# Acquire common sense knowledge between events via weakly supervised approach

**Wenlin Yao**
Department of Computer Science and Engineering
Texas A&M University
`wenlinyao@tamu.edu`

## 1 Introduction

Capabilities to recognize temporal relations between two events can benefit many Natural Language Processing applications, including event timeline generation, script knowledge extraction, text summarization and event prediction.

This is a challenging task because temporal relations can be described in dramatically different contexts depending on domains and pairs of events, signifying different semantic meanings. In order to capture various contexts, large amounts of labeled data are needed to train a high-coverage temporal relation classifier. However, almost all existing datasets that contain event-event temporal relation annotations are limited in size and domains, such as Automatic Context Extraction (ACE) (Strassel et al., 2008) and TimeBank (Pustejovsky et al., 2003), which generally contain several hundred documents.

We observed that event pairs presenting regularities tend to show the same temporal relation despite of various contexts they may occur in. For instance, *arrest* events tend to happen after *attack* events, and the following sentential contexts all indicate the same temporal relation:

*Under pressure following suicide attacks, police arrested scores of activists on Monday.*

*Two men were arrested on suspicion of carrying out the Mumbai attacks.*

*Carlos was arrested in Sudan in August in connection with two bomb attacks in France in 1982.*

Leveraging this key observation, we propose a bootstrapping approach that focuses on recognizing *after* or *before* temporal relations and substantially reduces the reliance on human annotated data. We start by identifying regular event pairs that have occurred enough times with an explicit temporal pattern, i.e., EV_A *after (before)* EV_B. We then populate these seed event pairs in a large unlabeled corpus to quickly collect hundreds of thousands of sentences that contain a regular event pair, which are then used as training instances to obtain an initial contextual temporal relation classifier. Next, the classifier is applied back to the text corpus and label new sentential contexts that indicate a specific *after* or *before* temporal relation between events. Then new regular event pairs can be identified, which are event pairs that have a majority of their sentences labeled as describing a particular temporal relation. The newly identified regular event pairs will be used to augment seed event pairs and identify more temporal relation sentential contexts in the unlabeled corpus. The bootstrapping learning process iterates.

## 2 Related Work

Most of existing temporal relation classifiers were learned in a supervised manner and depend on human annotated data. In the TempEval campaigns (Verhagen et al., 2007, 2010; UzZaman et al., 2013), various classification models and linguistic features (Bethard, 2013; Chambers et al., 2014; Llorens et al., 2010; D'Souza and Ng, 2013; Mirza and Tonelli, 2014) have been applied to identify temporal relations between two events. For example, a recent study by (D'Souza and Ng, 2013) applied sophisticated linguistic, semantic and discourse features to classify temporal relations between events. They also included 437 hand-coded rules in building a hybrid classification model. CAEVO, a CAscading EVent Ordering architecture by Chambers et al. (2014), applied a sieve-based architecture for event temporal ordering. CAEVO is essentially a hybrid model as well. While the first few sieves are rule based and deterministic, the latter ones are machine learned using human annotated data.

# 3 Event Representations

## 3.1 Verb Event Phrases

To ensure a good coverage of regular event pairs, we consider all verbs[1] as event words except reporting verbs[2]. The thematic patient of a verb refers to the object being acted upon and is essentially part of an event, therefore, we first include the patient of a verb in forming an event phrase. We use Stanford dependency relations (Manning et al., 2014) to identify the direct object of an active verb or the subject of a passive verb. The agent is also useful to specify a event especially for a intransitive verb event, which does not have a patient.

## 3.2 Noun Event Phrases

We include a prepositional object of a noun event in forming an noun event phrase. We first consider an object headed by the preposition *of*, then an object headed by the preposition *by*, lastly an object headed by any other preposition.

## 3.3 Generalizing Event Arguments Using Named Entity Types

Including arguments into event representations generates specific event phrases though. In order to obtain generalized event phrase forms, we replace specific name arguments with their named entity types (Manning et al., 2014). We also consider replacing pronouns with their types, but concerned with poor quality of full coreference resolution, we only replace personal pronouns with their type PERSON. We observed that this strategy greatly improves generality of event phrases and facilitates the bootstrapping learning process. In section 5.1.2, we compare bootstrapping learning performance using generalized event representations v.s. using non-generalized event representations.

## 3.4 Regular Event Pair Candidates

Considering that it is not feasible to test all possible pairs of events in Gigaword and often two events that co-occur in a sentence have no temporal relation. In order to narrow down the search space, we identify candidate event pairs which are likely to have temporal relations.

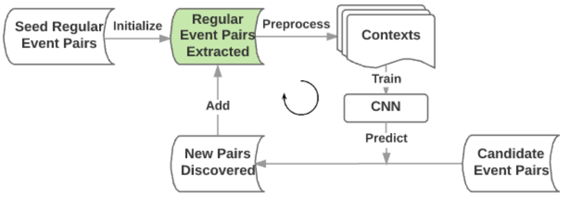

Figure 1: Overview of the Bootstrapping System

Two strategies are used to identify candidate event pairs. First, by intuition, if two event phrases co-occur (within a sentence) many times, the likelihood of the two events being related and having a temporal relation should be higher compared to event phrases that rarely co-occur. Therefore, we select event phrase pairs that co-occur within a sentence for more than 100 times as candidate event pairs. Second, we use two specific temporal relation patterns, EV_A *after (before)* EV_B, that explicitly indicate two events are in a after (before) relation. We extract an event pair as a candidate regular pair if it occurs three or more times with one of the patterns in the text corpus. The assumption is that if a pair of events shows a particular temporal relation regularly, it is likely to be seen in the above textual patterns as well. Specifically, we extract the governor and dependent word of the dependency relation *prep_after (prep_before)* in the annotated English Gigaword (Napoles et al., 2012) and check whether each word is an event[3]. If yes, we form an event phrase for each event and obtain an event pair. In addition, we expect regular event pairs to occur mostly in a single temporal order, either *before* or *after*, and discard event pairs that have showed mixed temporal orders. Specifically, a regular event pair is required to occur in a particular temporal relation more than 90% of times.

Overall by applying the two strategies, we obtained a candidate event pair pool that consists of 40,278 event pairs.

# 4 Bootstrapping Regular Event Pairs

Figure 1 illustrates how the bootstrapping system works. We first populate seed regular event pairs in the text corpus and identify sentences that contain a regular event pair as training instances. We train a contextual temporal relation classifier, us-

---

[1] We used POS tags to detect verb events.

[2] Reporting verbs, such as "said", "told" and "added", are commonly seen in news articles. We determined that most of event pairs containing a reporting verb are not very interesting and informative and we therefore discarded these event pairs.

[3] Note we consider any verb and a noun that is in our noun event list as an event.

ing Convolutional Neural Nets (CNNs), to identify specific contexts describing a temporal *after (before)* relation. We then apply the classifier to the corpus to identify new sentences that describe a particular temporal relation, from which new regular event pairs can be extracted. Note that the classifier is only applied to sentences that contain a candidate regular event pair. The bootstrapping process repeats until the number of newly identified regular event pairs is less than 100.

While we used the whole Gigaword (Napoles et al., 2012) to identify regular event pairs, we only use the New York Times section of Gigaword for bootstrapping learning.

### 4.1 Regular Event Pair Seeds

In order to ensure high quality of seed pairs, we only consider event pairs that have occurred in explicit temporal relation patterns, EV_A *after (before)* EV_B, as seed event pairs. Furthermore, we require each seed regular event pair to have occurred in a temporal relation pattern for at least ten times. Specifically, we identified 2,110 seed regular event pairs using the Gigaword corpus[4].

### 4.2 Contextual Temporal Relation Classification

We used a Convolutional Neural Net (CNN) as our classifier, inspired by recent successes of CNN models in various NLP tasks and applications, such as sentiment analysis (Kalchbrenner et al., 2014; Kim, 2014), sequence labeling (Collobert et al., 2011) and semantic parsing (Yih et al., 2014). As shown in figure 2, our CNN architecture is a slight variation of the previous models as described in (Kim, 2014; Collobert et al., 2011). It has one convolutional layer with 100 hidden nodes, one pooling layer and one fully connected softmax layer.

The input are word embeddings of an array of sentential context words. A convolution filter is applied to a sliding window of every $h$ words to provide input for each hidden node. We use Rectified Linear Unit (ReLU) as the non-linear activation function. We next apply a max-pooling operation to take the maximum value over a feature map. The final softmax layer output probability distributions over three classes (AFTER, BEFORE and OTHER) indicating the temporal relation between a pair of events in a sentence. Specifically, the temporal relations are defined with respect to the textual order the two events are presented in a sentence. If the first event is temporally BEFORE the second event as described in a sentence, this instance will be labeled as BEFORE. Otherwise if the first event is temporally AFTER the second event as described in a sentence, the instance will be labeled as AFTER. The class OTHER is to capture all the rest contexts that may describe a temporal relation other than *after (before)* or do not describe a temporal relation.

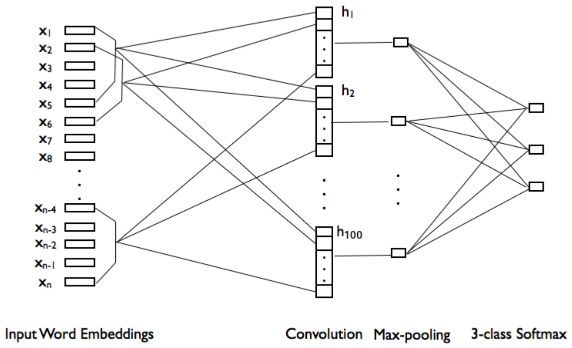

Figure 2: CNN Model Architecture

In our experiments, we use pre-trained 300-dimention word2vec word embeddings (Mikolov et al., 2013) that are trained on 100 billion words of Google News and we use a filter window size of 5. In training, we used stochastic gradient descent with Adadelta update rule (Zeiler, 2012) and mini-batch size of 100, in addition, we applied dropout (Hinton et al., 2012) with rate $p = 0.5$ to avoid overfitting of the CNN model. We also randomly selected 10% of the training data as the validation set and chose the classifier with the highest validation performance within the first 10 epochs.

#### 4.2.1 Sentential Contexts: Local Windows v.s. Dependency Paths

We explore two types of contexts, local windows v.s. dependency paths, in order to identify contexts that effectively describe temporal relations between two events.

First, the local window based context for an event pair includes five words before the first event, five words after the second event and all the words between the two events. Note that two event phrases can be arbitrarily far from each other and long contexts are extremely challenging for a classifier to capture. In our experiments, we only

---

[4] By populating seed regular event pairs in the New York Times section of the Gigaword corpus, we extracted 7191 sentences and 11339 sentences that contain an event pair in a "before" and "after" temporal relation respectively.

consider sentences where two event mentions are at most 10 words away.

Second, we observed that not every word between two events is useful to predict their temporal relation. In order to concentrate on relevant context words, we further construct dependency path[5] based context representation. Specifically, considering a dependency tree as an undirected graph, we use breadth-first-search to extract a sequence of words connecting the first event word to the second event word. In addition, to capture important information in certain syntactic structures such as conjunctions, we extract children nodes for each word in the path. Finally, we sort extracted words according to their textual order in the original sentence and the sorted sequence of words is provided as an input to the CNN classifier.

#### 4.2.2 Negative Training Instances

Reasonably, most sentences in a corpus do not contain an event pair that is in a temporal "before/after" relation. Therefore, we use negative instances that are 10 times of the total number of positive training instances (i.e., sentences that contain an event pair in a *after (before)* relation).

### 4.3 New Regular Event Pair Selection Criteria

Recall that regular event pairs are event pairs that tend to show a particular temporal relation despite of their various contexts. Therefore, we identify a candidate event pair as a new regular event pair if majority of its sentential contexts, specifically 60% of contexts, were consistently labeled as a particular temporal relation (*after* or *before*) by the CNN classifier. In addition, we require that at least 15 instances of a regular event pair have been labeled as the majority temporal relation. In order to control semantic drift (McIntosh and Curran, 2009) in bootstrapping learning, we increase the threshold by 5 after each iteration.

Furthermore, in order to filter out ambiguous event pairs that can be in either *before* or *after* temporal order depending on concrete contexts, we require the absolute difference between number of instances labeled as AFTER and labeled as BEFORE to be greater than a ratio of the total number of instances, specifically, we set the ratio to be 40%.

## 5 Evaluation

Our bootstrapping system learned regular event pairs as well as a contextual temporal relation classifier. We evaluate each of the two learning outcomes separately.

### 5.1 Regular Event Pair Acquisition

#### 5.1.1 System Variations

We compare three variations of our system:

*Basic System*: in the basic system, we did not apply event argument generalization as described in section 3.3. In addition, we use local window based sentential contexts as input for the classifier.

+ *Arg Generalization*: on top of the basic system, we apply event argument generalization.

+ *Dependency Path Contexts (Full System)*: in the full system, we apply event argument generalization and use dependency path based sentential contexts as input for the classifier.

Table 1 shows the number of regular new pairs that were generated after each bootstrapping iteration by each of the three systems. First, we can see that event argument generalization is useful in obtaining roughly two times of seed regular event pairs. Second, event argument generalization is useful in recognizing additional regular event pairs in bootstrapping learning as well. Third, dependency path based sentential contexts are effective in capturing relevant sentential contexts for temporal relation classification, which enables the bootstrapping system to maintain a learning momentum and learn more regular event pairs.

#### 5.1.2 Accuracy of Regular Event Pairs

For each of the three system variations, we randomly selected 50 pairs from seed regular event pairs and 50 from bootstrapped event pairs[6] and asked two human annotators to judge the correctness of these acquired regular event pairs.

Specifically, for each selected event pair, we ask two annotators to label whether a temporal AFTER or BEFORE relation exists between the two events.

Table 2 shows the accuracy of regular event pairs learned by each system variation. We determine that an event pair is correctly predicted by a system if the system predicted temporal relation is the same as the label that has been assigned by both of the two annotators. The over-

---

[5] Stanford CoreNLP (Manning et al., 2014) were used to generate dependency trees.

[6] The seed pairs for the second and the third system are the same, so we evaluate the same 50 randomly selected seed pairs for the two systems.

| Systems | 0 (Seeds) | 1 | 2 | 3 | 4 | 5 | Total |
|---|---|---|---|---|---|---|---|
| Basic System | 1057 | 213 | 102 | 48 | – | – | 1420 |
| + Arg Generalization | 2110 | 638 | 323 | 81 | – | – | 3152 |
| + Dependency Path Contexts (Full System) | 2110 | 1230 | 555 | 288 | 156 | 62 | 4401 |

Table 1: Number of New Regular Event Pairs Generated after Each Bootstrapping Iteration

| Systems | Seed Pairs | New Pairs |
|---|---|---|
| Basic System | 0.73 | 0.55 |
| + Arg Generalization | 0.71 | 0.63 |
| + Dependency Path Contexts | | 0.67 |

Table 2: Accuracy of 100 Randomly Selected Event Pairs

| | |
|---|---|
| Common Sense | PERSON **worked** ← **graduation**<br>**career** → **announced** retirement<br>**wash** hands → **eating**<br>PERSON **returned** ← **visit** |
| Politics | government be **formed** ← **elections**<br>**fled** mainland ← **losing** war<br>**imposed** sanctions ← **invasion** of LOCATION<br>LOCATION **split** ← **war** |
| Business | **reached** agreement ← **negotiations**<br>**hosted** banquet ← **meeting**<br>**trading** → stock **closed** |
| Health | **cause** of death ← **cancer**<br>PERSON be **hospitalized** ← **suffering** stroke<br>PERSON **died** ← **admitted** to hospital |
| Sports | **games** → **ended** season<br>PERSON be **sidelined** ← **undergoing** surgery<br>PERSON be **suspended** ← **testing** for cocaine<br>PERSON **returned** ← **recovering** from injury |
| Crime | **shooting** → PERSON be **arrested**<br>**spending** in jail → PERSON be **released**<br>PERSON be **arrested** ← **bombings**<br>driver **fled** ← **accident** |

Table 3: Examples of Learned Regular Event pairs. → represents *before* relation and ← represents *after* relation.

all kappa inter-agreement between the two annotators is 72%. We can see that with event argument generalization, the quality of acquired seed regular event pairs is roughly equal to that using specific name arguments. Furthermore, because we obtained two times of seed event pairs after using event argument generalization, the second and third bootstrapping systems received more guidance and continued to learn regular event pairs with a high quality. In addition, using dependency path based sentential contexts enables the classifier to further improve the accuracy of bootstrapped regular event pairs.

We have learned around 4,400 regular event pairs that are rich in commonsense knowledge and domain specific knowledge for domains including politics, business, health, sports and crime. Table 3 shows several examples in each category.

## 5.2 Weakly Supervised Contextual Temporal Relation Classifier

### 5.2.1 Accuracy of the Classifier

Recall that the contextual temporal relation classifier was trained on the New York Times section of Gigaword. In order to evaluate the accuracy of the classifier, we applied the weakly supervised learned classifier (the full system) to sentential contexts between pairs of events extracted from the Associated Press Worldstream section of Gigaword. We randomly sampled 100 instances from the ones that were labeled by the classifier as indicating a *after* or *before* relation and with a confidence score greater than 0.8. Then for each instance and its pair of events, we asked our two annotators to judge whether the sentence indeed describes a *after (before)* temporal relation between the two events. According to the annotations[7], the classifier predicted the correct temporal relation 74% of time.

### 5.2.2 Evaluation Using a Benchmark Dataset

To facilitate direct comparisons, we evaluate both our weakly supervised trained classifier and two supervised trained systems using a benchmark evaluation dataset, the TempEval-3-platinum corpus, which contains 20 news articles annotated with several temporal relations between events. We only evaluate system performance on identifying temporal "before/after" relations.

Table 4 shows the comparison results between these three systems. Note that we ran the original ClearTK system and we re-implemented the system described in (Mirza and Tonelli, 2014). In addition, both supervised systems were trained using TimeBank v1.2 (Pustejovsky et al., 2006). The performance across the three systems is overall low, one reason is that the pairs of events that are in a temporal relation were not provided to the classi-

---

[7] The two annotators achieved a Kappa inter-agreement score of 0.71.

| | Approaches | F1 | P | R |
|---|---|---|---|---|
| 1 | ClearTK (Bethard, 2013) | 0.27 | 0.36 | 0.22 |
| 2 | Mirza and Tonelli (2014) | 0.29 | 0.24 | 0.38 |
| 3 | Our classifier | 0.28 | 0.35 | 0.24 |

Table 4: Performance on TempEval-3 Test Data

fiers. Therefore, the classifiers had to identify temporally related event pairs as well as classify their temporal relations. We can see that the weakly supervised classifier achieved roughly equal performance as ClearTK, while the other supervised system presents a a different precision-recall tradeoff. Overall, without using any annotated data or sophisticated hand crafted features, our weakly supervised system achieved a F1-score comparable to both supervised trained systems.

## 6 Conclusion

We presented a weakly supervised bootstrapping approach that learns both regular event pairs and a contextual temporal relation classifier, by exploring the observation that regular event pairs tend to show a consistent temporal relation despite of their diverse contexts. Evaluation shows that the learned regular event pairs are of high quality and rich in commonsense knowledge and domain knowledge. In addition, the weakly supervised trained temporal relation classifier achieves comparable performance with state-of-the-art supervised classifiers.

## 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.

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.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Jennifer D'Souza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *HLT-NAACL*. pages 918–927.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving Neural Networks by Preventing Coadaptation of Feature Detectors. In *arXiv preprint arXiv:1207.0580*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of 2014 the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*.

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*. Association for Computational Linguistics, pages 284–291.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. pages 55–60.

Tara McIntosh and James R Curran. 2009. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, pages 396–404.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

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

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Association for Computational Linguistics, pages 95–100.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*. volume 2003, page 40.

James Pustejovsky, Marc Verhagen, Roser Saurí, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen, and Andrea Setzer. 2006. Timebank 1.2. *Linguistic Data Consortium* 40.

Stephanie Strassel, Mark A Przybocki, Kay Peterson, Zhiyi Song, and Kazuaki Maeda. 2008. Linguistic Resources and Evaluation Techniques for Evaluation of Cross-Document Automatic Content Extraction. In *Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC-08)*.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3 evaluating time expressions, events, and temporal relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*.

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

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*. Association for Computational Linguistics, pages 57–62.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.