The TempEval Challenge: Identifying Temporal Relations in Text

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz and James Pustejovsky
Brandeis University, University of Sheffield, and Thomson Reuters

Abstract. TempEval is a framework for evaluating systems that automatically annotate texts with temporal relations. It was created in the context of the SemEval 2007 workshop and uses the TimeML annotation language. The evaluation consists of three subtasks of temporal annotation: anchoring an event to a time expression in the same sentence, anchoring an event to the document creation time, and ordering main events in consecutive sentences. In this paper we describe the TempEval task and the systems that participated in the evaluation. In addition, we describe how further task decomposition can bring even more structure to the evaluation of temporal relations.

Keywords: TimeML, temporal annotation, temporal relations, information extraction, evaluation, corpus creation

1. Introduction

When the SenseEval workshop widened its scope and became SemEval-2007, temporal relation evaluation was added to the mix and TempEval was created as a new task. The ability to identify the events described in a text and locate these in time would sigificantly benefit a wide range of NLP applications, such as document summarization and question answering, and one of the main aims of TempEval was to aid research in this area by developing a common evaluation framework.

TempEval was conceived in the context of TimeML and TimeBank. TimeML is an ISO standard for annotation of events, temporal expressions and the anchoring and ordering relations between them (Pustejovsky et al., 2003a; Pustejovsky et al., 2005). TimeBank is a hand-annotated corpus conforming to the TimeML specifications (Pustejovsky et al., 2003b; Boguraev et al., 2007). TimeML and TimeBank have already been used as the basis for automatic time, event and temporal relation annotation tasks in a number of recent research projects (Mani et al., 2006; Boguraev and Ando, 2006; Chambers et al., 2007).

Evaluation of the complete temporal content of a document is a rather challenging task. It is not entirely clear how such an evaluation should proceed, given the many dependencies between temporal relations in a text. TempEval opted to take the first steps towards a

comprehensive evaluation and picked three limited subtasks of temporal annotation. The annotation study carried out for this evaluation still turned out to be more difficult than for other annotation tasks. This article discusses our findings from this study and proposes recommendations for future endeavors.

In this article, we first lay out the context in which TempEval originated (section 2). We describe the task and its participants in sections 3 and 4 and thoughts for future directions in section 5.

2. Annotating Times, Events and Temporal Relations

In this section we briefly review the TimeML annotation scheme, the target annotation scheme in terms of which the first TempEval challenge was defined. Our purpose is to provide enough detail about TimeML to contextualize the TempEval exercise and not to provide an exhaustive account of TimeML; for more complete accounts readers are referred to (Pustejovsky et al., 2003a; Pustejovsky et al., 2005).

TimeML addresses three key temporal elements: times, events and temporal relations, but its focus on temporal relations is what distinguishes it from other annotation efforts (see section 2.4). Reflecting this focus the following account concentrates on temporal relation annotation in TimeML; however as annotating temporal relations presupposes annotating times and events we begin with a short account of these.

2.1. Annotating Temporal Expressions

Perhaps the most obvious temporal features in natural language texts are temporal referring expressions, i.e., expressions referring to times (*five to eight*), dates (*July 1, 1867*), durations (*three months*), or frequencies or sets of regularly recurring times (*weekly*). Being able to identify and distinguish these types of expression is crucial to being able to situate the events described in text either absolutely, in terms of a calendrical time frame, or relatively, with respect to other events.

While these examples may seem straightforward, several further features of natural language time expressions make matters more complex. These include: indexicals (*yesterday, Wednesday*), which require context to fully interpret them; relational expressions, which specify a time in relation to another time or event (*the night before Christmas*, *two minutes after the explosion*); and vague expressions referring to times whose boundaries are inherently vague (*spring, evening*) or which contain modifiers which blur the time reference (*sometime after 7 p.m.*).

In TimeML, temporal referring expressions are annotated by enclosing them within a TIMEX3 XML tag. TIMEX3 tags have three primary

attributes: (1) TID – a unique id for this expression, serving as a "handle" for it in temporal relations; (2) TYPE – one of TIME, DATE, DURATION or SET, corresponding to the four types of temporal referring expressions discussed above; (3) VALUE – for time expressions of type time or date this is a fully interpreted or normalized time expression in ISO 8601 form; for durations it encodes the length and units of measure (e.g. P3D for *3 days*); for sets it works along with the attributes QUANT and/or FREQ to fully specify a set of times. In addition to these three core attributes other optional attributes are used to specify how indexical or relative temporal expressions are to be interpreted. See Pustejovsky et al. (2003a) for details.

2.2. Annotating Events

TimeML also provides guidelines for annotating linguistic expressions denoting events and some states. Such events and states (loosely referred to as "events" in TimeML) may be expressed by finite clauses, nonfinite clauses, nominalizations and event-referring nouns, adjectives and even some kinds of adverbial clauses – see (1) for examples.[1]

(1) a. When the Beagle *sailed* from Plymouth in December 1837 ...
b. *Sailing* for Madeira, Darwin became *seasick* ...
c. The *voyage* of the Beagle lasted almost five years ...
d. While *on board*, Darwin amused himself by ...

The italicized words in (1) are annotated in TimeML using the EVENT tag. Attributes attached to the EVENT tag are used to record further information relevant to the temporal anchoring or ordering of the EVENT, and to address some of the other complexities just mentioned. The EID attribute records a unique id for this expression, serving, as with the TID for TIMEX3's, as a handle for referencing this event in temporal relations. The CLASS attribute subcategorizes events into one of seven classes where members of each class have characteristic temporal properties or implications regarding events that may be subordinated to them. Classes include: PERCEPTION (*see*, *hear*), ASPECTUAL (*begin*, *continue*), I_ACTION – "intentional action" – (*try*, *prevent*), and OCCURRENCE – the default class – (*walk*, *sell*). Other attributes recording temporal information are (1) TENSE (2) ASPECT (3) MODALITY, whose value is the surface form of the modal verb to which the EVENT is subordinated, and (4) POLARITY, one of the two values POS or NEG.

[1] Event annotation is not as simple as annotating all expressions of the sort italicized in these examples, however. Negation and modal operators introduce another layer of complexity in the annotation process. For a full treatment of event annotation see Pustejovsky et al. (2003a).

2.3. Annotating relations between times and events

The primary aim of TimeML is not just the annotation of time and event expressions but the annotation of *relations* between events and times and events and other events. Such relations serve to anchor events in time and to order events temporally with respect to each other. Identifying these relations was the focus of the TempEval challenge.

Time-event relational information may be conveyed in various ways. The most explicit route is via a prepositional phrase in which a preposition signals a relation between a temporal referring expression and an event denoting expression, e.g., *John flew to Boston on Friday.* Another mechanism, one which avoids any explicit lexical signal of the relation, is through syntactic constructions such as nominal modifiers, e.g. *John's Friday flight to Boston*, or elliptical/reduced relatives, e.g. *John's flight, Friday at 5, will be crowded.* However, in many cases the relational information is derived by the reader using world or lexical semantic knowledge, or narrative convention and discourse interpretation.

(2) John arrived home at 9pm. He went to the kitchen, unpacked the groceries and cooked a light pasta dish.

In (2) we infer the going-to-the-kitchen event took place shortly after 9pm based not on any explicit signal of temporal relation, but on our world knowledge that kitchens are in homes and on the narrative convention of relaying events in sequence.

As with time-event relations, event-event temporal relations may be conveyed explicitly or implicitly. The chief mechanism for explicit relation is the temporal conjunction, typically used to relate the event expressed in a subordinated clause to one in a main clause; e.g., *After the game, John called Bob.* As with time-event relations, event-event temporal relations are frequently expressed implicitly, relying on world or lexical semantic knowledge, or narrative convention. So in (2) we know that the grocery unpacking took place after going-to-the-kitchen and the cooking after the unpacking because of our script-like knowledge of how these activites relate and sequential story-telling convention.

A question for the designers of any temporal relation annotation scheme is whether to annotate only explicitly signaled temporal relations or to annotate implicit relations as well. In TimeML the aim is to capture time-event and event-event temporal relations as completely as possible. Therefore TimeML proposes an approach to relational tagging that allows temporal relations to be marked between any pair of event-denoting expressions or between any time and event expressions, regardless of whether the relation is explicitly signaled or not.

Relation annotation is implemented via an XML element TLINK which consumes no text but links EVENT and TIMEX3 elements via their unique IDs and associates a relation type with the link. Information about both the relation type and the linked elements is recorded using attributes on the TLINK tag. The set of relation types employed in TimeML is based on the thirteen relations proposed by Allen in his interval algebra (Allen, 1983; Allen, 1984) and includes the six binary relations, BEFORE, IMMEDIATELY BEFORE, INCLUDES, BEGINS, ENDS, DURING, the six inverses of these, SIMULTANEOUS, and in addition IDENTITY, which holds for two events X and Y if their intervals are simultaneous and the events are identical.

Further information on the annotation of TLINKs can be found in Pustejovsky et al. (2003a). In addition they discuss at length the complexities of subordinated and aspectual contexts, which can have significant implications for the reality of embedded events, and how they are annotated in TimeML. However, as these complexities were ignored within TempEval we do not discuss them further here.

2.4. Related Work

Work to devise annotation schemes for temporal referring expressions began in earnest in MUC-6 (MUC-6, 1995) and was extended to cover relative as well as absolute time expressions in MUC-7 (MUC-7, 1998). These evaluations defined the TIMEX tag. Interpreting or evaluating these time expressions into a normalized ISO calendrical time form was introduced as a task within the TIDES program, which through its guidelines (Ferro et al., 2001) defined the TIMEX2 tag. Identifying and normalising temporal expressions according to the TIMEX2 guidelines for both English and Chinese texts became a challenge task in the Time Expression Recognition and Normalization (TERN) evaluation first held in 2004 [2] and repeated subsequently as part of the Automatic Content Extraction (ACE) program[3].

In the context of prior work on temporal information extraction, various approaches have been taken to the identification of events and their relation to times or other events. Filatova and Hovy (2001) treat each clause as an event and try to assign each a calendrical time-stamp. Schilder and Habel (2001) treat each verb and a specific set of nominals as event-denoting expressions and attempt to relate each such event expression to a time in cases where the relation is explicitly signaled or syntactically implicit. The ACE program specifies a small set of event types and participants must identify all mentions (whole sentences)

[2] See http://fofoca.mitre.org/tern.html.
[3] See http://www.nist.gov/speech/tests/ace/.

of these events types along with their arguments, one of which may be a TIMEX2 expression[4]. While few authors have considered event-event relations, notable exceptions are Li et al. (2005), Bramsen et al. (2006), Setzer and Gaizauskas (2000) and Katz and Arioso (2001). None of these efforts has been concerned with the development of an annotation scheme for marking up all event occurrences and temporal relations between events and times or other events.

Within the broader computational linguistics community there has been other work on semantic annotation that overlaps with efforts in TimeML. Both the PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998) projects aim to assign semantic roles to verbal arguments. While both have some concern with temporality – Propbank annotates temporal adjuncts of verbs with the ArgM-TMP tag and FrameNet has frame elements for time, duration and frequency – neither is concerned with anchoring or ordering the events to which the adjunct or frame elements pertain. Similarly Kim et al. (2008) describe the annotation of domain-specific event-signaling expressions and their arguments in the GENIA biomedical corpus. Only events in the domain ontology are annotated and while temporal terms in the context of the event expression are recorded, no use for them is proposed.

TimeML allows for a fairly fine-grained definition of temporal orderings, similar to the Allen relations, partly because many of these distinctions can be observed in language. However, past research has shown that a reduced set of temporal relations, some corresponding to disjunctions of the Allen relations, may be more appropriate for capturing temporal relations expressed in language (Schilder, 1997; Freksa, 1992). Restricting oneself to a limited set of so-called coarse relations also has computational advantages. Vilain et al. (1990) show, for example, that the convex relations algebra is a computationally tractable subset of Allen's interval calculus. Finally, since annotating temporal relations is very challenging, a reduced set of relations may be preferable for the purpose of reliably annotating temporal relations (Verhagen, 2005). In TempEval we have opted for such a reduced set of relations.

3. TempEval 2007: Design and Resources

TempEval was organized in the context of SemEval-2007, the international workshop on semantic evaluations, held in Prague, summer 2007. Open evaluation challenges have proved valuable in many areas of NLP, serving to drive forward research and technology development.

[4] See: http://www.nist.gov/speech/tests/ace/2007/doc/ace-evalplan.v1.3a.pdf.

In the area of automatic temporal annotation, previous shared task evaluations have addressed the tasks of identifying time expressions (MUC-6 and MUC-7) and of normalizing them with respect to the conventional calendrical timeline (TERN). None, however, had addressed the problem of establishing temporal relations, and TempEval was proposed specifically to bring the benefits of shared task evaluations to this area.

3.1. Tasks

In section 2, we have introduced and motivated the TimeML scheme for annotating temporal entities and relations. The automatic identification of *all* temporal entities and relations within a text is the ultimate aim of research in this area, and so one possibility for a shared task evaluation would be the automatic assignment of full TimeML annotations to texts, as represented in TimeBank. This aim, however, was judged to be too difficult for a first evaluation challenge, for participants developing systems within a limited timeframe, and potentially also for organizers in regard to the creation of training and test data. Instead, TempEval proposed three limited tasks in temporal relation assignment, which considerably simplified the ultimate aim of full temporal annotation, and yet which would have application potential if they could be performed automatically.

Several simplifications were made 'by design' in creating the tasks. First, it was decided that all events and temporal referring expressions, for which temporal relations would be required, would be pre-annotated in the training and test data provided. This was to allow participants to focus on the core concern of temporal relation assignment and to provide a 'level playing field' for evaluation so that observed performance differences for temporal relation recognition could not be blamed elsewhere, e.g. on differences in event recognition. Secondly, the full set of temporal relation types used in TimeML was reduced to a core set of basic relations (BEFORE, AFTER and OVERLAP, the latter encompassing all cases where intervals have a non-empty overlap), in order to reduce the set of discriminations to be made, and hence, in turn, the burden of providing data in which the required discriminations are adequately represented. Later, when data was being created, it was found to be useful to include three *additional* relations for cases that were ambiguous (BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER) or where no relation could be assigned (VAGUE). The reduced set of relations partially counteracts the data sparseness in the TimeBank corpus, where for some relation types only a few examples are available. Thirdly, the set of events for which temporal relation assignment would be required was

restricted down from the complete set of events that would appear in a full TimeML annotation (as might be found in TimeBank), although this restriction was done in different ways across the three tasks. For the first two tasks (A and B), a restricted set of events was identified, known as the the Event Target List or ETL. An event term was included in this list if there were at least twenty occurrences of terms having the same stem in TimeBank. For the third task, attention was restricted to the 'main event' of sentences, corresponding typically, but not always, to the syntactically dominant verb of the sentence. Given this background setting, the three tasks are defined as follows.

Task A. Assign the temporal relations holding between time and event expressions that occur *within the same sentence*. Only event expressions occuring in the ETL are considered. These events and all time expressions are annotated in the training and test data.

Task B. Assign the temporal relations holding between the Document Creation Time (DCT) and event expressions. Again only event expressions that occur in the ETL are considered, and these events and the time expressions are annotated in the data. For this task, the special TIMEX3 tag that refers to the document creation time (DCT) is interpreted as an interval that spans a whole day.

Task C. Assign the temporal relation holding between the *main* events of adjacent sentences. Events considered to be main events will be identified during data preparation and explicitly marked as such in the data.

It can be seen that Tasks A and B involve restricted cases of *temporal anchoring*, while Task C covers a limited case of *temporal ordering*. Note that for Task A, a relation label must be assigned for *all* possible pairings of relevant events and time expressions, a fact which produces a strong requirement for the relation label VAGUE, for use in the cases where a more contentful relation is unclear.

3.2. Data Resources

The data set used for training and evaluation was based on TimeBank version 1.2.[5]. In particular, the EVENT and TIMEX3 annotations were taken verbatim from there. The main difference with TimeBank is in

[5] The training set consisted of 162 documents and the evaluation set of 20 documents. TimeBank 1.2 is available for free from the Linguistic Data Consortium at `http://www.ldc.upenn.edu`. The TempEval corpus is available at `http://www.timeml.org`.

the TLINK tag. The TimeML relation types are a fine-grained set based on James Allen's interval logic, but, for TempEval , only the six relation types described above were used. The annotation procedure for TLINK tags involved dual annotation by seven annotators using a web-based interface. After this phase, three experienced annotators looked at all occurrences where two annotators differed as to what relation type to select and decided on the best option. For task C, main events were marked up in an extra annotation phase before TLINK annotation.

It should be noted that annotation of temporal relations is not an easy task for humans due to rampant temporal vagueness in natural language. As a result, inter-annotator agreement (IAA) scores are well below the recently suggested threshold of 90% (Palmer et al., 2005; Hovy et al., 2006). The inter-annotator agreement for the relation types of TimeML TLINKs in TimeBank was reported to be 77% with a Kappa score of 0.71. The numbers for TempEval are displayed in table I.[6]

Table I. Inter-annotator agreement on all tasks: precision and Kappa (lowest and highest refer to annotator pairings)

Task	Precision	Kappa	lowest	highest
A	69%	0.54	0.28	0.70
B	74%	0.54	0.27	0.76
C	65%	0.47	0.18	0.63

Closer observation of the Kappa scores showed that one annotator consistently generated the lowest Kappa scores in all three tasks. Removing this outlier gets average Kappa scores that are much closer to the highest score. It was expected that the TempEval IAA scores would be higher given the reduced set of relations, but the TempEval annotation task is more complicated in the sense that it did not allow annotators to ignore certain pairs of events, which made it impossible to skip hard-to-classify temporal relations. Note also that unweighted Kappa scores were computed which do not give any credit to agreement in terms of strict and coarse temporal relations (e.g., BEFORE vs. BEFORE-OR-OVERLAP).

[6] The scores were computed as micro-averages (i.e., averaged over all annotations rather than over documents). P-values < 0.0001 for all scores. See Cohen (1960) for details on the Kappa score. Note that since all annotators were presented with the identical instances to annotate precision and recall will be the same and in fact the same as simple accuracy.

We constructed a confusion matrix to examine disagreements within the initial dual annotation. The largest number of disagreements (53%) were between BEFORE and OVERLAP and between AFTER and OVERLAP. Also noticeable is the small number of cases tagged using the disjunctive relation labels BEFORE-OR-OVERLAP and OVERLAP-OR-AFTER. This is surprising as these labels were added to the task to facilitate relation type assignment in precisely the sort of cases where disagreement suggests the annotators are having difficulty. A further 19% of the disagreements involved one of the annotators assigning the VAGUE label, suggesting a non-trivial number of cases were indeed difficult for the annotators to judge. Around 9% of disagreements were between AFTER and BEFORE perhaps suggesting a simple confusion about which argument was which in the relation, rather than a difficulty in temporal interpretation. Finally it is worth noting that for the disjunctive relation types there was far more disagreement than agreement, calling into question the utility of these labels in a temporal relation annotation scheme. Further investigation is required.

3.3. Evaluation Measures

In full temporal annotation, evaluation of temporal annotation runs into the same issues as evaluation of anaphora chains: simple pairwise comparisons may not be the best way to evaluate. In temporal annotation, for example, one may wonder how the response *[A<B, A<C, B=C]* should be evaluated given the key *[A>B, A>C, B=C]*. Scoring this at 33% precision misses the interdependence between the temporal relations. What we need to compare is not individual judgements but two temporal graphs, as was previously argued by Setzer et al. (2006). For TempEval however, the tasks were defined in such a way that a simple pairwise comparison was possible since the aim was not to create a full temporal graph and judgements were made in isolation. TempEval used standard definitions of precision and recall:

$$Precision = R_c/R$$
$$Recall = R_c/K$$

Here, R_c is the number of correct answers in the response, R the total number of answers in the response, and K the total number of answers in the key. Note that when there is a fixed set of items to be classified (as for TempEval, where the data for each task identifies precisely the temporal entity pairs to which a relation label must be assigned), scores for precision, recall and F-measure should be identical, being the same as a simple accuracy score. We chose to use precision and recall as our measures, however, in recognition of the fact that participants might not want to be bound by a rigid requirement to label all and only a

fixed set of items. This supposition was correct, as evidenced by the system performance figures given later.

A complication arises with the disjunctive relations. How does one score the response BEFORE given the key BEFORE-OR-OVERLAP? TempEval uses two scoring schemes: strict and relaxed. The strict scoring scheme only counts exact matches as success. For the relaxed scoring scheme, a response is not simply counted as 1 (correct) or 0 (incorrect), but is assigned a value v where $0 \leqslant v \leqslant 1$. For exact matches, v is 1, but for partial matches a number between 0 and 1 is assigned. For example, if the response is BEFORE and the key is BEFORE-OR-OVERLAP, then v is 0.5. This scheme gives partial credit for disjunctions, but not so much that non-commitment edges out precise assignments. For example, assigning VAGUE as the relation type for every temporal relation results in a precision of 0.33. For more details on task definition, data collection and evaluation metrics see Verhagen et al. (2007).

4. TempEval 2007: Participating Systems and Results

Six teams participated in the TempEval tasks. In this section we provide a short description of each of the participating systems and also present the official scoring results.

4.1. Participants

4.1.1. *University of Colorado at Boulder (CU-TMP)*

The CU-TMP (Bethard and Martin, 2007) approach to the challenge used pairwise classification such that each event/time pair was assigned one of the TempEval relations. The pairs were encoded using syntactically and semantically motivated features that were then used to train support vector machine (SVM) classifiers. Preliminary results showed that the system for task B performed the best, so the result of this task was fed into the other tasks as a feature.

4.1.2. *Language Computer Corporation (LCC-TE)*

The LCC-TE team (Min et al., 2007) made use of NLP tools and linguistic resources already developed at LCC. Temporal relations are identified using both machine learning and rule-based approaches. The feature set used for machine learning consisted of four kinds of features. First-class features are those that were directly obtained from the TempEval data. Derived features are those that are derived based on the first-class features including tense and aspect shifts and whether a modal auxiliary is present. Extended features include semantic and syn-

tactic information provided by the LCC tools. Finally, merged features combine the output of one system with the features of another.

4.1.3. *Nara Institute of Science and Technology (NAIST)*
The NAIST.Japan system (Cheng et al., 2007) uses both a sequence labeling model and a dependency parse tree to identify TempEval relations. For the sequence labeling model, event/time pairs were ordered according to the position of the events and times in the document. For the machine learning-based component, dependency features were introduced such that each word was labeled according to its position in the tree relative to the event and time.

4.1.4. *University of Sheffield (USFD)*
The Sheffield system (Hepple et al., 2007) takes a straightforward classification approach to the TempEval tasks, using features taken either directly from the TempEval event/time annotations, or that can easily be computed from documents without any 'deep' NLP analysis. As such, the approach may be viewed as a 'shallow analysis baseline' against which to compare systems using deeper NLP analysis, such as syntactic analysis. Use of the WEKA ML workbench (Witten and Frank, 2005) to perform classification allowed easy experimentation with different ML algorithms, and the optimally performing one was chosen for each task (specifically `lazy.KStar`, `rules.DecisionTable` and `functions.SMO` (a version of SVM) for tasks A, B and C, respectively).

4.1.5. *Universities of Wolverhampton and Allicante (WVALI)*
The WVALI team (Puşcaşu, 2007) used their system TICTAC, which combines knowledge based and statistical methods. For example, intrasentential temporal relations were found using sentence-level syntactic trees and a bottom-up propagation of the temporal relations between syntactic constituents followed by a temporal reasoning mechanism that relates two targeted temporal entities to their closest ancestor and to themselves.

4.1.6. *XEROX Research Centre Europe (XRCE-T)*
The team from XEROX Research Center Europe (Hagège and Tannier, 2007) created a temporal processor, XTM, which is an extension of a rule-based in-house tool called XIP (Xerox Incremental Parser (Aït-Mokhtar et al., 2002). XRCE-T decided not to change their system's output to match the TempEval events and temporal expressions because that would require dramatic changes to their parser. In order to relate temporal expressions and events, the system begins by attaching any prepositional phrase, including temporal PPs, to the predicate

it modifies through a very general dependency link. Within a given sentence, the system can detect if events are temporally related and, if so, what kind of relationship that is.

4.2. Results

The results for the six teams are presented in Table II, which shows precision, recall and f-measure scores for both the strict and the relaxed scoring scheme (with precision/recall scores being suppressed when they are identical to the f-measure). The table also shows baseline performance figures (based on a simplistic classifier that always assigns the most common category), as well as averages and standard deviations for system scores.[7]

Table II. Results for Tasks A, B and C. Scores are percentages and have the form *strict score/relaxed score*. Precision and recall figures are omitted when they are identical to the F-measure.

	Task A			Task B			Task C
System	P	R	F	P	R	F	F
baseline			57/60			56/57	47/53
CU-TMP			61/63			75/76	54/58
LCC-TE	59/61	57/60	58/60	75/76	71/72	73/74	**55**/58
NAIST			61/63			75/76	49/53
USFD*			59/60			73/74	54/57
WVALI			**62/64**			**80/81**	54/**64**
XRCE-T	53/63	25/30	34/41	78/84	57/62	66/71	42/58
average	59/62	54/57	56/59	76/78	74/72	74/75	51/58
stddev	03/01	13/12	10/08	03/03	08/06	05/03	05/04

The differences between the systems are not large. The only system that stands out is WVALI for task B (strict and relaxed scoring) and task C (relaxed scoring). Interestingly, the baseline is close to the average system performance on task A, but for other tasks the system scores noticeably exceed the baseline. Note that the XRCE-T system is somewhat conservative in assigning TLINKS for tasks A and B, producing lower recall scores than other systems. For task A, this is

[7] The entry for USFD in the table is starred, as its developers were co-organizers of the TempEval task, although a strict separation was maintained at the site between people doing annotation work and those involved in system development.

mostly due to a decision only to assign a temporal relation between elements that can also be linked by the syntactic analyser.

To determine where system performances differ significantly we used the McNemar test, which assesses the likelihood that the observed disagreements between two systems could arise for systems that have the same error rate (Dietterich, 1998). For Task A, the only significant differences (p=.05) involve the XRCE-T system, which scores below the baseline and is significantly different from it and all other systems. For task B, however, there are significant differences between all systems and the baseline, except for XRCE-T which this time does not differ significantly from it.[8] In addition, for Task B, there are significant differences between XRCE-T and all other systems, and also between WVALI and all other systems except CU-TMP.[9] Finally, for Task C, like Task A, the only significant differences involve the XRCE-T system, whose performance does not differ significantly from the NAIST system or the baseline, but does differ signficantly from all other systems.

5. Temporal Evaluation and Task Decomposition

TempEval proposed a relatively simple way to evaluate systems that extract temporal relations. In this section we extend the task decomposition approach taken in TempEval and present a larger set of tasks where each task can be associated with its own guidelines, evaluation measure, data creation tools and even relation set.

One motivation for the evaluation approach of TempEval was to avoid the interdependencies that are inherent to a network of temporal relations, where relations in one part of the network may constrain relations in any other part of the network. TempEval deliberately focused on three subtasks of the larger problem of automatic temporal relation annotation and for each of these subtasks simple pairwise evaluation could be used. But task decomposition can also be of service to machine learning approaches. Some of the participating systems fed the results of task B as a feature into other tasks. With a large set of tasks, that approach will be potentially much more fruitful, especially if we can rank

[8] The lack of a significant difference for task B between XRCE-T and the baseline may appear puzzling, given the 10 point difference in f-measure. This is due to treating those tests instances to which XRCE-T did not assign a temporal relation as incorrect for purposes of the McNemar test (which requires a system response for each test instance). A similar move in calculating precision for the task would of course produce a lower f-measure score.

[9] The McNemar measure makes it possible for classifier 1 to differ significantly from classifier 2 and not from classifier 3 even if 2 and 3 have the same accuracy, as CU-TMP and NAIST do here, for instance.

the reliability of automatic taggers for each task. A final motivation is that task decomposition facilitates faster and more reliable creation of evaluation data because a specialized workflow with specialized tools and guidelines can be created.

In this section, we critique the TempEval tasks, present a new set of tasks and lay out how the test and evaluation corpus can be created using task decomposition and layered annotation. We conclude with some discussion on how the results of individual tasks can be combined into one consistent graph.

5.1. TempEval's Limitations

It was clear from the outset that the set of tasks chosen for TempEval was not complete, but merely a first step towards a fuller set of tasks. The main goal of the division in subtasks was to aid evaluation, but the larger goal of temporal annotation should not be forgotten, namely, to create a temporal characterization of a document that is as complete as possible. The three tasks of TempEval 2007 are not sufficient to create that characterization. Another problem is that the inter-annotator agreement scores are a bit outside the comfort zone. The scores reported in section 3.2 fall below widely accepted thresholds and raise some issues on how to interpret the system scores.

We believe that the experience with TempEval has shown that the methodology of splitting the temporal annotation task into sub tasks is sound, for the following reasons: (i) ease of evaluation, (ii) ease of data creation for sub tasks, (iii) ability to analyze a single task and propose enhancements, and (iv) ability to assign confidence measures to each task, enabling a greedy algorithm to merge data from all tasks.

Drawing on these reasons, we now introduce a more complete set of tasks and lay out how tasks can be created and how task results can be combined into one temporal graph. It should be pointed out that the following sections present a new research direction that is currently being used to create a much larger corpus with temporal annotation.

5.2. A Classification of Temporal Ordering Tasks

The proposed set of tasks is structured on the basis of both logical and linguistic considerations between candidate events and times in the corpus. Logical considerations pertain to the class of elements in temporal relations. This includes the TIMEX-EVENT distinction, but also the subtypes of those tags. Syntactic considerations include syntactic dominance, argument structure and discourse structure. The temporal linking tasks that we initially consider are given in table III.

Table III. Initial set of tasks

1	Anchoring a nominal event to a time expression in its immediate context: *the April blizzard*
2	Anchoring a verbal event to a time expression that is governed by the event (a temporal adjunct): *the game starts at 8pm*
3	Ordering consecutive events in a sentence: *he walked over thinking about the consequences*
4	Determining the temporal relation between two dates
5	Ordering events that occur in syntactic subordination relations: (a) event subject with governing verb event: *the massive explosion shook the building*, (b) verbal event with object event: *they observed the election*, (c) reporting event with subordinated event: *the witness said it happened too fast*, (c) perception event with subordinated event: *she heard an explosion*, (d) an intentional process or state with subordinated event: *I want to sleep for a week*
6	Ordering events in coordinations: *walking and talking*
7	Anchoring an event to the document creation time; a task that can be split up according to the event's class
8	Ordering two main non-reporting events in consecutive sentences: *John fell after the marathon. He got hurt.*
9	Ordering two arguments in a discourse relation: *I am resting because I just lifted a barrel of rum.*

Note that although this list is linguistically motivated, it is still defined in a top-down manner (glossing over for a moment that linguistic considerations are generally based on language data and therefore not completely top-down). A confrontation with data can ground a task and sharpen its definition. In addition, exploring a corpus with a task in mind is needed to figure out a fast way of creating task data. This is not an issue if the temporal annotation task is not split into sub tasks. In fact, creation of annotation data for a task is part and parcel of defining and creating a task.

5.3. Creating the Task Data

One assumption in defining and creating a task is that the task can be structurally defined. Typically, this means that a task can be associated with a set of syntactic and/or semantic patterns. For example, for task 2 in table III the following patterns can be isolated (the event and time expression in the construction are in italics, examples are from the Wall Street Journal corpus):

- PP inside VP with event verb:
  ... is scheduled VG[to *expire*] PP[at *the end of November*].
- PP attached to an S with event verb:
  ... and the company VG[will begin *mailing*] NP[materials] PP[to shareholders] PP[at *the end of this week*].
- Sentence-initial PP:
  In fiscal *1989*, Elco *earned* $7.8 million.

Looking at instances of a task that are not covered by these patterns can suggest additional patterns, and analysis of existing patterns may suggest that a task might be split into two or more tasks.[10] The guiding principles are that tasks should be easy to annotate (i.e., exhibit high inter-annotator agreement) and that there should be enough instances in a corpus to make the task relevant. For example, TempEval's task A has been replaced by tasks 1 and 2 from table III, these new tasks are more narrowly defined and indeed exhibit higher inter-annotator agreement.[11]

In our on-going work to create a larger corpus with temporal annotations, we have taken an approach similar to the "90% rule" used in OntoNotes (Hovy et al., 2006), where the observation of low agreement is taken to motivate a modification of the task, through the merging (or sometimes splitting) of senses. In our case, low agreement indicates the need to define tasks more narrowly.

Another way of changing the definition of a task is to structure the relation set in such a way that it maximizes annotator agreement for a task, using different relation sets depending on the task. For example, tasks where the data always provide an explicit temporal signal, as with some anchoring tasks (*I eat at 5pm*), can use the full set of TimeML relations, but other tasks, like ordering consecutive events, it may be advantageous to use a smaller set of vague relations. Obviously, we need a theory about what set of relations each task can draw upon. This theory would need to limit the disjunctive relations that can be used by proposing some kind of restriction on what disjunctions of basic relations are available, following earlier approaches by Vilain et al (1990), Freksa (1992) and Schilder (1997).

[10] Bethard et al. (2007) also suggest using syntactic patterns.

[11] An extreme version of task decomposition would be to annotate relations based on lemmas or pairs of lemmas. For example, we could annotate the orderings of all instances of *hear*. We have decided not to follow this approach for two reasons: (i) data sparseness makes it unlikely that there are enough occurrences for many verbs to actually see this as a task, (ii) we expect that many verbs exhibit similar ordering characteristics. We have considered splitting on classes of verbs and it is clear that further research is needed to establish what classes we can employ.

Temporal annotation tasks could also be defined using existing corpora that contain syntactic annotations or any other useful annotation. In many cases, tasks listed in table III can be defined using sets of patterns on the Penn Treebank or corpora build on top of it, like Propbank, Nombank and the Penn Discourse Treebank (Palmer et al., 2005; Meyers et al., 2004; Miltsakaki et al., 2004). In some cases, one annotation category from a corpus resource could actually be used to completely define a task. For example, task 2 could conceivably be defined by the ARG-TMP relation in PropBank. This kind of layered annotation can take much of the guesswork out of task data creation and speed it up significantly, since the syntactic and semantic patterns defining the task can be used to extract task data automatically.

5.4. Towards a Completely Annotated Graph

The goal of temporal tagging is to provide a temporal characterization of a set of events that is as complete as possible. If the annotation graph of a document is not completely connected then it is impossible to determine temporal relations between two arbitrary events, as these events could be in separate subgraphs. The tasks from section 5.2 produce the basic building blocks of such a complete characterization, but the results of separate tasks need to be put together with some care.

Results from all the tasks cannot simply be merged by taking the union of all relations. This is because the temporal relations proposed in separate tasks could clash with each other and there is no guarantee that adding all relations to the temporal graph would generate a consistent temporal network. This is not a problem inherent to task decomposition; any process of temporal annotation has to take into account that one set of temporal judgements puts constraints on other judgements. But task decomposition affords an elegant way to manage inconsistencies. Assume that we have precision numbers for each task and that we have ranked the tasks.[12] Resolving these inconsistencies is a rather complex manual task, but we can let the task precision scores drive a greedy algorithm that adds relations one by one, applying a constraint propagation algorithm at each step to ensure consistency.[13] First temporal relations for the task with the highest precision are added as constraints, followed by temporal relations from the next highest precision task, and so on. This allows higher-precision relations

[12] This works for both manually annotated data and results of automatic taggers. For manually annotated data we will take the results of adjudications, but assume that the inter-annotator agreement from the dual annotation phase is indicative of the precision. For automatic taggers we take the performance of the tagger on the task evaluation data.

[13] See Allen (1983) and Verhagen (2005) for details on the algorithm.

to take precedence over lower-precision relations elsewhere in the graph. The resulting graph is consistent and we know it was built using the highest precision temporal links that were available.

6. Conclusion

In this article, we have described TempEval, a framework for evaluating systems that automatically annotate texts with temporal relations. TempEval was the first major community effort of this type.

The annotation task was subdivided into three subtasks and these subtasks were defined in a way that precluded the complexity that emerges with a full temporal annotation task. But the tasks still proved difficult to carry out, as evidenced by relatively low inter-annotator agreement. In order to try to reduce the difficulty of the annotation task, a smaller set of ambiguous temporal relations was used. Whether this actually improved performance of the systems is unclear, and further research is necessary to answer this question.

Six different research groups partipated in the evaluation. While several different techniques were used, the performances of the systems were very similar; indeed in some cases they did not differ significantly from the baseline. Clearly there is substantial room for improvement and a thorough error analysis of the results would be very useful.

Based on our experience with this evaluation task, we suggest that the task decomposition approach be extended, thereby facilitating a more complete temporal annotation evaluation. We propose a set of further subtasks and discuss how the relational annotations produced from these subtasks might be combined to yield a more complete temporal graph. This methodology is currently being investigated in the NSF-funded Unified Linguistic Annotation project (`www.timeml.org/ula`) and the iARPA funded TARSQI project (`www.tarsqi.org`), as well as for TempEval-2, which has been accepted as a multi-lingual task for SemEval-2010.

It remains an open question whether it is possible or meaningful to have a single evaluation measure that purports to assess all temporal relations in a document. A weighted average of the results of all subtask evaluations could be a good start. However, the merging procedure in section 5.4 reintroduces some of the interdependencies that TempEval attempted to avoid. Some initial ideas on evaluating an entire graph (Ben Wellner, p.c.) include transforming the temporal graph of the document into a set of partial orders built around precedence and inclusion relations; these partial orders could then each be evaluated using an edit distance measure of some kind.

Acknowledgements

We would like to thank the organizers of SemEval 2007: Eneko Agirre, Lluís Màrquez and Richard Wicentowski. TempEval may not have happened without SemEval as a home. Thanks also to the members of the six teams that participated in the TempEval task: Steven Bethard, James Martin, Congmin Min, Munirathnam Srikanth, Abraham Fowler, Yuchang Cheng, Masayuki Asahara, Yuji Matsumoto, Andrea Setzer, Caroline Hagège, Xavier Tannier and Georgiana Puşcaşu. Additional help to prepare the data for the TempEval task came from Emma Barker, Yonit Boussany, Catherine Havasi, Emin Mimaroglu, Hongyuan Qiu, Anna Rumshisky, Roser Saurí and Amber Stubbs.

Part of the work in this paper was carried out in the context of the DTO/AQUAINT program and funded under grant number N61339-06-C-0140, and part was performed under the UK MRC-funded CLEF-Services grant ref: GO300607.

References

Aït-Mokhtar, S., J.-P. Chanod, and C. Roux: 2002, 'Robustness beyond shallowness: Increamental deep parsing'. *Natural Language Engineering* **8**, 121–144.

Allen, J.: 1983, 'Maintaining Knowledge about Temporal Intervals'. *Communications of the ACM* **26**(11), 832–843.

Allen, J.: 1984, 'Towards a General Theory of Action and Time'. *Artificial Intelligence* **23**, 123–154.

Baker, C., C. Fillmore, and J. Lowe: 1998, 'The Berkeley FrameNet Project'. In: *Joint 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conf. on Computation Linguistics (COLING-ACL'98)*. pp. 86–90.

Bethard, S. and J. H. Martin: 2007, 'CU-TMP: Temporal Relation Classification Using Syntactic and Semantic Features'. In: *Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pp. 129–132, Association for Computational Linguistics.

Bethard, S., J. H. Martin, and S. Klingenstein: 2007, 'Timelines from Text: Identification of Syntactic Temporal Relations'. In: *ICSC '07: Proc. of the Int. Conf. on Semantic Computing*. Washington, DC, USA, pp. 11–18, IEEE Computer Society.

Boguraev, B. and R. K. Ando: 2006, 'Analysis of TimeBank as a Resource for TimeML parsing'. In: *Language Resources and Evaluation Conf., LREC 2006*. Genoa, Italy.

Boguraev, B., J. Pustejovsky, R. Ando, and M. Verhagen: 2007, 'TimeBank evolution as a community resource for TimeML parsing'. *Language Resource and Evaluation* **41**(1), 91–115.

Bramsen, P., P. Deshpande, Y. Keok, and R. Barzilay: 2006, 'Inducting Temporal Graphs'. In: *Proc. of the 2006 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2006)*. pp. 189–198.

Chambers, N., S. Wang, and D. Jurafsky: 2007, 'Classifying Temporal Relations Between Events'. In: *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proc. of the Demo and Poster Sessions*. Prague, Czech Republic, pp. 173–176, Association for Computational Linguistics.

Cheng, Y., M. Asahara, and Y. Matsumoto: 2007, 'NAIST.Japan: Temporal Relation Identification Using Dependency Parsed Tree'. In: *Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pp. 245–248, Association for Computational Linguistics.

Cohen, J.: 1960, 'A Coefficient of Agreement for Nominal Scales'. *Educational and Psychological Measurement* **20**, 37–46.

Dietterich, T.: 1998, 'Approximate statistical tests for comparing supervised classification learning algorithms'. *Neural Computation* **10**(7), 1895–1923.

Ferro, L., I. Mani, B. Sundheim, and G. Wilson: 2001, 'TIDES Temporal Annotation Guidelines, version 1.0.2'. Technical report, The MITRE Corporation, McLean, Virginia. Report MTR 01W0000041.

Filatova, E. and E. Hovy: 2001, 'Assigning Time-Stamps to Event-Clauses'. In: *Proc. of the 2001 ACL Workshop on Temporal and Spatial Information Processing*.

Freksa, C.: 1992, 'Temporal Reasoning Based on Semi-Intervals'. *Artificial Intelligence* **54**(1), 199–227.

Hagège, C. and X. Tannier: 2007, 'XRCE-T: XIP Temporal Module for TempEval campaign.'. In: *Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pp. 492–495, Association for Computational Linguistics.

Hepple, M., A. Setzer, and R. Gaizauskas: 2007, 'USFD: Preliminary Exploration of Features and Classifiers for the TempEval-2007 Task'. In: *Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pp. 438–441, Association for Computational Linguistics.

Hovy, E., M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel: 2006, 'OntoNotes: The 90% Solution'. In: *Proc. of the Human Language Technology Conf. of the NAACL, Companion Volume: Short Papers*. New York City, USA, pp. 57–60, Association for Computational Linguistics.

Katz, G. and F. Arosio: 2001, 'The Annotation of Temporal Information in Natural Language Sentences'. In: *Proc. of ACL-EACL 2001, Workshop for Temporal and Spatial Information Processing*. Toulouse, France, pp. 104–111.

Kim, J.-D., T. Ohta, and J. Tsujii: 2008, 'Corpus annotation for mining biomedical events from literature'. *BMC Bioinformatics* **9**(10).

Li, W., K.-F. Wong, and C. Yuan: 2005, 'A Model for Processing Temporal References in Chinese'. In: *The Language of Time*. Oxford, UK: Oxford University Press.

Mani, I., B. Wellner, M. Verhagen, C. M. Lee, and J. Pustejovsky: 2006, 'Machine Learning of Temporal Relations'. In: *Proc. of the 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia.

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinkska, B. Young, and R. Grishman: 2004, 'The NomBank Project: An Interim Report'. In: *In Proc. of HLT-EACL Workshop: Frontiers in Corpus Annotation*.

Miltsakaki, E., R. Prasad, A. Joshi, and B. Webber: 2004, 'The Penn Discourse Treebank'. In: *In Proc. of 4th Int. Conf. on Language Resources and Evaluation (LREC 2004)*.

Min, C., M. Srikanth, and A. Fowler: 2007, 'LCC-TE: A Hybrid Approach to Temporal Relation Identification in News Text'. In: *Proc. of the Fourth Int. Workshop*

on Semantic Evaluations (SemEval-2007). Prague, Czech Republic, pp. 219–222, Association for Computational Linguistics.

MUC-6: 1995, 'Proc. of the Sixth Message Understanding Conf. (MUC-6)'. Defense Advanced Research Projects Agency, Morgan Kaufmann.

MUC-7: 1998, 'Proc. of the Seventh Message Understanding Conf. (MUC-7)'. Defense Advanced Research Projects Agency. Available at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/.

Palmer, M., D. Gildea, and P. Kingsbury: 2005, 'The Proposition Bank: an Annotated Corpus of Semantic Roles'. *Computational Linguistics* **31**(1), 00–00.

Puşcaşu, G.: 2007, 'WVALI: Temporal Relation Identification by Syntactico-Semantic Analysis'. In: *Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pp. 484–487, Association for Computational Linguistics.

Pustejovsky, J., J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, and G. Katz: 2003a, 'TimeML: Robust Specification of Event and Temporal Expressions in Text'. In: *Proc. of the Fifth Int. Workshop on Computational Semantics (IWCS-5)*. Tilburg.

Pustejovsky, J., P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo: 2003b, 'The TIMEBANK Corpus'. In: *Proc. of Corpus Linguistics 2003*. Lancaster, pp. 647–656.

Pustejovsky, J., R. Knippen, J. Littman, and R. Saurí: 2005, 'Temporal and Event Information in Natural Language Text'. *Language Resources and Evaluation* **39**, 123–164.

Schilder, F.: 1997, 'Temporal Relations in English and German Narrative Discourse'. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

Schilder, F. and C. Habel: 2001, 'From temporal expressions to temporal information: semantic tagging of news messages'. In: *Proc. of the ACL-2001 workshop on temporal and spatial information processing*. Toulouse, France, pp. 1–8, Association for Computational Linguistics.

Setzer, A. and R. Gaizauskas: 2000, 'Annotating Events and Temporal Information in Newswire Texts'. In: *LREC 2000*.

Setzer, A., R. Gaizauskas, and M. Hepple: 2006, 'The Role of Inference in the Temporal Annotation and Analysis of Text'. *Journal of Language Resources and Evaluation* **39**(2-3), 243–265.

Verhagen, M.: 2005, 'Temporal Closure in an Annotation Environment'. *Language Resources and Evaluation* **39**, 211–241.

Verhagen, M., R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky: 2007, 'SemEval-2007 Task 15: TempEval Temporal Relation Identification'. In: *Proc. of the Fourth Int. Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic, pp. 75–80, Association for Computational Linguistics.

Vilain, M., H. Kautz, and P. van Beek: 1990, 'Constraint propagation algorithms: A revised report'. In: D. S. Weld and J. de Kleer (eds.): *Qualitative Reasoning about Physical Systems*. San Mateo, California: Morgan Kaufman, pp. 373–381.

Witten, I. and E. Frank: 2005, *Data Mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann, second edition.