On the contribution of word embeddings to temporal relation classification

Paramita Mirza
Max Planck Institute for Informatics
Saarland Informatics Campus, Germany
paramita@mpi-inf.mpg.de

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

Abstract

Temporal relation classification is a challenging task, especially when there are no explicit markers to characterise the relation between temporal entities. This occurs frequently in inter-sentential relations, whose entities are not connected via direct syntactic relations making classification even more difficult. In these cases, resorting to features that focus on the semantic content of the event words may be very beneficial for inferring implicit relations. Specifically, while morpho-syntactic and context features are considered sufficient for classifying event-timex pairs, we believe that exploiting distributional semantic information about event words can benefit supervised classification of other types of pairs. In this work, we assess the impact of using word embeddings as features for event words in classifying temporal relations of event-event pairs and event-DCT (document creation time) pairs.

1 Introduction

The classification of temporal relations between events in text has been long studied and attacked from different perspectives in the NLP community. However, existing approaches heavily rely on information overtly expressed in text, such as explicit temporal markers (e.g. *before*, *during*), the tense, aspect and modality of event words, as well as specific syntactic constructions. In case overt indicators are missing, the task becomes significantly more challenging, as is often the case when two events take place in different sentences, or in anchoring an event to the document creation time (DCT). See for example the sentences in (i), where the label for the event pair (e_1, e_2) is BEFORE, and the sentence in (ii), where e INCLUDES the DCT.

(i) *When Wong Kwan* ***spent*** e_1 *seventy million dollars for this house, he thought it was a great deal. He* ***sold*** e_2 *the property to five buyers and said he'd double his money.*

(ii) *The U.N. Security Council on Aug. 6 ordered a global* ***embargo*** e *on trade with Iraq as punishment for seizing Kuwait.*

Inter-sentential event relations are quite frequent, covering for example 32.76% of the event pairs in the TempEval-3 evaluation corpus (UzZaman et al., 2013). Around 42.37% of pairs of an event and a time expression in the same corpus are actually pairs of an event and the DCT. Moreover, the TimeBank corpus contains 718 temporal relations which are co-ordinated by temporal signals, i.e., only 11.2% of all temporal links (Derczynski and Gaizauskas, 2013). These make research on implicit temporal ordering very relevant.

One common approach, first proposed in Marcu and Echihabi (2002), incorporates word-based information in the form of word pair feature vectors. Conventionally, a word is converted into a symbolic ID, which is then transformed into a feature vector using a *one-hot* representation: the feature vector has the same length as the size of the vocabulary, and only one dimension is on. From a machine learning point of view, this type of sparse representation makes parameter estimation extremely difficult and prone to over-fitting. It is also very challenging to achieve any interesting semantic generalization with this representation. Consider for instance, (*attack*, *injured*) that would be at equal distance from a synonymic pair (*raid*, *hurt*) and an antonymic pair (*died*, *shooting*).

Other approaches make use of semantic features extracted from external knowledge bases such as WordNet synsets (Fellbaum, 1998) and VerbOcean semantic relations between verbs (Chklovski and Pantel, 2004), capturing for instance that *marriage* happens-before *divorce*. Mirza and Tonelli (2014) exploit the list of event duration distribution from Gusev et al. (2011) for temporal relation classification, showing that it gives no benefit to classifier performance. The problem with such knowledge bases is that they have limited coverage, while approaches based on distributional semantics require no supervision and have a much better coverage.

The main goal of this work is to assess the contribution of dense vector representations of words and word pairs to temporal relation type classification, as detailed in Section 3. Specifically, we want to establish (i) which vector combination schemes are more suitable for classifying pairs of events, (ii) how well word embeddings can be used for this particular task compared to traditional features (Section 4.2), and finally, (iii) whether the combination of traditional features and word embeddings yields a better performance than using the two components in isolation. To the latter purpose, we compare vector concatenation and stacked learning (Section 4.3).

Experiments and evaluations are performed on the TimeBank-Dense corpus (Chambers et al., 2014), which was designed to address the sparsity issue in existing corpora with temporal annotation. We also compare our approach with CAEVO, a CAscading EVent Ordering system evaluated on the same corpus (Section 5).

2 Related Work

Many natural language processing applications such as information extraction (IE), question answering (QA), topic detection and tracking require understanding about temporally located events, i.e., to anchor events in time and order them. This temporal information is often modelled as a graph, with *times* and *events/states* as the nodes and *temporal relations* holding between them as the arcs. The details of how these three primitives are expressed in English, as well as their conceptual background (Allen, 1984; Moens and Steedman, 1987) have been discussed in Setzer (2001), and formalized in the TimeML annotation standard (Pustejovsky et al., 2003). In this work we focus on the task of ordering temporal entities, i.e., the classification of temporal relation types.

Current state-of the-art systems for temporal ordering resort to data-driven approaches (Bethard, 2013; Laokulrat et al., 2013; Mirza and Tonelli, 2014) or hybrid approaches combining rules and supervised classifiers (D'Souza and Ng, 2013; Chambers et al., 2014; Mirza and Tonelli, 2016). In building the classification models, most approaches rely primarily on morpho-syntactic features as well as lexical semantic information derived from WordNet synsets (Chambers et al., 2007; Laokulrat et al., 2013; Chambers et al., 2014) and VerbOcean semantic relations between verbs (Mani et al., 2006; D'Souza and Ng, 2013).

Other approaches exploit sentence-level semantic information, i.e. predicate-argument structure, as features for the classifiers (Llorens et al., 2010; Laokulrat et al., 2013; D'Souza and Ng, 2013). However, the evaluation results of TempEval-3 (UzZaman et al., 2013) show that a system with basic morpho-syntactic and lexical semantic features, such as ClearTK (Bethard, 2013), is hard to beat even if using more sophisticated semantic features. Indeed, ClearTK indirectly uses distributed lexical semantic features in the form of context (tokens appearing) between events.

As far as we know, there is no work on the task of ordering/anchoring temporal entities which specifically addresses the issue of implicit relations often recurring when two events are in different sentences, or when an event is related to the DCT. Such implicit relations are probably covered by hand-crafted rules or features based on the tense, aspect and modality of event words (Chambers et al., 2014), but sometimes such an overt indicator is lacking, as exemplified in previous examples (Section 1).

Most works on implicit discourse relations focused on the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), in which relations are annotated at the discourse level and organized into a three-level hierarchy. The top level relations, for example, include *Temporal*, *Contingency*, *Comparison* and *Expansion*. Braud and Denis (2015) presented a detailed comparative studies for assessing the benefit of unsupervised word representations, i.e. one-hot word pair representations against low-dimensional ones based on Brown cluster (Brown et al., 1992) and word embeddings, for identifying implicit discourse relations in PDTB. However, only the top level relations are considered, for instance, whether there exists a *Temporal* relation without investigating further into the more fine-grained temporal ordering.

3 Classifying Temporal Relations

Temporal relations, or temporal links, are annotations that connect markables bearing temporal information in a text, and express their temporal order. TimeML (Pustejovsky et al., 2003) is the widely known annotation framework for creating such representation that was used in the TempEval series, i.e., evaluation exercises focusing on temporal information processing. One of the main tasks in TempEval is temporal relation (TLINK) classification: given a pair of temporal entities (te_1 , te_2), namely events and time expressions (timex), determine their ordering relations (e.g. BEFORE, AFTER, INCLUDES, etc.).

Our goal is to compare classification performance on temporal relations using traditional features and word embeddings, which have recently shown to achieve good generalisation capabilities in several NLP tasks. Specifically, we want to evaluate whether, and in which configurations, they can contribute to advance state-of-the-art performance on temporal relation classification. To this purpose, we build and evaluate two different classifiers.

3.1 Temporal Relation Classification with Traditional Features

The first system is inspired by state-of-the-art approaches presented at TempEval-3 (UzZaman et al., 2013). Following UTTime (Laokulrat et al., 2013), we build three LIBLINEAR (Fan et al., 2008) classifiers (L2-regularized logistic regression): one for event-document creation time (E-D), one for event-timex (E-T) and one for event-event (E-E) edges. For timex-timex (T-T) relations, we implement a simple set of rules based on the values of time expressions, which proved to be effective for most T-T edges.

E-D, E-T and E-E Classifiers A set of features, listed in Table 1, is used for each type of edge, largely inspired by the best performing systems in TempEval-2 (Verhagen et al., 2010) and TempEval-3 (UzZaman et al., 2013) campaigns. We assume that pairs of temporal entities are given, and we rely on EVENT and TIMEX3 *attributes* in annotated TimeML documents, *morphosyntactic information* generated by MorphoPro (Pianta et al., 2008) and *dependency information* from the Mate tools (Bjorkelund et al., 2010).

Derczynski and Gaizauskas (2013) show the importance of *temporal signals* in temporal relation labelling, hence, we include also a similar set of features. However, we take the list of temporal signals from the TimeBank corpus, further expand it using the Paraphrase Database (Ganitkevitch et al., 2013), and manually cluster synonymous signals together, e.g. {*before*, *prior to*, *in advance of*}. The cluster ID is then included in the feature set instead of the signal text.

Note that the only *lexical semantic information* we include in the feature set is the Wordnet semantic similarity/relatedness (Lin, 1998) between event words. In order to have a feature vector of reasonable size, we simplify the possible values of some features during the one-hot encoding:

- *dependencyPath*. We only consider the existence of a dependency path between an E-E pair when it describes coordination, subordination, subject or object relation. For E-T pairs, we only consider the dependency path expressing temporal modification.
- *tempSignalCluster*. Given a temporal signal, we include the *clusterID* of the cluster containing synonymous signals, e.g. {*before*, *prior to*, *in advance of*} instead of the signal text.
- *wnSim*. The value of WordNet similarity measure is discretized as follows: $sim \leq 0.0$, $0.0 < sim \leq 0.5$, $0.5 < sim \leq 1.0$ and $sim > 1.0$.

T-T Rules Only temporal expressions of types DATE and TIME are considered in the hand-crafted set of rules, based on their *normalized values*. For example, *7 PM tonight* with `2015-12-12T19:00` as value IS_INCLUDED in *today* with `2015-12-12` as value.

Feature		TLINK		Rep.	Description
	E-D	E-T	E-E		
EVENT attributes					
class	x	x	x	one-hot	EVENT attributes as specified in TimeML.
tense	x	x	x	one-hot	
aspect	x	x	x	one-hot	
polarity	x	x	x	one-hot	
sameClass			x	binary	Whether e_1 and e_2 have the same EVENT attributes.
sameTenseAspect			x	binary	
samePolarity			x	binary	
TIMEX3 attributes					
type	x	x		one-hot	TIMEX3 attributes as specified in TimeML.
Morphosyntactic information					
PoS	x	x	x	one-hot	Part-of-speech tags of e_1 and e_2.
phraseChunk	x	x	x	one-hot	Shallow phrase chunk of e_1 and e_2.
samePoS		x	x	binary	Whether e_1 and e_2 have the same PoS.
Textual context					
entityOrder		x		binary	Appearance order of e_1 and e_2 in the text.[1]
sentenceDistance		x	x	binary	`0` if e_1 and e_2 are in the same sentence, `1` otherwise.
entityDistance		x	x	binary	`0` if e_1 and e_2 are adjacent, `1` otherwise.
Dependency information					
dependencyPath			x	one-hot	Dependency path between e_1 and e_2.
isMainVerb	x	x	x	binary	Whether e_1/e_2 is the main verb of the sentence.
hasModalVerb	x	x	x	binary	Whether e_1/e_2 is governed by a modal verb.
Temporal signals					
tempSignalCluster		x	x	one-hot	Cluster ID of temporal signal existing around e_1 and e_2.
tempSignalPosition		x	x	one-hot	Temporal signal position w.r.t e_1/e_2 (BETWEEN, BEFORE, BEGIN, etc.)
tempSignalDependency		x	x	one-hot	Temporal signal dependency path between signal tokens and e_1/e_2.
lexical semantic information					
wnSim			x	one-hot	WordNet similarity computed between the lemmas of e_1 and e_2.

Table 1: Feature set for TLINK classification model for event-document creation time (E-D), event-timex (E-T) and event-event (E-E) pairs, along with representation type (Rep.) and brief description.

3.2 Temporal Relation Classification with Word Embeddings

Recently there has been an increasing interest in using word embeddings as an alternative source of information to traditional hand-crafted features. Word embeddings represent (embed) the semantics of a word in a continuous vector space, where semantically similar words are mapped to nearby points. The underlying principle is the *Distributional Hypothesis* (Harris, 1954), which states that words which are similar in meaning occur in similar contexts.

Baroni et al. (2014) divide approaches based on this principle into two categories: (i) *count-based* models and (ii) *predictive* models. They also provide a systematic comparison of word vectors from the two models, on a wide range of lexical semantic tasks, including semantic relatedness, synonym detection, concept categorization, selectional preferences and analogy. The main takeaway is that predictive models, such as Word2Vec (Mikolov et al., 2013), are shown to perform better than count-based ones.

Levy et al. (2015) reveal that much of the performance gains of word embeddings are due to hyperparameter optimizations rather than the embedding algorithms themselves, thus refuting the claim that prediction-based methods are superior to count-based approaches. However, they also state that the Skip-Gram model with Negative Sampling (SGNS), which is used to build Word2Vec pre-trained word vectors, can be a robust baseline since it does not significantly underperform in any scenario.

In this work, we explore how well word embeddings can be used—as lexical semantic features—to capture the temporal order of events (e.g. *attack* often happens BEFORE *injured*) and the temporal anchoring of an event to the document creation time (e.g. *embargo* usually spans longer than a day, hence, INCLUDES the DCT). Again, we build LIBLINEAR (Fan et al., 2008) classifiers (L2-regularized logistic regression), one for E-D and another for E-E pairs. Instead of the traditional feature sets explained in Section 3.1, word embeddings are used as feature vectors.

[1] The order of e_1 and e_2 in E-E pairs is always according to the appearance order in the text, while in E-T pairs, e_2 is always a timex regardless of the appearance order.

Pre-trained word vectors We take pre-trained word vectors from Word2Vec[2], which are 300-dimensional vectors for 3 million words and phrases trained on part of Google News dataset (about 100 billion words). Given an E-E pair (e_1, e_2), we retrieve the pair of word vectors $(\vec{w}_1, \vec{w}_2)$ based on vector look-up for the head words of e_1 and e_2 in the pre-trained word vectors. Meanwhile, for an E-D pair (e, t) we retrieve word vectors $\vec{w}$ according to the head word of e.

Vector combinations For E-E pairs, we test three different strategies in combining the word vectors of a pair of events: we consider (i) *concatenation* $(\vec{w}_1 \oplus \vec{w}_2)$, (ii) *addition* $(\vec{w}_1 + \vec{w}_2)$ and (iii) *subtraction* $(\vec{w}_2 - \vec{w}_1)$, as vector combination schemes. Note that in (i) the word ordering information is retained, which is not the case in (ii) and (iii).

We only consider word embeddings for events, specifically their head words, because the embeddings for all events annotated in the dataset, which are mostly verbs and nouns, are readily obtainable from the pre-trained word vectors. Meanwhile, representing time expressions with single word vectors is non-trivial, since most of them express dates (e.g. *Friday the 13th*) and times (e.g. *half past ten*), which are usually multi-word expressions.

4 Experimental Setup and Evaluation

We present two sets of experiments: first, we investigate how well word embeddings can be used for temporal relation classification compared with traditional features, and then we analyse whether the combination of these two types of features is beneficial.

4.1 Dataset: TimeBank-Dense

We evaluate the temporal classifiers using the TimeBank-Dense corpus (Chambers et al., 2014), which was created to address the sparsity issue in existing TimeML corpora. Using a specialized annotation tool, annotators were prompted to label all pairs of events and time expressions in the same sentence, all pairs of events and time expressions in two adjacent sentences, and all pairs of events and document creation time. This solution was introduced to solve the problem of sparse annotation of temporal relations in the TempEval-3 evaluation corpus, which made it difficult to evaluate and compare different systems.

The VAGUE relation introduced at the first TempEval task (Verhagen et al., 2007) was also adopted in TimeBank-Dense to cope with ambiguous temporal relations, or to indicate pairs for which no clear temporal relation exists. The resulting corpus contains 12,715 temporal relations under 6 labels, i.e. BEFORE, AFTER, INCLUDES, IS_INCLUDED, SIMULTANEOUS and VAGUE, over 36 documents taken from TimeBank. Annotation here is much denser than in the TimeBank corpus, which contains 6,418 temporal relations under 14 labels over 183 documents.

We follow the experimental setup in Chambers et al. (2014), in which the TimeBank-Dense corpus is split into a 22 document training set, a 5 document development set and a 9 document test set.[3] All the classification models are trained using the training set, as well as the rule set development for T-T edges. We evaluate our classification performances on (i) stratified 10-fold cross validation over the training set and (ii) on the test set.

4.2 Experiment 1: Comparing Traditional Features vs. Word Embeddings

In Table 2 we report the performances (micro-averaged F1-scores) of each classifier using word vectors $\vec{w}$ as features, compared with the classifier performance using traditional features $\vec{f}$, evaluated on stratified 10-fold cross-validation. For E-E pairs, we also report the F1-scores for each vector combination scheme. Since we classify all possible event pairs in the dataset, precision and recall are the same.

From the different vector combinations, concatenation $(\vec{w}_1 \oplus \vec{w}_2)$ is shown to be the best combination. Using the concatenated Word2Vec embeddings $(\vec{w}_1 \oplus \vec{w}_2)$ as features results in .605 F1-score, significantly better than using only traditional features (.529 F1-score). The fact that this representation retain the word order information may be the reason why it beats the other vector combinations.

[2] `http://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/`

[3] Available at `http://www.usna.edu/Users/cs/nchamber/caevo/`.

TLINK type	**E-D**		**E-E**			
	$\vec{f}$	$\vec{w}$	$\vec{f}$	$(\vec{w}_1 \oplus \vec{w}_2)$	$(\vec{w}_1 + \vec{w}_2)$	$(\vec{w}_2 - \vec{w}_1)$
BEFORE	**.551**	.547	.338	**.521**	.281	.508
AFTER	.171	**.326**	.330	**.544**	.325	.504
SIMULTANEOUS	-	-	**.094**	-	.032	.080
INCLUDES	.245	**.428**	.140	**.363**	.065	.326
IS_INCLUDED	.478	**.503**	.144	**.385**	.145	.364
VAGUE	.445	**.463**	.664	**.693**	.676	.426
Overall	.449	**.476**	.529	**.605****	.516	.443

Table 2: Micro-averaged F1-scores per TLINK type with different feature vectors, evaluated on stratified 10-fold cross-validation over the training set. ** denotes $p < .01$.

TLINK type	**E-D**		**E-E**			
	$\vec{f}$	$\vec{w}$	$\vec{f}$	$(\vec{w}_1 \oplus \vec{w}_2)$	$(\vec{w}_1 + \vec{w}_2)$	$(\vec{w}_2 - \vec{w}_1)$
BEFORE	.587	**.627**	.388	**.443**	.264	.408
AFTER	.265	**.444**	.267	.412	.246	**.467**
SIMULTANEOUS	-	-	-	-	-	-
INCLUDES	.067	**.217**	.066	.068	-	**.156**
IS_INCLUDED	**.559**	.524	.111	**.435**	.154	.155
VAGUE	**.474**	.424	**.612**	.592	.611	.313
Overall	.476	**.479**	.493	**.496**	.444	.338

Table 3: F1-scores per TLINK type with different feature vectors, evaluated on the test set.

With the exception of SIMULTANEOUS and VAGUE, all of the other TLINK types are asymmetric, e.g. BEFORE/AFTER, INCLUDES/IS_INCLUDED.

Note that the classifier with subtracted word vectors $(\vec{w}_2 - \vec{w}_1)$ as features is able to capture event pairs labelled as SIMULTANEOUS, which are failed to be detected by the classifier with concatenated word embeddings. In some cases, the SIMULTANEOUS event pairs are co-referring events that have similar meanings, e.g. (*attack*, *strike*). By subtracting the word vectors of such event headwords, we could get a feature vector close to the origin $(0, 0, ..., 0)$, which can be used by the classifier to capture the SIMULTANEOUS relation. This kind of information is not available if we use concatenated word vectors as features. However, the traditional feature set containing Wordnet semantic similarity/relatedness information still yields a better performance than subtracted word embeddings.

For E-D pairs, using word vectors $\vec{w}$ as features (.476 F1-score) is better than using traditional features $\vec{f}$ (.449 F1-score), in particular for INCLUDES and AFTER labels. Given that in a news text, the document creation time is usually of TIME or DATE types, this means that word embeddings are able to predict when events last longer than a day, hence the appropriate label, i.e. INCLUDES, is chosen.

The same phenomena are also observed in the evaluation on test data, shown in Table 3: (i) concatenation $(\vec{w}_1 \oplus \vec{w}_2)$ is the best vector combination scheme for E-E edges and (ii) using word embeddings increases the performance on INCLUDES and AFTER labelling of E-D pairs. Pairs labelled as SIMULTANEOUS are highly under-represented in the dataset for all types of edges, composing only around 1.1% of the training data and 1.3% of the test data. This explains the failure of the classification models in capturing this particular relation in the test data.

Overall, using word vectors is better than using carefully crafted traditional features, particularly on the 10-fold cross-validation evaluation over the training data. However, on the evaluation over the test set, the improvements on the overall F1-scores are not significant ($p > .05$), even though we observe statistically significant improvements on specific TLINK types (e.g., INCLUDES for E-D pairs and IS_INCLUDED for E-E pairs).

	E-D			**E-E**		
TLINK type		**S1**	**S2**		**S1**	**S2**
	$\vec{w}$	$(\vec{w} \oplus \vec{f})$	$(\vec{p} \oplus \vec{f})$	$(\vec{w_1} \oplus \vec{w_2})$	$((\vec{w_1} \oplus \vec{w_2}) \oplus \vec{f})$	$(\vec{p} \oplus \vec{f})$
BEFORE	.627	.667	**.671**	.443	**.501**	.471
AFTER	.444	**.518**	.466	.412	**.435**	.430
SIMULTANEOUS	-	-	-	-	**.154**	-
INCLUDES	.217	**.333**	.250	.068	**.085**	.049
IS_INCLUDED	.524	.554	**.600**	**.435**	.288	.250
VAGUE	.424	.449	**.502**	.592	.596	**.613**
Overall	.479	.524*	**.534***	.496	.512*	**.519****

Table 4: F1-scores on both experimental settings S1 (concatenating word vectors and traditional features) and S2 (stacking), evaluated on the test set. * denotes significance at $p < .05$, ** denotes $p < .01$.

4.3 Experiment 2: Combining Traditional Features and Word Embeddings

To assess whether traditional features are still relevant in the presence of word vectors, two experimental settings are considered:

S1 We simply concatenate word vectors and traditional feature vectors together as the feature sets for the classifiers: $(\vec{w} \oplus \vec{f})$ is taken as the feature set for E-D and $((\vec{w_1} \oplus \vec{w_2}) \oplus \vec{f})$ for E-E pairs.

S2 We employ an ensemble learning technique, namely *stacking*, to combine both sets of features. First, the classifiers with word vectors as features, i.e., $\vec{w}$ for E-D and $(\vec{w_1} \oplus \vec{w_2})$ for E-E, are trained on the training data in a 10-fold cross-validation scheme, producing prediction vectors $\vec{p}$, i.e., one-hot representation of predicted labels, for the whole training data. Then, a combined classifier is trained to make a final prediction using $(\vec{p} \oplus \vec{f})$ as the feature set.

Again, LIBLINEAR (L2-regularized logistic regression) is used to build all the classifiers, which are then evaluated on the test data. Table 4 gives an overview of system performances on both settings, along with F1-scores when using the best feature vectors according to previous experiments, i.e., $\vec{w}$ for E-D and $(\vec{w_1} \oplus \vec{w_2})$ for E-E pairs.

For both E-D and E-E pairs, a super classifier trained using $\vec{p} \oplus \vec{f}$ as the feature vector performs better than a classifier trained with concatenated word vectors and traditional features. Furthermore, the super classifiers significantly outperform classifiers trained with the best single feature sets (word embeddings for E-D and E-E pairs), i.e., .534 vs. .479 F1-scores for E-D ($p < .05$) and .519 vs. .496 F1-scores for E-E pairs ($p < .01$).

5 Discussion

Our final system is composed of a rule set for T-T pairs, and three LIBLINEAR (L2-regularized logistic regression) classifiers for E-D, E-T and E-E pairs. Following the results from our experiments detailed in Section 4, we consider the best feature set for E-D and E-E pairs, i.e., the combination of word embeddings and traditional features via stacked learning $(\vec{p} \oplus \vec{f})$. Meanwhile, for E-T edges, we use the traditional feature vector $\vec{f}$ as the feature set.

Comparison of system performances We compare our system performance with two baseline systems. The first baseline labels all edges as VAGUE, which is the baseline system reported in Chambers et al. (2014). The second baseline system chooses the majority labels (non-VAGUE) for each type of edges, i.e., BEFORE for T-T, E-D and E-E, and AFTER for E-T pairs. As reported in Table 5, our system outperforms both baselines.

We also report in Table 5 our system performance in comparison with CAEVO (Chambers et al., 2014), the only existing temporal ordering system evaluated on the TimeBank-Dense corpus. Note that CAEVO is a hybrid system combining several rule-based and machine-learned classifiers in a sieve-based architecture, which includes transitive reasoning after each classifier labels the entity pairs. Our

System	T-T	E-D	E-T	E-E	Overall
Baseline: All VAGUE	.203	.277	.388	.447	.409
Baseline: Majority (non-VAGUE)	.508	.241	.305	.269	.278
Our system	**.780**	.534	.468	**.519**	**.518**
CAEVO	.712	**.553**	**.494**	.494	.507

Table 5: The comparison of system performances (F1-scores) for each edge type and the overall entity pairs.

	Our system			CAEVO		
Relation	**P**	**R**	**F1**	**P**	**R**	**F1**
BEFORE	.58	.46	**.51**	.52	.45	.49
AFTER	.59	.35	.44	.55	.38	**.45**
SIMULTANEOUS	.92	.28	**.43**	.71	.31	**.43**
INCLUDES	.15	.09	.11	.44	.21	**.28**
IS_INCLUDED	.51	.44	.47	.57	.43	**.49**
VAGUE	.49	.71	**.58**	.48	.66	.56

Table 6: The comparison of system performances on individual relation types.

system, composed of only one rule-set and three supervised-classifiers, is marginally better than CAEVO, particularly for T-T and E-E pairs, with the overall F1-scores of .518 vs. .507 (CAEVO).

CAEVO includes hand-crafted rules for E-D and E-T pairs, for instance rules to label edges between reporting events and DCTs (as IS_INCLUDED) and rules for edges between one verbal event and one timex based on temporal prepositions connecting the two (e.g. prepositions *for*, *at* and *throughout* signal a SIMULTANEOUS relation). It is very likely that our system based on general-purpose features cannot beat these very specific and carefully designed rules.

Finally, we present our system performance (in terms of precision, recall and F1-score) per-relation in Table 6, which is again compared to CAEVO. Overall, the two systems have comparable performances for all relation types, with the exception of the INCLUDES relation, in which CAEVO clearly outperforms our system.

Intra- vs. inter-sentential entity pairs Since we argue that embedding-based features may be particularly beneficial when no overt clues to express the temporal relation are present, as is often the case in inter-sentential relations, we compare the performance of the different feature vectors on inter- and intra-sentential relations in the test set. Results are reported in Table 7.

Feature vector	**E-D**		**E-E**	
	same	**diff**	**same**	**diff**
$\vec{f}$	-	.476	.466	.507
$\vec{w}$ or $(\vec{w_1} \oplus \vec{w_2})$	-	.479	.488	.501
S1: $(\vec{w} \oplus \vec{f})$ or $((\vec{w_1} \oplus \vec{w_1}) \oplus \vec{f})$	-	.524	**.516**	.509
S2: $(\vec{p} \oplus \vec{f})$	-	**.534**	.492	**.533**

Table 7: F1-scores for different feature vectors, evaluated on pairs in the test set belonging to the same sentence (same) and different sentences (diff).

Combining word embeddings and traditional features in *stacking* setting (S2) is shown to be beneficial for entity pairs occurring in different sentences. Interestingly, the combination of word embeddings and traditional features in the concatenated setting (S1) is quite beneficial to the classification of E-E pairs in the same sentence.

6 Conclusions

We have analysed the contribution of word embeddings to temporal relation type classification, specifically for E-D and E-E edges. The evaluation results shed some light on how word embeddings can potentially improve a classifier performance for this particular task, i.e., in combination with traditional features in the stacked learning scheme. These results confirm that word embeddings can become effective features when there are no overt markers of temporal relations.

Compared with the state-of-the-art system evaluated on the same corpus, CAEVO, our system achieves quite similar performances, even though it is based on a much simpler architecture. We believe that, integrating our rule set (for T-T pairs) and classifiers (for E-D, E-T and E-E pairs) in CAEVO's sieve-based architecture completed with transitive reasoning, may result in an improvement of the state-of-the-art on the task, which we plan to evaluate soon.

Several works have recently presented methods for building task-specific word embeddings (Hashimoto et al., 2015; Boros et al., 2014; Nguyen and Grishman, 2014; Tang et al., 2014). We believe that this may be beneficial also for temporal ordering, and we plan to build this kind of embeddings in the future, instead of using general-purpose vectors.

Acknowledgments

The research leading to this paper was partially supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404) and the National University of Singapore. We thank Ilija Ilievski, Min-Yen Kan and Hwee Tou Ng, who provided insight and expertise that greatly assisted the research.

References

James F. Allen. 1984. Towards a general theory of action and time. *Artif. Intell.*, 23(2):123–154, July.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Anders Bjorkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Coling 2010: Demonstrations*, pages 33–36, Beijing, China, August. Coling 2010 Organizing Committee.

Emanuela Boros, Romaric Besançon, Olivier Ferret, and Brigitte Grau. 2014. Event role extraction using domain-relevant word representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1852–1857, Doha, Qatar, October. Association for Computational Linguistics.

Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2201–2211, Lisbon, Portugal, September. Association for Computational Linguistics.

Peter F. Brown, Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic, June. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.

Leon Derczynski and Robert Gaizauskas. 2013. Temporal signals help label temporal relations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 645–650, Sofia, Bulgaria, August. Association for Computational Linguistics.

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, Atlanta, Georgia, June. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

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

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT 2013*, pages 758–764, Atlanta, Georgia, June. ACL.

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.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2015. Task-oriented learning of word embeddings for semantic relation classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 268–278, Beijing, China, July. Association for Computational Linguistics.

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

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291, Uppsala, Sweden, July. 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 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760, Sydney, Australia, July. Association for Computational Linguistics.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Paramita Mirza and Sara Tonelli. 2014. Classifying temporal relations with simple features. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 308–317, Gothenburg, Sweden, April. Association for Computational Linguistics.

Paramita Mirza and Sara Tonelli. 2016. CATENA: Causal and Temporal relation Extraction from Natural language texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, Osaka, Japan, December. Association for Computational Linguistics.

Marc Moens and Mark Steedman. 1987. Temporal ontology in natural language. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 1–7, Stanford, California, USA, July. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 68–74, Baltimore, Maryland, June. Association for Computational Linguistics.

Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The TextPro Tool Suite. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*.

Andrea Setzer. 2001. *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study*. University of Sheffield.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014. Coooolll: A Deep Learning System for Twitter Sentiment Classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

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 *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 Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic, June. Association for Computational Linguistics.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden, July. Association for Computational Linguistics.