A Sequential Model for Classifying Temporal Relations between Intra-Sentence Events

Prafulla Kumar Choubey and **Ruihong Huang**
Department of Computer Science and Engineering
Texas A&M University
(prafulla.choubey, huangrh)@tamu.edu

Abstract

We present a sequential model for temporal relation classification between intra-sentence events. The key observation is that the overall syntactic structure and compositional meanings of the multi-word context between events are important for distinguishing among fine-grained temporal relations. Specifically, our approach first extracts a sequence of context words that indicates the temporal relation between two events, which well align with the dependency path between two event mentions. The context word sequence, together with a parts-of-speech tag sequence and a dependency relation sequence that are generated corresponding to the word sequence, are then provided as input to bidirectional recurrent neural network (LSTM) models. The neural nets learn compositional syntactic and semantic representations of contexts surrounding the two events and predict the temporal relation between them. Evaluation of the proposed approach on TimeBank corpus shows that sequential modeling is capable of accurately recognizing temporal relations between events, which outperforms a neural net model using various discrete features as input that imitates previous feature based models.

1 Introduction

Identifying temporal relations between events is crucial to constructing events timeline. It has direct application in tasks such as question answering, event timeline generation and document summarization.

Bush said he saw little reason to be optimistic about a settlement of the **dispute**, which stems from Iraq's **invasion** of oil-wealthy Kuwait and its subsequent military **buildup** on the border of Saudi Arabia.
Relations: (dispute $after$ ^{rel_1} invasion, invasion $ibefore$ ^{rel_2} buildup, dispute $after$ ^{rel_3} buildup)

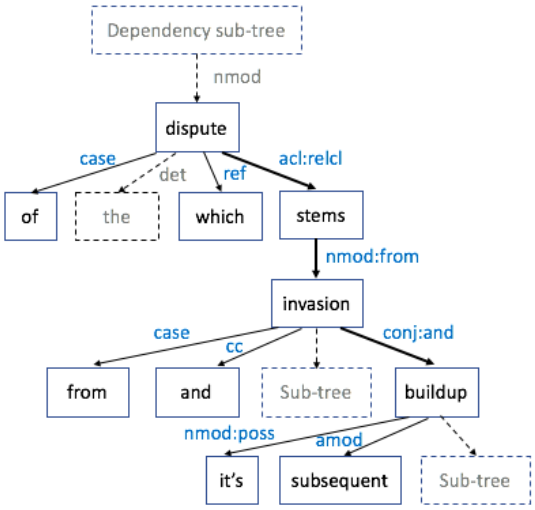

Figure 1: Example sentence to illustrate the temporal context for event pairs.

Previous works studied this task as the classification problem based on discrete features defined over lexico-syntactic, semantic and discourse features. However, these features are often derived from local contexts of two events and are only capable of capturing direct evidences indicating the temporal relation. Specifically, when two events are distantly located or are separated by other events in between, feature based approaches often fail to utilize compositional evidences, which are hard to encode using discrete features.

Consider the example sentence in Figure 1. Here, the first two temporal re-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1796–1802
Copenhagen, Denmark, September 7–11, 2017.

lations, *dispute* **after** $^{\mathbf{rel_1}}$ *invasion* and *invation* **ibefore** $^{\mathbf{rel_2}}$ *buildup*, involve events that are close by and discrete features, such as dependency relations and bag-of-words extracted from local contexts of two events, might be sufficient to correctly detect their relations. However, for the temporal relation *dispute* **after** $^{\mathbf{rel_3}}$ *buildup*, the context between the two events is long, complex and involves another event (*invasion*) as well, which makes it challenging for any individual feature or feature combinations to capture the temporal relation.

We propose that the overall syntactic structure of in-between contexts including the linear order of words as well as the compositional semantics of multi-word contexts are critical for predicting the temporal relation between two events. Furthermore, the most important syntactic and semantic structures are derived along dependency paths between two event mentions[1]. This aligns well with the observation that semantic composition relates to grammatical dependency relations (Monroe and Wang, 2014; Reddy et al., 2016).

Our approach defines rules on dependency parse trees to extract temporal relation indicating contexts. First, we extract the dependency path between two event mentions. Then we apply two heuristic rules to enrich extracted dependency paths and deal with complex syntactic structures such as punctuations. Empirically, we found that parts-of-speech tags (POS) and dependency sequences generated following the dependency path provide evidences to predict the temporal relation as well.

We use neural net sequence models to capture structural and semantic compositionality in describing temporal relations between events. Specifically, we generate three sequences for each dependency path, the word sequence, the POS tag sequence and the dependency relation sequence. Using the three types of sequences as input, we train bi-directional LSTM models that consume each of the three sequences and model compositional structural information, both syntactically and semantically.

The evaluation shows that each type of sequences is useful to temporal relation classification between events. Our complete neural net model taking all the three types of sequences performs the best, which clearly outperforms feature based models.

[1]In this paper, we restrict ourselves to study temporal relation classification between event mentions that are within one sentence.

2 Related Works

Most of the previous works on temporal relation classification are based on feature-based classifiers. Mani et al. (2006) built MaxEnt classifier on hand-tagged features in the corpus, including tense, aspect, modality, polarity and event class for classifying temporal relations. Later Chambers et al. (2007) used a two-stage classifier which first learned imperfect event attributes and then combined them with other linguistic features in the second stage to perform the classification.

The following works mostly expanded the feature sets (Cheng et al., 2007; Bethard and Martin, 2007; UzZaman et al., 2012; Bethard, 2013; Kolomiyets et al., 2012; Chambers, 2013; Laokulrat et al., 2013). Specifically, Chambers (2013) used direct dependency path between event pairs to capture syntactic context. Laokulrat et al. (2013) used 3-grams of paths between two event mentions in a dependency tree as features instead of full paths as those are too sparse. We found that modeling the entire path as one sequence provides greater compositional evidence on the temporal relation. In addition, modifiers attached to the words in a path with specific dependency relations like *nmod:tmod* are also informative.

Ng (2013) proposed a hybrid system for temporal relation classification that combines the learned classifier with 437 hand-coded rules. Their system first applied high-accuracy rules and then used the learned classifier, trained on rich features including those high-accuracy rules as features, to classify the cases that were not handled by the rules. Ng et al. (2013) also showed the effectiveness of different discourse analysis frameworks for this task. Later Mirza and Tonelli (2014) showed that a simpler approach based on lexico-syntactic features achieved results comparable to Ng (2013). They also reported that dependency order between events, either governor-dependent or dependent-governor, was not useful in their experiments. However, we show that dependency relations, when modeled as a sequence, contribute significantly to this task.

3 Temporal Link Labeling

In this section, we describe the task of temporal relation classification, dataset, context words se-

quence extraction model and the used recurrent neural net based classifier.

3.1 Task description

Early works on temporal relation classification Mani et al. (2006); Chambers et al. (2007) and the first two versions of TempEval (Verhagen et al., 2007, 2010) simplified the task by considering only six relation types. They combined the pair of relation types that are the inverse of each other and ignored the relations *during* and *during_inv*. Then TempEval-3 (Uzzaman et al., 2013) extended the task to complete 14 class classification problem and all later works have considered all 14 relations. Our model performs 14-class classification following the recent works, as this is arguably more challenging (Ng, 2013). Also, we consider gold annotated event pairs, mainly because the corpus is small and distribution of relations is very skewed. All previous works focusing on the problem of classifying temporal relation types assumed gold annotation.

3.2 Dataset

Relations	Train	Validate	Test
After	419	60	120
Before	337	48	97
Simultaneous	288	41	83
Identity	147	21	43
Includes	141	20	41
IS_included	93	13	27
Ended_by	66	9	19
During_inv	26	4	8
Begun_by	25	3	7
Begins	22	3	7
IBefore	16	2	5
IAfter	12	2	4
During	11	1	3
Ends	9	2	3
Total	1612	229	467

Table 1: Distribution of temporal relations in TimeBank v1.2.

We have used TimeBank corpus v1.2 for training and evaluating our model. The corpus consists of 14 temporal relations between 2308 event pairs, which are within the same sentence. These relations (Saurı et al., 2006) are *simultaneous, before, after, ibefore, iafter, begins, begun_by, ends, ended_by, includes, is_included, during, during_inv, identity*. Six pairs among them are inverse of each other and other two types are commutative ($e_1 \mathcal{R} e_2 \equiv e_2 \mathcal{R} e_1, \mathcal{R} \in \{$*identical, simultaneous*$\}$). Our sequential model requires that relation should always be between e_1 and e_2, where e_1 occurs before e_2 in the sentence. Therefore, before extracting the sequence, we inverted the relation types in cases where relation type was annotated in opposite order. Final distribution of dataset is given in Table 1.

3.3 Extracting Context Word Sequence

First, we extract words that are in the dependency path between two event mentions. However, event pairs can be very far in a sentence and are involved in complex syntactic structures. Therefore, we also apply two heuristic rules to deal with complex syntactic structures, e.g., two event mentions are in separate clauses and have a punctuation sign in their context. We describe our specific rules below. We used the Stanford parser (Chen and Manning, 2014) for generating dependency relations and parts-of-speech tags and all notations follow enhanced universal dependencies (De Marneffe and Manning, 2008).

Rule 1 (punctuation): Comma directly influences the meaning in text and omitting it may alter the meaning of phrase. Therefore, include comma if it precedes or follows e_1, e_2 or their modifiers.

Rule 2 (children): Modifiers like *now, then, will, yesterday, subsequent, when, was, etc.* contains information on the temporal order of events and help in grounding events to the timeline. These modifiers are often related to event mentions with a specific class of dependency relations. Include all such children of e_1, e_2 and other words in the path between them, which are connected with dependency relations nmod:tmod, mark, case, aux, conj, expl, cc, cop, amod, advmod, punct, ref.

3.4 Sequences and Classifier

We form three sequences on the extracted context words (with *t* words), which are based on (i) parts-of-speech tags: $\mathcal{P}_\mathcal{T} = p_1, p_2, ..., p_t$ (ii) dependency relations: $\mathcal{D}_\mathcal{T} = d_1, d_2, ..., d_n$[2] and (iii) word forms: $\mathcal{W}_\mathcal{T} = w_1, w_2, ..., w_t$.

We transform each p_i and d_i to a one-hot vector and each w_i to a pre-trained embedding vector (Pennington et al., 2014). Then each sequence of vectors are encoded using their corresponding forward ($LSTM_f$) and backward ($LSTM_b$) LSTM layers.

Classifier: Figure 2 shows an overview of our model. It consists of six LSTM (Hochreiter and

[2]we only consider dependency relations for words in path connecting e_1 and e_2.

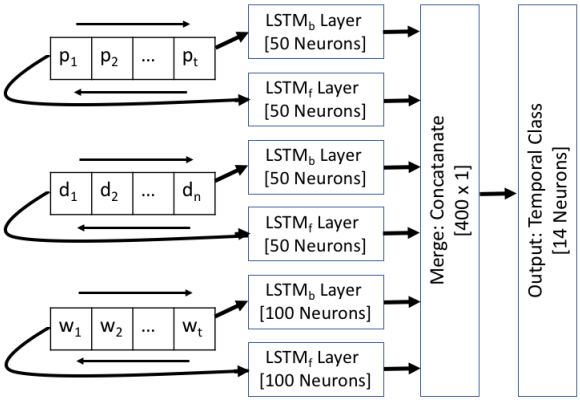

Figure 2: Bi-directional LSTM based classifier used for temporal relations classification.

Schmidhuber, 1997) layers, three of them encode feature sequences in forward order and remaining in reverse order. LSTM layers for POS tag and dependency relation have 50 neurons and have dropouts of 0.20. LSTM layers for word form have 100 neurons and have dropout of 0.25. All LSTM layers use 'tanh' activation function. Forward and backward embeddings of all sequences are concatenated and fed into another neural layer with 14 neurons corresponding to 14 fine-grained temporal relations. This neural layer uses softmax activation function. We train model for 100 iterations using rmsprop optimizer on batch size of 100 and error defined by categorical cross-entropy (Chollet, 2015) .

4 Evaluation

We evaluate our model using accuracy which has been used in previous research works for temporal relation classification. We also compare model performance using per-class F-score and macro F-score. We briefly describe all the systems we have used for evaluation.

Majority Class: assigns "after" relation to all event pairs.

Unidirectional LSTMs: use single LSTM layer to encode each sequence (POS tags, dependency relation and word forms) individually for extracted phrase in forward order.

Bidirectional LSTMs: use two LSTM layers to encode each sequence individually, taken from POS tags, dependency and word forms sequences. The first layer encodes sequence in forward and second in reverse order.

2 Sequences: bi-directional LSTM based models considering all combinations of two sequences taken from POS tags, dependency and word forms sequences.

Full model: our complete sequential model considering POS, dependency and word forms sequences.

Direct dependency path: the same as **Full model** except that the two heuristic rules were not applied in extracting sequences.

Baseline I: a neural network classifier using discrete features described in Mirza and Tonelli (2014); Ng (2013). The features used are: POS tag, dependency relation, token and lemma of $e_1(e_2)$; dependency relations between $e_1(e_2)$ and their children; binary features indicating if e_1 and e_2 are related with the 'happens-before' or the 'similar' relation according to VerbOcean (Chklovski and Pantel, 2004), if e_1 and e_2 have the same POS tag, or if $e_1(e_2)$ is the root and e_1 modifies (or governs) e_2; the dependency relation between e_1 and e_2 if they are directly connected in the dependency parse tree; prepositions that modify (or govern) $e_1(e_2)$; signal words (Derczynski and Gaizauskas, 2012) and entity distance between e_1 and e_2. These features are concatenated and fed into an output neural layer with 14 neurons.

Baseline II: a neural network classifier using POS tags and word forms of words in the *surface path* as input. The *surface path* consists of words that lie in between two event mentions based on the original sentence. The classifier uses four LSTM layers to encode both POS tag and word sequences in forward and backward order. The output neural layer and parameters for all LSTM layers are kept the same as the *Full model*.

Baseline III: a neural network classifier based on event embeddings for both event mentions that were learned using bidirectional LSTMs (Kiperwasser and Goldberg, 2016). The learning uses two LSTM layers, each with 150 neurons and dropout of 0.2, to embed the forward and backward representations for each event mention. The input to LSTM layers are sequences of concatenated word embeddings and POS tags; each sequence corresponding to 19 context words to the left or to the right side of an event mention for the forward or the backward LSTM layer respectively. Event embeddings are then concatenated and fed into an output neural layer with 14 neurons.

All baselines are trained using rmsprop optimizer on an objective function defined by categorical cross entropy and their output layer uses softmax activation function.

4.1 Results and Discussion

Models	*Accuracy*
Majority Class	25.69
Baseline I	41.97
Unidirectional LSTM: only POS	34.90
only Word	35.12
only Dependency	34.48
Bidirectional LSTMs: only POS	39.19
only Word	37.69
only Dependency	40.04
2 Sequences: POS + Word	44.54
Dependency + Word	45.18
Dependency + POS	47.75
Full Model	**53.32**
Direct dependency path	49.25
Baseline II	43.90
Baseline III	44.75

Table 2: Temporal relation classification result on TimeBank corpus.

Relations	*OurSystem*			*BaselineI*		
	P	R	F	P	R	F
After	**0.62**	**0.68**	**0.65**	0.56	0.48	0.45
Before	**0.56**	**0.52**	**0.53**	0.37	0.45	0.41
Simultan.	**0.44**	**0.51**	**0.47**	0.32	0.43	0.37
Identity	**0.47**	**0.56**	**0.51**	0.45	0.53	0.49
Includes	**0.59**	**0.39**	**0.47**	0.43	0.30	0.35
IS_includ.	0.5	**0.56**	0.53	**0.61**	0.51	**0.56**
Ended_by	**0.48**	**0.63**	**0.55**	0.41	0.47	0.44
During_in.	0	0	0	0	0	0
Begun_by	**0.75**	**0.43**	**0.55**	0	0	0
Begins	**1.0**	**0.29**	**0.44**	0	0	0
IBefore	**0.4**	**0.4**	**0.4**	0	0	0
IAfter	**0.33**	**0.25**	**0.29**	0	0	0
During	0	0	0	0	0	0
Ends	0	0	0	0	0	0
Macro Av.	**0.44**	**0.37**	**0.40**	0.23	0.22	0.22

Table 3: Per-class results of our best system and the baseline I.

Table 2 reports accuracy scores for all the systems. We see that simple sequential models outperform the strong feature based system, *Baseline I*, which used various discrete features. Note that dependency relation and POS tag sequences alone achieve reasonably high accuracies. This implies that an important aspect of temporal relation is contained in the syntactic context of event mentions. Moreover, Mirza and Tonelli (2014) observed that discrete features based on dependency parse tree did not contribute to improving their classifier's accuracy. On the contrary, using the sequence of dependency relations yields a high accuracy in our setting which signifies the advantages of using sequential representations for this task. Our *Full Model* achieves a performance gain of 11.35% over *Baseline I*.

We developed two more baselines (*Baseline II and III*) that do not require syntactic information as well as the *Direct dependency path* model that used no rules. The *Full Model* outperformed them by 9.42%, 8.57% and 4.07% respectively. This affirms that the most useful syntactic and semantic structures are derived along dependency paths and additional context words, including prepositions, signal words and punctuations that are indirectly attached to event words, entail evidence on temporal relations as well.

Table 3 compares precision, recall and F_1 scores of our *Full Model* with *Baseline I*. Our model performs reasonably well compared to the baseline system for most of the classes. In addition, it is able to identify relations present in small proportion like *begun_by, ibefore, iafter etc.*, which the baseline system couldn't identify. A similar observation was also reported by Mirza and Tonelli (2014) that relation types *begins, ibefore, ends* and *during* are difficult to identify using feature based systems, which often generate false positives for *before* and *after* relations.

5 Conclusion and Future work

In this paper, we have focused on modeling syntactic structural information and compositional semantics of contexts in predicting temporal relations between events in the same sentence. Our approach extracts lexical and syntactic sequences from contexts between two events and feed them to recurrent neural nets. The evaluation shows that our sequential models are promising in distinguishing among fine-grained temporal relations.

In the future, we will extend our sequential models to predict temporal relations for event pairs spanning across multiple sentences, for instance by incorporating discourse relations between sentences in a sequence.

Acknowledgments

We want to thank our anonymous reviewers for their insightful review comments and suggestions that helped making evaluations more extensive.

References

Steven Bethard. 2013. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 10–14.

Steven Bethard and James H Martin. 2007. Cu-tmp: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 129–132. Association for Computational Linguistics.

Nathanael Chambers. 2013. Navytime: Event and time ordering from raw text. Technical report, DTIC Document.

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*, pages 173–176. Association for Computational Linguistics.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. 2007. Naist. japan: Temporal relation identification using dependency parsed tree. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 245–248. Association for Computational Linguistics.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *EMNLP*, volume 4, pages 33–40.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

Leon Derczynski and Robert Gaizauskas. 2012. Using signals to improve automatic classification of temporal relations. *arXiv preprint arXiv:1203.5055*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *arXiv preprint arXiv:1603.04351*.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 88–97. Association for Computational Linguistics.

Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Uttime: Temporal relation classification using deep syntactic features.

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*, pages 753–760. Association for Computational Linguistics.

Paramita Mirza and Sara Tonelli. 2014. Classifying temporal relations with simple features. In *EACL*, volume 14, pages 308–317.

Will Monroe and Yushi Wang. 2014. Dependency parsing features for semantic parsing.

Jun-Ping Ng, Min-Yen Kan, Ziheng Lin, Vanessa Wei Feng, Bin Chen, Jian Su, and Chew Lim Tan. 2013. Exploiting discourse analysis for article-wide temporal classification. In *EMNLP*, pages 12–23.

Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Roser Saurı, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines version 1.2. 1.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333*.

Naushad Uzzaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations.

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*, pages 75–80. 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. Association for Computational Linguistics.