Classifying Temporal Relations with Simple Features

Paramita Mirza
Fondazione Bruno Kessler
and University of Trento
Trento, Italy
paramita@fbk.eu

Sara Tonelli
Fondazione Bruno Kessler
Trento, Italy
satonelli@fbk.eu

Abstract

Approaching temporal link labelling as a classification task has already been explored in several works. However, choosing the right feature vectors to build the classification model is still an open issue, especially for event-event classification, whose accuracy is still under 50%. We find that using a simple feature set results in a better performance than using more sophisticated features based on semantic role labelling and deep semantic parsing. We also investigate the impact of extracting new training instances using inverse relations and transitive closure, and gain insight into the impact of this bootstrapping methodology on classifying the full set of TempEval-3 relations.

1 Introduction

In recent years, temporal processing has gained increasing attention within the NLP community, in particular since TempEval evaluation campaigns have been organized on this topic (Verhagen et al., 2007; Verhagen et al., 2010; UzZaman et al., 2013). In particular, the classification of temporal relations holding between entities such as events and temporal expressions (timex) is crucial to build event timelines and to reconstruct the plot of a story. This could be exploited in decision support systems and document archiving applications, among others.

In this work we focus on the problem of classifying temporal relation types, assuming that the links between events and time expressions are already established. This task is part of TempEval-3 evaluation campaign, hence we follow the guidelines and the dataset provided by the organizers, so that we can compare our system with other systems participating in the challenge. Recent works have tried to address this complex classification task by using sophisticated features, based on deep parsing, semantic role labelling and discourse parsing (D’Souza and Ng, 2013; Laokulrat et al., 2013). We argue that a simpler approach, based on lexico-syntactic features, achieves comparable results, while reducing the processing time needed to extract the features. Besides, the performance of complex NLP tools may strongly vary when moving to new domains, affecting in turn the classification performance, while our approach is likely to be more stable across different domains.

Our features include some basic information on the position, the attributes and the PoS tags of events and timexes, as well as other information obtained from external lexical resources such as a list of typical event durations and a list of temporal signals. The few processing steps required include PoS-tagging, dependency parsing and the semantic tagging of connectives (based on the parser output).

We also investigate the impact of extending the number of training instances through inverse relations and transitive closure, which is a ‘simplified’ version of temporal closure covering only entities connected via the same relation type.

2 Related Work

The task we deal with in this paper was proposed as part of the TempEval-3 shared task (UzZaman et al., 2012). Compared to previous TempEval campaigns, the TempEval-3 task involved recognizing the full set of temporal relations in TimeML (14 types) instead of a reduced set, increasing the task complexity. This specific temporal relation classification task becomes the main focus of this paper.

Supervised classification of temporal relation types has already been explored in some earlier works. Mani et al. (2006) built a MaxEnt classifier to label the temporal links using training data.
which were bootstrapped by applying temporal closure. Chambers et al. (2007) focused on classifying the temporal relation type of event-event pairs using previously learned event attributes as features. However, both works use a reduced set of temporal relations, obtained by collapsing the relation types that inverse each other into a single type.

Our work is most similar to the recent work by D’Souza and Ng (2013). The authors perform the same task on the full set of temporal relations, but adopt a much more complex approach. They utilize lexical relations extracted from the Merriam-Webster dictionary and WordNet (Fellbaum, 1998), as well as semantic and discourse features. They also introduce 437 hand-coded rules to build a hybrid classification model.

Since we conduct our experiments based on TempEval-3 task setup, this work is also comparable with the systems participating in the task. UzZaman et al. (2013) report that three groups submitted at least one system run to the task. The best performing one (Laokulrat et al., 2013) uses, among others, sentence-level semantic information from a deep syntactic parser, namely predicate-argument structure features. Another system (Chambers, 2013) is composed of four MaxEnt classifiers, two of which have been trained for event-event links (inter- and intra-sentence) and two for event-time links. The third-ranked system (Kolya et al., 2013), instead, implements a much simpler set of features accounting for event tense, modality and aspect, event and timex context, etc.

3 Temporal Link Labelling

In this section we detail the task of temporal relation labelling, the features implemented in our classification system and the strategy adopted to bootstrap new training data.

3.1 Task description

The full set of temporal relations specified in TimeML version 1.2.1 (Sauri et al., 2006) contains 14 types of relations, as illustrated in Table 1. Among them there are six pairs of relations that inverse each other.

Note that according to TimeML 1.2.1 annotation guidelines, the difference between DURING and IS_INCLUDED (also their inverses) is that DURING relation is specified when an event persists throughout a temporal duration (e.g. John drove for 5 hours), while IS_INCLUDED relation is specified when an event happens within a temporal expression (e.g. John arrived on Tuesday).

Table 1: Temporal relations in TimeML annotation

| a | b | a is BEFORE b |
|---|---|---------------|
| b | a | a is BEFORE b |
| a | b | a IS_INCLUDED in b |
| b | a | b IS_INCLUDED in a |

In TimeML annotation, temporal links are used to (i) establish the temporal order of two events (event-event pair), (ii) anchor an event to a time expression (event-timex pair) and (iii) establish the temporal order of two time expressions (timex-timex pair).

The problem of determining the label of a given temporal link can be regarded as a classification problem. Given an ordered pair of entities \((e_1, e_2)\) that could be either event-event, event-timex or timex-timex pair, the classifier has to assign a certain label, namely one of the 14 temporal relation types. We train a classification model for each category of entity pair, as suggested in several previous works (Mani et al., 2006; Chambers, 2013).

However, because there are very few examples of timex-timex pairs in the training corpus, it is not possible to train the classification model for these particular pairs. Moreover, they only add up to 3.2% of the total number of extracted entity pairs; therefore, we decided to disregard these pairs.

3.2 Feature set

We implement a number of features for temporal relation classification. Some of them are basic ones which take into account morpho-syntactic information on events and time expressions, their textual context and their attributes. Others rely on semantic information such as typical event durations and connective type. However, we avoid complex processing of data. Such semantic information is based on external lists of lexical items.
and on the output of the addDiscourse tagger (Pitler and Nenkova, 2009).

Some features are computed independently based on either $e_1$ or $e_2$, while some others are pairwise features, which are computed based on both elements. Some pairwise features are only relevant for event-event pairs, for example, the information on discourse connectives and the binary features representing whether two events have the same event attributes or not. Similarly, the features related to time expression attributes are only relevant for event-timex pairs, since this information can only be obtained if $e_2$ is a time expression. The selection of features that contribute to the improvement of event-event and event-timex classification will be detailed in Section 4.3.

**String features.** The tokens and lemmas of $e_1$ and $e_2$.

**Grammatical features.** The part of speech (PoS) tags of $e_1$ and $e_2$, and a binary feature indicating whether $e_1$ and $e_2$ have the same PoS tag. The binary feature only applies to event-event pairs since we do not include the PoS tag of a time expression in the feature set of event-timex pairs. The grammatical information is obtained using the Stanford CoreNLP tool.\(^1\)

**Textual context.** The textual order, sentence distance and entity distance of $e_1$ and $e_2$. Textual order is the appearance order of $e_1$ and $e_2$ in the text, while sentence distance measures how far $e_1$ and $e_2$ are from each other in terms of sentences, i.e. 0 if they are in the same sentence. The entity distance is only measured if $e_1$ and $e_2$ are in the same sentence, and corresponds to the number of entities occurring between $e_1$ and $e_2$ (i.e. if they are adjacent, the distance is 0).

**Entity attributes.** Event attributes and time expression attributes of $e_1$ and $e_2$ as specified in TimeML annotation. Event attributes consist of class, tense, aspect and polarity, while the attributes of a time expression are its type, value and dct (indicating whether a time expression is the document creation time or not). Events falling under the category of noun, adjective and preposition do not have tense and aspect attributes in TimeML. We retrieve this information by extracting the tense and aspect of the verbs that govern them, based on their dependency relation. For event-event pairs we also include four binary features representing whether $e_1$ and $e_2$ have the same event attributes or not.

**Dependency relations.** Similar to D’Souza and Ng (2013), we use the information related to the dependency relation between $e_1$ and $e_2$. We include as features (i) the type of the dependency relation that exists between them, (ii) the dependency order which is either governor-dependent or dependent-governor and (iii) binary features indicating whether $e_1/e_2$ is the root of the sentence. This information is based on the collapsed representation of dependency relations provided by the parsing module of Stanford CoreNLP. Consider the sentence “John left the office and drove back home for 20 minutes”. Using the collapsed typed dependencies we could get the direct relations between the existing entities, which are conj and (left, drove) and prep for (drove, minutes).

**Event durations.** To our knowledge, we are the first to exploit event duration information as features for temporal relation classification. In fact, duration can be expressed not only by a predicate’s tense and aspect but also by its aktsinsart, i.e. the inherent temporal information connected to the meaning of a predicate. The typical event duration allows us to infer, for instance, that a punctual event is more likely to be contained in a durative one. If we consider the sentence “State-run television broadcast footage of Cuban exiles protesting in Miami”, this feature would tell us that broadcast lasts for hours while protesting lasts for days, thus contributing in determining the direction of DURING relation between the events.

The approximate duration for an event is obtained from the list of 1000 most frequent verbs and their duration distributions compiled by Gusev et al. (2011).\(^2\) The types of duration include seconds, minutes, hours, days, weeks, months, years and decades. We also add the duration difference between $e_1$ and $e_2$ as a feature.

---

\(^1\)http://nlp.stanford.edu/software/corenlp.shtml

\(^2\)The list is available at http://cs.stanford.edu/people/agusev/durations/
with the value varied between *same*, *less* or *more*. Similar to tense and aspect attributes for events, the duration of events under the category of noun, adjective and preposition are estimated by the governing verb. As for time expressions, their durations are estimated from their type and value attributes using a set of simple rules, e.g. the duration of *Thursday morning* (with the type of *TIME* and the value of *xxx-xx-xxTMO*) is hours.

**Temporal signals.** Derczynski and Gaizauskas (2012) show the importance of temporal signals in temporal link labelling. We take this into account by integrating in our features the list of signals extracted from TimeBank 1.2 corpus. We believe that the system performance will benefit from distinguishing between event-related signals and timex-related signals, therefore we manually split the signals into two separate lists. Signals such as *when*, *as* and *then* are commonly used to temporally connect events, while signals such as *at*, *for* and *within* more likely occur with time expressions. There are also signals that are used in both cases such as *before*, *after* and *until*, and those kind of signals are added to both lists.

Besides the signal token, the position of the signal with respect to the events or time expressions is also an important feature. Consider the position of a signal in the sentences (i) “John taught high school *before* he worked at a bank” and (ii) “*Before* John taught high school, he worked at a bank”, which is crucial to determine the order of John’s occupations. We also include in the feature set whether a signal occurs at the beginning of a sentence, as it is usually used to temporally relate events in different sentences, e.g. “John taught high school. *Previously*, he worked at a bank.”

**Temporal discourse connectives.** Consider the following sentences:

(i) “*John has been taking that driving course since the accident that took place last week.*”

(ii) “*John has been taking that driving course since he wants to drive better.*”

In order to label the temporal link holding between two events, it is important to know whether there are temporal connectives in the surrounding context, because they may contribute in identifying the relation type. For instance, it may be relevant to distinguish whether *since* is used as a temporal or a causal cue (example (i) and (ii) resp.). This information about discourse connectives is acquired using the *addDiscourse* tool (Pitler and Nenkova, 2009), which identifies connectives and assigns them to one of four semantic classes: *Temporal, Expansion, Contingency* and *Comparison*. Note that this is a much shallower approach than the one proposed by D’Souza and Ng (2013), who perform full discourse parsing.

We include as feature whether a discourse connective belonging to the *Temporal* class occurs in the textual context of *e*1 and *e*2. Similar to temporal signals, we also include in the feature set the position of the discourse connective with respect to the events.

### 3.3 Inverse Relations and Transitive Closure

Since Mani et al. (2006) demonstrate that bootstrapping training data through temporal closure results in quite significant improvements, we try to provide the classifier with more data to learn from using the inverse relations and closure-based inferred relations.

There are six pairs of relation types in TimeML that inverse each other (see Table 1). By switching the order of the entities in a given pair and labelling the pair with the inverse relation type, we basically multiply the number of training data.

As for temporal closure, there have been attempts to apply it to improve temporal relation classification. Mani et al. (2006) use SputLink (Verhagen, 2005), which was developed based on Allen’s closure inference (Allen, 1983), to infer the relations based on temporal closure. UzZaman and Allen (2011b) employ Timegraph (Gerevini et al., 1995) to implement the scorer for TempEval-3 evaluation, since precision and recall for temporal relation classification are computed based on the closure graph.

We use a simpler approach to obtain the closure graph of temporal relations, by applying the transitive closure only within the same relation type, e.g. *e*1 *BEFORE* *e*2 ∧ *e*2 *BEFORE* *e*3 → *e*1 *BEFORE* *e*3. It can be seen as partial temporal closure since it produces only a subset of the relations produced by temporal closure, which covers more complex cases, e.g. *e*1 *BEFORE* *e*2 ∧ *e*2 *INCLUDES* *e*3 → *e*1 *BEFORE* *e*3.
As shown in Fischer and Meyer (1971), the problem of finding the transitive closure of a directed acyclic graph can be reduced to a boolean matrix multiplication. For each temporal relation type, we build its boolean matrix with the size of $n \times n$, $n$ being the number of entities in a text. Given a temporal relation type $R$ and its boolean matrix $M$, the transitive closure-based relations of $R$ can be inferred from the matrix $M^2$ by extracting its non-zero elements.

4 Experiment Description

4.1 Dataset

Since we want to compare our work with existing approaches to temporal relation classification, we use the same training and test data as in TempEval-3 challenge\(^4\). Two types of training data were made available in the challenge: TBAQ-cleaned and TE3-Silver-data. The former includes a cleaned and improved version of the AQUAINT TimeML corpus, containing 73 news report documents, and the TimeBank corpus, with 183 news articles. TE3-Silver-data, instead, is a 600K word corpus annotated by the best performing systems at TempEval-2, which we do not use in our experiments.

Our test data is the newly created TempEval-3-platinum evaluation corpus that was annotated/reviewed by the Tempeval-3 task organizers. The distribution of the relation types in all previously mentioned datasets is shown in Table 2. We report also the statistics obtained after applying inverse relations and transitive closure, that increase the number of training instances.

It is worth noticing that DURING_INV relation does not exist in the training data but appears in the test data. In this case, inverse relations help in automatically acquiring training instances. The BEFORE relation corresponds to the majority class and makes the instance distribution quite unbalanced, especially in the TBAQ corpus. Finally, five event-timex instances in the TBAQ training data are labeled with IDENTITY relation and can be assumed to be falsely annotated.

4.2 Experimental Setup

We build our classification models using the Support Vector Machine (SVM) implementation pro-

vided by YamCha\(^5\). The experiment involves conducting 5-fold cross validation on the TimeBank corpus to find the best combination of features for the event-event and event-timex classifiers. We first run our experiments using YamCha default parameters (pairwise method for multi-class classification and polynomial kernel of degree 2). After identifying the best feature sets for the two classifiers, we evaluate them using different kernel degrees (from 1 to 4).

4.3 Feature Engineering

In order to select from our initial set of features only those that improve the accuracy of the event-event and event-timex classifiers, we incrementally add them to the baseline (the model with string feature only), and compute their contribution. Table 3 shows the results of this selection process, by including the average accuracy from the 5-fold cross validation.

In Table 3, we report the feature contributions of the entity attributes and dependency relations sets in more details, because within those categories only some of the features have a positive impact on accuracy. Instead, for features within textual context, signal and discourse categories, incrementally adding each feature results in increasing accuracy, therefore we report only the overall accuracy of the feature group. Similarly, for duration features, adding each feature incrementally results in decreasing accuracy.

Regarding entity attributes, it can be seen that aspect and class features have no positive impact on the accuracy of event-event classification, along with pairwise features same_class and same_polarity. As for event-timex classification, all event attributes except for polarity contribute to accuracy improvements. Among time expression attributes, only the information about whether a time expression is a document creation time or not (dct feature) helps improving the classifier.

The dependency_order feature does not give positive contribution to the accuracy in both cases. Besides, information on whether an event is the root of the sentence (dependency_is_root feature) is not relevant for event-timex classification.

Adding the temporal signal feature very slightly improves the accuracy of event-event classification, not as much as its contribution to event-timex
Table 2: The distribution of each relation type in the datasets for both event-event and event-timex pairs.

| Relation | event-event |          | event-timex |          |
|----------|-------------|----------|-------------|----------|
|          | train       | test     | train       | test     |
| TB       | TBAQ        | TBAQ-I   | TBAQ-IC     | TE3-P    |
| BEFORE   | 496         | 2,115    | 2,938       | 5,685    |
| AFTER    | 458         | 823      | 2,938       | 5,685    |
| IBEFORE  | 22          | 60       | 103         | 105      |
| IAFTER   | 27          | 43       | 103         | 105      |
| BEGINS   | 24          | 44       | 86          | 85       |
| BEGIN_BY | 24          | 42       | 86          | 85       |
| ENDS     | 12          | 17       | 79          | 79       |
| ENDED_BY | 44          | 62       | 79          | 79       |
| DURING   | 46          | 80       | 80          | 84       |
| DURING_INV| 0           | 0        | 80          | 84       |
| INCLUDES | 170         | 308      | 724         | 726      |
| IS_INCLUDED| 212        | 416      | 724         | 726      |
| SIMULTANEOUS | 456     | 519      | 519         | 518      |
| IDENTITIY | 534        | 742      | 742         | 742      |
| Total    | 2,519       | 5,271    | 9,281       | 27,828   |

The best classifier performances are achieved with the polynomial kernel of degree 4, both for event-event and event-timex classification. The accuracy for event-event classification is 43.69%, while for event-timex classification it is 66.62%. However, using a high polynomial kernel degree introduces more complexity in training the classification model, thus more time is required to train such models.

D’Souza and Ng (2013) evaluate their system on the same corpus, but with a slightly different setting. They also split TimeBank into 5 folds, but they only use two of them to perform 2-fold cross validation, while they use another part of the corpus to develop rules for their hybrid system. Their best configuration gives 46.8% accuracy for event-event classification and 65.4% accuracy for event-timex classification. Although the two approaches are not directly comparable, we can assume that the systems’ performance are likely to be very similar, with a better accuracy on event-event classification by D’Souza and Ng (2013) and a better performance on event-timex pairs by our system. Probably, the hybrid system by D’Souza and Ng, which integrates supervised classification and manual rules, performs better on event-event classification because it is a more complex task than event-timex classification, where simple lexical and syntactic features are still very effective.
Table 3: Feature contributions for event-event and event-timex classification. Features in italics have a negative impact on accuracy and are not included in the final feature set.

5 Evaluation

We perform two types of evaluation. In the first one, we evaluate the system performance with the best feature sets and the best parameter configuration using the four training sets presented in Table 2. Our test set is the TempEval-3-platinum corpus. The goal of this first evaluation is to specifically investigate the effect of enriching the training data with inverse relations and transitive closure.

We compute the system accuracy as the percentage of the correct labels out of all annotated links. In the second evaluation, we compare our system to the systems participating in the task on temporal relation classification at TempEval-3. The test set is again TempEval-3-platinum, i.e. the same one used in the competition. The task organizers introduced an evaluation metric (UzZaman and Allen, 2011b) capturing temporal awareness in terms of precision, recall and F1-score. To compute precision and recall, they verify the correctness of annotated temporal links using temporal closure, by checking the existence of the identified relations in the closure graph. In order to replicate this type of evaluation, we use the scorer made available to the task participants.

5.1 Evaluation of the Effects of Inverse Relations and Transitive Closure

Table 4 shows the classifiers’ accuracies achieved using different training sets. After performing a randomization test between the best performing classifier and the others, we notice that on event-event classification the improvement is significant ($p < 0.005$) only between TBAQ and TimeBank. This shows that only extending the TimeBank corpus by adding AQUAINT is beneficial. In all other cases, the differences are not significant. Applying inverse relations and transitive closure extends the number of training instances but makes the dataset more unbalanced, thus it does not result in a significant improvement.

Table 4: Classifier accuracies with different training data

This result is in contrast with the improvement brought about by temporal closure reported in Mani et al. (2006). The difference between our approach and Mani et al.’s is that (i) we apply only the transitive closure instead of the full temporal one, and (ii) our classification task includes 14 relations, while the other authors classify 6 relations. In our future work, we will investigate whether the benefits of closure are affected by the number of relations, or whether our simplified version is actually outperformed by the full one.

Furthermore, we plan to investigate the effect of over-sampling to handle highly skewed datasets, for instance by applying inverse relations and transitive closure only to minority classes.
5.2 Evaluation of the System Performance in TempEval-3 task

We train our classifiers for event-event pairs and event-timex pairs by exploiting the best feature combination and best configuration acquired from the experiment, and using the best reported dataset for each classifier as the training data. Even though it has been shown that inverse relations and transitive closure do not bring significantly positive impact to the accuracy, using the TBAQ-IC corpus as the training set for event-timex classification is still the best option. The two classifiers are part of a temporal classification system called TRelPro.

We compare in Table 5 the performance of TRelPro to the other systems participating in TempEval-3 task, according to the figures reported in (UzZaman et al., 2013). TRelPro is the best performing system both in terms of precision and of recall.

| System       | F1   | Precision | Recall |
|--------------|------|-----------|--------|
| TRelPro      | 58.48% | 58.30%    | 58.17% |
| UTTime-1, 4  | 56.45% | 55.58%    | 57.35% |
| UTTime-3, 5  | 54.70% | 53.85%    | 55.58% |
| UTTime-2     | 54.26% | 53.20%    | 55.36% |
| NavyTime-1   | 46.83% | 46.59%    | 47.07% |
| NavyTime-2   | 43.92% | 43.65%    | 44.20% |
| JU-CSE       | 34.77% | 35.07%    | 34.48% |

Table 5: Tempeval-3 evaluation on temporal relation classification

In order to analyze which are the most common errors made by TRelPro, we report in Table 6 the number of true positives (tp), false positives (fp) and false negatives (fn) scored by the system on each temporal relation. The system generally fails to recognize IBefore, Begins, Ends and During relations, along with their inverse relations, primarily because of the skewed distribution of instances in the training data, especially in comparison with the majority classes. This explains also the large number of false positives labelled for the Before class (event-event pairs) and for the Is Included class (event-timex pairs), which are the majority classes for the two pairs respectively.

| Relation      | event-event | event-timex |
|---------------|-------------|-------------|
| BEFORE        | 186         | 82          |
| AFTER         | 63          | 14          |
| IBefore       | 0           | 0           |
| IAFTER        | 0           | 0           |
| Begins        | 0           | 0           |
| Begin_By      | 0           | 0           |
| Ends          | 0           | 0           |
| End_by        | 1           | 0           |
| During        | 0           | 0           |
| During INV    | 0           | 0           |
| Includes      | 1           | 2           |
| Is Included   | 2           | 4           |
| Simultaneous  | 20          | 14          |
| Identity      | 9           | 33          |

Table 6: Relation type distribution for TempEval-3-platinum test data, annotated with TRelPro. The tp fields indicate the numbers of correctly annotated instances, while the fp/fn fields correspond to false positives/negatives.

In conclusion, we find that by using a simple feature set we can build a system that outperforms the systems built using more sophisticated features, based on semantic role labelling and deep semantic parsing. This may depend on the fact that more complex features are usually extracted from the output of NLP systems, whose performance impacts on the quality of such features.

We find that bootstrapping the training data with inverse relations and transitive closure does not help improving the classifiers’ performances significantly as it was reported in previous works, especially in event-event classification where the accuracy decreases instead. In the future, we will further investigate the reason of this difference. We will also explore other variants of closure, as well as over-sampling techniques to handle the highly skewed dataset introduced by closure.

Finally, the overall performance of our system, using the best models for both event-event and event-timex classification, outperforms the other systems participating in the TempEval-3 task. This confirms our intuition that using simple features and reducing the amount of complex semantic and discourse information is a valuable alternative to more sophisticated approaches.

Acknowledgments

The research leading to this paper was partially supported by the European Union’s 7th Framework Programme via the NewsReader Project (ICT-316404).
References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, November.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL ’07, pages 173–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nate Chambers. 2013. Navytime: Event and time ordering from raw text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 73–77, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Leon Derczynski and Robert J. Gaizauskas. 2012. Using Signals to Improve Automatic Classification of Temporal Relations. *CoRR*, abs/1203.5055.

Jennifer D’Souza and Vincent Ng. 2013. Classifying Temporal Relations with Rich Linguistic Knowledge. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 918–927.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Michael J. Fischer and Albert R. Meyer. 1971. Boolean matrix multiplication and transitive closure. In *SWAT (FOCS)*, pages 129–131. IEEE Computer Society.

Alfonso Gerevini, Lenhart Schubert, and Stephanie Schaeffer. 1995. The temporal reasoning tools Timegraph I-II. *International Journal of Artificial Intelligence Tools*, 4(1-2):281–299.

Andrey Gusev, Nathanael Chambers, Pranav Khaitan, Divye Khilnani, Steven Bethard, and Dan Jurafsky. 2011. Using query patterns to learn the duration of events. In *Proceedings of the Ninth International Conference on Computational Semantics, IWCS ’11*, pages 145–154, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anup Kumar Kolya, Amitava Kundu, Rajdeep Gupta, Asif Ekbal, and Sivaji Bandyopadhyay. 2013. Ju_cse: A crf based approach to annotation of temporal expression, event and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 64–72, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Utime: Temporal relation classification using deep syntactic features. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 88–92, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44*, pages 753–760, Stroudsburg, PA, USA. Association for Computational Linguistics.

Emily Pittler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort ’09, pages 13–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Roser Saurí, Jessica Litman, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. *TimeML Annotation Guidelines, Version 1.2.1*.

Naushad UzZaman and James Allen. 2011a. Temporal Evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 351–356, Portland, Oregon, USA, June. Association for Computational Linguistics.

Naushad UzZaman and James F. Allen. 2011b. Temporal evaluation. In *ACL (Short Papers)*, pages 351–356. The Association for Computer Linguistics.

Naushad UzZaman, Hector Llorens, James F. Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. TempEval-3: Evaluating Events, Time Expressions, and Temporal Relations. *CoRR*, abs/1206.5333.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval ’07*, pages 75–80, Stroudsburg, PA, USA. Association for Computational Linguistics.
Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval ’10, pages 57–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marc Verhagen. 2005. Temporal closure in an annotation environment. Language Resources and Evaluation, 39(2-3):211–241.