctly, the HeidelTime GATE wrapper requires this meta information as a document-level annotation. For this, GATE-Time contains a component to parse standard formats of document creation times, the so-called DCTParser.

To ensure the high temporal tagging quality of HeidelTime within the GATE framework, we perform evaluations on several temporally annotated corpora to show that HeidelTime's evaluation results are consistent when using HeidelTime as a UIMA or GATE component. Details will be presented in Section 5.

4. Event Extraction

Along with entities, event recognition is one of the major tasks within Information Extraction, and has been successfully applied in research areas such as ontology generation, bioinformatics, news aggregation, business intelligence and text classification. Recognising events in these fields is generally carried out by means of pre-defined sets of relations, possibly structured into an ontology, which makes such tasks feasible but usually domain-dependent.

There are many possible definitions of events. We refer to an event as a situation within the domain (states, actions, processes, properties) expressed by one or more relations. These may be unique events such as the first landing on the moon or a natural disaster, or regularly occurring events such as elections or TV serials.

Events can be expressed by text elements such as:

- verbal predicates and their arguments (e.g. "The committee dismissed the proposal");

[1] https://github.com/HeidelTime/heideltime

- noun phrases headed by nominalizations (e.g. "economic growth");
- adjective-noun combinations (e.g. "governmental measure");
- event-referring nouns (e.g. "crisis", "cash injection").

Events can denote different levels of semantic granularity, i.e., general events can contain more specific sub-events. For instance, the performances of various bands form subevents of a wider music event, while a general event like "Turkey's EU negotiation has sub-events such as the European Parliament approving Turkey's Progress Report.

4.1. Popular Tools for Event Extraction

Prior work on event extraction encompasses both rule-based and statistical work, though there are not many examples of existing systems.

The EVITA system (Saurí et al., 2005) is a freely available tool for TimeML event recognition, and has achieved good results on the official task. It uses linguistic pre-processing and shallow syntactic information as features for machine learning. It requires a corpus annotated with tokens, sentences, POS tags, NP and VP chunks, possessive modifiers, and heads of noun phrases (NPs).

TIPSem (Llorens et al., 2010) takes a different tack, using semantic role labelling to identify temporal uses of language, and then building this into a structured learning approach to event recognition.

ATT1 (Jung and Stent, 2013) achieved the best scores in event extraction at TempEval-3. This system relies on both semantic and also syntactic information to perform event extraction, but it relies on lexical rather than semantic role features. It takes a sequence labelling approach to event extraction, using BIO tags, though it labels them using MaxEnt, a non-structured classifier with mild independence bias – quite different to the TIPSem approach.

We decided to use the Evita system as our starting point for event extraction, as it was most compatible with GATE. As with HeidelTime, we could theoretically have just written a GATE wrapper for Evita, but this would have been cumbersome and would not have enabled us to easily incorporate changes to Evita or to experiment with different pre-processing components, so we decided to reproduce Evita using GATE components.

4.2. Event Extraction in GATE

In this section, we describe the resources in GATE we have developed for annotating TimeML events. TimeML (Pustejovsky et al., 2003a) takes quite a broad view of events, defining them as situations that happen or occur, or elements describing states or circumstances in which something obtains or holds the truth (Llorens et al., 2013). These events are mostly, though not exclusively, expressed by verbs and nominalisations.

The event extraction component we have developed in GATE comprises two parts: first, the re-implementation of EVITA for verb-based events, and a component for extracting events described by nominalisations.

The event detection component consists of a combination of various approaches. The top-down approach involves a form of template filling, by selecting a number of known events in advance, and then identifying relevant verbs and their subjects and objects to match the slots. For example, a "performance" event might consist of a band name, a verb denoting some kind of "performing" verb, and optionally a date and location. This kind of approach tends to produce high precision but relatively low recall. We therefore supplement this with a bottom-up approach which consists of identifying verbal relations in the text, and classifying them into semantic categories, from which new events can be suggested. This kind of approach produces higher recall, but lower precision.

The event extraction method we adopt involves the recognition of entities and the relations between them in order to find domain-specific events and situations. In a semi-closed domain, this approach is preferable to an open IE-based approach which holds no preconceptions about the kinds of entities and relations possible.

The GATE application we have developed for event recognition is similar in structure to the one for entity recognition, and is designed to be run after the entity recognition application has first been run on the corpus. It therefore does not need to have components such as document and linguistic pre-processing, as these have already been run.

It consists of the following PRs:

- Event Gazetteer: this matches against some manually compiled lists of event-related words and terms, e.g. "financial crisis", "downturn", "strike" etc. This gives us some initial starting points on which the rules build.
- Verb Phrase Chunker: annotates base verb phrases (VPs) in the documents.
- Event Recognition Grammar: this is a set of hand-crafted JAPE grammars which makes use of the above information, combined with the entity information from the previous application, to identify potential events. We also go beyond TimeML and add context at this point. Any Date, Organisation or Location information contained within the span of the event in the text is added as features of the Event annotation: for example, if the event occurred in Athens in June 2010 then the normalised date is added as the value of a Date feature, and "Athens" is added as the value of a Location feature.

4.3. Recognising Verb-Based Events

For verb-based event recognition, we built an ML application designed for NER in GATE, which uses lemmatised tokens, POS tags and gazetteer lookup as input for machine learning, for which we used the PAUM algorithm (Li et al., 2005b). More specifically, the linguistic features used for the event recognition were domain-independent and include token kind (word, number, punctuation), lemma, POS tag, gazetteer class, and NP and VP chunking. Previous experimental results with NE detection have shown that even simple features alone such as token form, morphological feature and simple token types can achieve quite good results, and that using more sophisticated features such as POS tag and named entities from ANNIE or a gazetteer only obtain a small improvement (less than 2%).

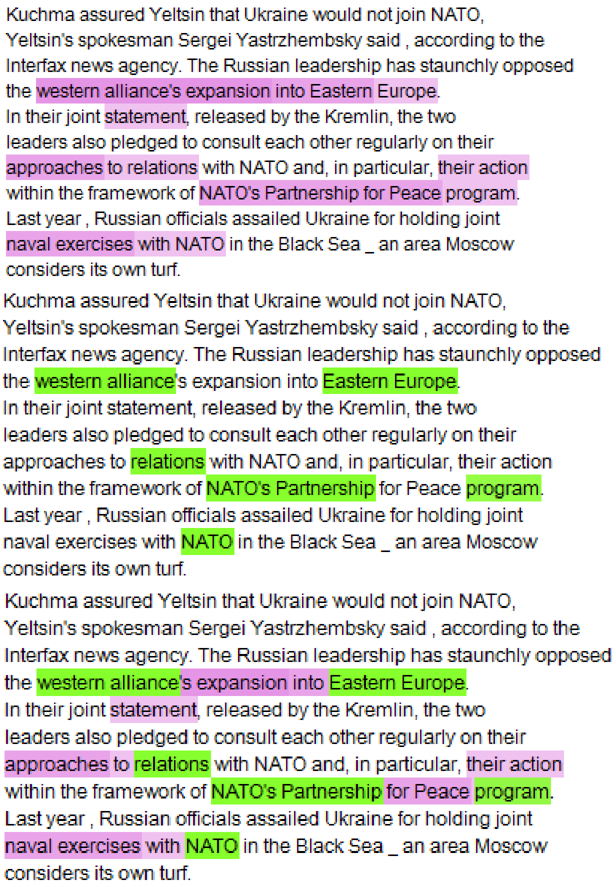
We first experimented with Evita by combining standard GATE pre-processing components with the existing Evita event extraction. Since Evita uses the same POS tagset (the Penn Treebank) as GATE's POS tagger, it is simple to preprocess the texts in GATE and then pass the resulting corpus to Evita. Preliminary experimentation on a small testset provided with Evita gave us results comparable to Evita's published results of around 86% accuracy for event recognition.

4.4. Applying PAUM algorithm to Event Recognition and Classification

The corpus was pre-processed to enable us to use a number of linguistic features, in addition to information already present in the document such as words and capitalisation.
Based on the linguistic information, an input vector is constructed for each token, as we iterate through the tokens in each document (including word, number, punctuation and other symbols) to see if the current token belongs to an information entity or not. Since in event recognition the context of the token is usually as important as the token itself, the features in the input vector come not only from the current token, but also from the preceding and following ones. As the input vector incorporates information from the context surrounding the current token, features from different tokens can be weighted differently, based on their position in the context. The weighting scheme we use is the reciprocal scheme, which weights the surrounding tokens reciprocally to the distance to the token in the centre of the context window. This reflects the intuition that the nearer a neighbouring token is, the more important it is for classifying the given token. Previous experiments have shown that such a weighting scheme typically obtains better results than the commonly used equal weighting of features (Li et al., 2005a).
In our experiments, the same number of left and right tokens was taken as a context window. The window size was set to 4, which means that the algorithm uses features derived from 9 tokens: the 4 preceding, the current, and the 4 following tokens. Due to the use of a context window, the input vector is the combination of the feature vector of the current token and those of its neighbouring tokens. We also experimented with the use of some additional features for event recognition, although these did not make a vast difference to the results.
In future work, we plan to experiment with combining our existing rule-based approaches (and the features created by them) with this ML approach. We also need to combine the methods for adding event participants (entities) to the event, and to decide which events are most relevant. One possible method for this is to weight the events based on where they occur in the document: for example, we would expect events mentioned in the title or first paragraph to be more crucial. We also plan to experiment with frequency-based methods.

4.5. Recognising Nominal events

This section describes the rule-based extraction of nominal events (events expressed by a noun phrase) exploiting semantic and linguistic resources.
The identification of different nominal events is reflected by five gazetteers in GATE, each specific to the resource they have been created from, and the type of nominal event they cover. These gazetteers annotate nominalizations and non-deverbal nominal events through lookup. In the derivational gazetteers, both WordNet's "event" and "state" subclassifications are preserved.



Kuchma assured Yeltsin that Ukraine would not join NATO, Yeltsin's spokesman Sergei Yastrzhembsky said , according to the Interfax news agency. The Russian leadership has staunchly opposed the western alliance's expansion into Eastern Europe.
In their joint statement, released by the Kremlin, the two leaders also pledged to consult each other regularly on their approaches to relations with NATO and, in particular, their action within the framework of NATO's Partnership for Peace program.
Last year , Russian officials assailed Ukraine for holding joint naval exercises with NATO in the Black Sea _ an area Moscow considers its own turf.

Kuchma assured Yeltsin that Ukraine would not join NATO, Yeltsin's spokesman Sergei Yastrzhembsky said , according to the Interfax news agency. The Russian leadership has staunchly opposed the western alliance's expansion into Eastern Europe.
In their joint statement, released by the Kremlin, the two leaders also pledged to consult each other regularly on their approaches to relations with NATO and, in particular, their action within the framework of NATO's Partnership for Peace program.
Last year , Russian officials assailed Ukraine for holding joint naval exercises with NATO in the Black Sea _ an area Moscow considers its own turf.

Kuchma assured Yeltsin that Ukraine would not join NATO, Yeltsin's spokesman Sergei Yastrzhembsky said , according to the Interfax news agency. The Russian leadership has staunchly opposed the western alliance's expansion into Eastern Europe.
In their joint statement, released by the Kremlin, the two leaders also pledged to consult each other regularly on their approaches to relations with NATO and, in particular, their action within the framework of NATO's Partnership for Peace program.
Last year , Russian officials assailed Ukraine for holding joint naval exercises with NATO in the Black Sea _ an area Moscow considers its own turf.

Figure 2: Nominal events in GATE-Time

For each event nominalization the following information is added in the gazetteers:

- Base verb e.g. value "complain" associated with "complaint"
- A syntactic subcategorization of these base verbs into transitive, intransitive, transitive/intransitive verbs and verbs which do not have any information in WordNet.
- Given the polysemy of the verbs involved, a small number have more than one syntactic subcategorization if this polysemy is reflected in different syntactic patterns.
- The exploitation of the linguistic resources yielded a gazetteer covering 4743 event nominalizations and 774 state nominalizations. The non-deverbal gazetteers contain 6528 and 745 entries for event and state hyponyms respectively.

In our pipeline, the function of these gazetteers is to bootstrap the event recognition process, and therefore they feature as the first modules in the GATE nominal event pipeline.
The screenshots in Figure 2 illustrate the results of nominal event extraction within the same text span. Nominal events are highlighted in purple, and their participants in green.

	relaxed extr.			extr. & norm.
corpus	p	r	f1	value F1
TE3 TimeBank	93.1	90.8	91.9	79.6
TE3 plat.	93.1	88.4	90.7	78.1
WikiWars	98.3	86.1	91.8	83.1

Table 2: Evaluation results of HeidelTime within GATE and UIMA are identical. We used HeidelTime 2.0 setting the domain to *news* for TimeBank and TempEval-3 platinum and to *narratives* when processing WikiWars.

	strict extr.			extr.&class.
corpus	p	r	f1	class F1
TE3 plat.	69.1	84.1	75.9	62.2

Table 3: Evaluation results of event recognition over the events in TempEval-3. The training data is the merged TBAQ dataset. Only strict evaluation is rational because TimeML events are only one token, and annotations are performed at token level.

5. Evaluation of GATE-Time

Here we present results of GATE-Time's temporal information extraction capabilities. We first evaluate HeidelTime as integrated in the GATE pipeline, and then present evaluation results of GATE-Time's event extraction component.

5.1. Temporal Tagging Evaluation

For evaluating the temporal tagging quality of HeidelTime as part of GATE-Time, we use the three corpora TimeBank[2] (Pustejovsky et al., 2003b), TempEval-3 platinum (UzZaman et al., 2013), and WikiWars (Mazur and Dale, 2010). Note that in contrast to the other two corpora, WikiWars contains narrative-style documents, namely Wikipedia documents about well-known wars in history. As HeidelTime supports domain-dependent normalization strategies, HeidelTime and thus GATE-Time can be used to temporally tag documents of different domains with high extraction and normalization quality.

Table 2 shows the evaluation results using precision, recall, and F1-score with relaxed (overlapping) matching for the extraction, and the value F1 measure for reflecting correct extraction (overlapping) and correct value normalization. Using these measures, we follow the TempEval-3 organizers who also stated that the value F1 measure with relaxed matching is the most important measure to evaluate temporal taggers.

5.2. Event Extraction Evaluation

Table 3 shows results using our event extraction. This is based on just one corpus, as there are fewer datasets annotated for events than for temporal expressions. For comparison, the best TempEval-3 system reached the higher precision of 81.4 but lower recall of 80.7. Class F1 is the F1-score in cases where not only are extents matched but also the TimeML event class matches. Unlike Evita and other prior approaches, this event extraction is independent of domain-specific or license-restricted tools such as TreeTagger, and instead relies on whatever preprocessing is selected in a given GATE application.

6. Conclusions

We have presented GATE-Time, an integrated, open-source toolkit for the annotation of times and events in ISO-TimeML. GATE-Time achieves state-of-the-art results, integrated in a large and popular framework, and is openly available for use in the community.

7. Acknowledgements

This work was partially funded by the EU's Seventh Framework Programme for research, under grants No. 270239 ARCOMEM, No. 611223 PHEME, and No. 610829 DECARBONET. This work also received funding under the uComp project by EPSRC EP/K017896/1, FWF 1097-N23 and ANR-12-CHRI-0003-03 in the framework of the CHIST-ERA ERA-NET program.

8. Bibliographical References

Angeli, G. and Uszkoreit, J. (2013). Language-independent discriminative parsing of temporal expressions. In *Proc. ACL*, volume 1, pages 83–92.

Bethard, S. and Parker, J. (2016). A Semantically Compositional Annotation Scheme for Time Normalization. In *Proc. LREC*.

Bethard, S. (2013). A synchronous context free grammar for time normalization. In *Proc. EMNLP*, pages 821–826.

Chang, A. X. and Manning, C. D. (2012). SUTime: A Library for Recognizing and Normalizing Time Expressions. In *Proc. LREC*, pages 3735–3740. ELRA.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: an architecture for development of robust HLT applications. In *Proc. ACL*, pages 168–175.

Demidova, E., Maynard, D., Tahmasebi, N., Stavrakas, Y., Plachouras, V., Dietze, S., Hare, J., Dupplaw, D., Funk, A., Peters, W., Papailiou, N., Barbieri, N., Tran, G. B., Risse, T., and Nunes, B. P. (2013). ARCOMEM Deliverable D3.3: Extraction and Enrichment.

Derczynski, L., Llorens, H., and UzZaman, N. (2013). TimeML-strict: clarifying temporal annotation. *arXiv preprint arXiv:1304.7289*.

Ferrucci, D. A. and Lally, A. (2004). UIMA: an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4):327–348.

Jung, H. and Stent, A. (2013). ATT1: Temporal annotation using big windows and rich syntactic and semantic features. In *Proc. SemEval*, pages 20–24.

Li, Y., Bontcheva, K., and Cunningham, H. (2005a). SVM based learning system for information extraction. In *Deterministic and statistical methods in machine learning*, pages 319–339. Springer.

Li, Y., Bontcheva, K., and Cunningham, H. (2005b). Using uneven margins SVM and perceptron for information extraction. In *Proc. CoNLL*, pages 72–79.

[2] We used the slightly improved TempEval-3 version of the TimeBank corpus (UzZaman et al., 2013).

Llorens, H., Saquete, E., and Navarro-Colorado, B. (2010). TimeML events recognition and classification: learning CRF models with semantic roles. In *Proc. COLING*, pages 725–733.

Llorens, H., Saquete, E., and Navarro-Colorado, B. (2013). Applying semantic knowledge to the automatic processing of temporal expressions and events in natural language. *Information Processing & Management*, 49(1):179–197.

Mazur, P. and Dale, R. (2009). The DANTE Temporal Expression Tagger. In *Proc. Language and Technology Conference*, pages 245–257. Springer.

Mazur, P. and Dale, R. (2010). WikiWars: A New Corpus for Research on Temporal Expressions. In *Proc. EMNLP*, pages 913–922.

Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., and Radev, D. R. (2003a). TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al. (2003b). The TimeBank corpus. In *Proc. Corpus Linguistics conference*, volume 2003, page 40.

Pustejovsky, J., Lee, K., Bunt, H., and Romary, L. (2010). ISO-TimeML: An International Standard for Semantic Annotation. In *Proc. LREC*.

Saurí, R., Knippen, R., Verhagen, M., and Pustejovsky, J. (2005). Evita: a robust event recognizer for QA systems. In *Proc. EMNLP*, pages 700–707.

Strötgen, J. and Gertz, M. (2012). Temporal Tagging on Different Domains: Challenges, Strategies, and Gold Standards. In *Proc. LREC*, pages 3746–3753. ELRA.

Strötgen, J. and Gertz, M. (2013). Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Strötgen, J. (2015). *Domain-sensitive Temporal Tagging for Event-centric Information Retrieval*. Ph.D. thesis, Institute of Computer Science, Heidelberg University.

UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J., and Pustejovsky, J. (2013). SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proc. SemEval*.

Verhagen, M. and Pustejovsky, J. (2012). The TARSQI Toolkit. In *Proc. LREC*, pages 2043–2048.

Verhagen, M., Knippen, R., Mani, I., and Pustejovsky, J. (2006). Annotation of temporal relations with Tango. In *Proc. LREC*.