Temporal Information Extraction for Question Answering Using Syntactic Dependencies in an LSTM-based Architecture

Yuanliang Meng, Anna Rumshisky, Alexey Romanov
{ymeng,arum,aromanov}@cs.uml.edu
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854

Abstract

In this paper, we propose to use a set of simple, uniform in architecture LSTM-based models to recover different kinds of temporal relations from text. Using the shortest dependency path between entities as input, the same architecture is implemented to extract intra-sentence, cross-sentence, and document creation time relations. A "double-checking" technique reverses entity pairs in classification, boosting the recall of positive cases and reducing misclassifications between opposite classes. An efficient pruning algorithm resolves conflicts globally. Evaluated on QA-TempEval (SemEval2015 Task 5), our proposed technique outperforms state-of-the-art methods by a large margin. We also conduct intrinsic evaluation and post state-of-the-art results on Timebank-Dense.

1 Introduction

Recovering temporal information from text is essential to many text processing tasks that require deep language understanding, such as answering questions about the timeline of events or automatically producing text summaries. This work presents intermediate results of an effort to build a temporal reasoning framework with contemporary deep learning techniques.

Until recently, there has been remarkably few attempts to evaluate temporal information extraction (TemporalIE) methods in context of downstream applications that require reasoning over the temporal representation. One recent effort to conduct such evaluation was SemEval2015 Task 5, a.k.a. QA-TempEval (Llorens et al., 2015a), which used question answering (QA) as the target application. QA-TempEval evaluated systems producing TimeML (Pustejovsky et al., 2003) annotation based on how well their output could be used in QA. We believe that application-based evaluation of TemporalIE should eventually completely replace the intrinsic evaluation if we are to make progress, and therefore we evaluated our techniques mainly using QA-TempEval setup.

Despite the recent advances produced by multi-layer neural network architectures in a variety of areas, the research community is still struggling to make neural architectures work for linguistic tasks that require long-distance dependencies (such as discourse parsing or coreference resolution). Our goal was to see if a relatively simple architecture with minimal capacity for retaining information was able to incorporate the information required to identify temporal relations in text.

Specifically, we use several simple LSTM-based components to recover ordering relations between temporally relevant entities (events and temporal expressions). These components are fairly uniform in their architecture, relying on dependency relations recovered with a very small number of mature, widely available processing tools, and require minimal engineering otherwise. To our knowledge, this is the first attempt to apply such simplified techniques to the TemporalIE task, and we demonstrate this streamlined architecture is able to outperform state-of-the-art results on a temporal QA task with a large margin.

In order to demonstrate generalizability of our proposed architecture, we also evaluate it intrinsically using TimeBank-Dense[1] (Chambers et al., 2014). TimeBank-Dense annotation aims to approximate a complete temporal relation graph by including all intra-sentential relations, all relations between adjacent sentences, and all relations with document creation time. Although our system

[1] https://www.usna.edu/Users/cs/nchamber/caevo/#corpus

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 887–896
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

was not optimized for such a paradigm, and this data is quite different in terms of both the annotation scheme and the evaluation method, we obtain state-of-the-art results on this corpus as well.

2 Related Work

A multitude of TemporalIE systems have been developed over the past decade both in response to the series of shared tasks organized by the community (Verhagen et al., 2007, 2010; UzZaman et al., 2012; Sun et al., 2013; Bethard et al., 2015; Llorens et al., 2015b; Minard et al., 2015) and in standalone efforts (Chambers et al., 2014; Mirza, 2016).

The best methods used by TemporalIE systems to date tend to rely on highly engineered task-specific models using traditional statistical learning, typically used in succession (Sun et al., 2013; Chambers et al., 2014). For example, in a recent QA-TempEval shared task, the participants routinely used a series of classifiers (such as support vector machine (SVM) or hidden Markov chain SVM) or hybrid methods combining hand crafted rules and SVM, as was used by the top system in that challenge (Mirza and Minard, 2015). While our method also relies on decomposing the temporal relation extraction task into subtasks, we use essentially the same simple LSTM-based architecture for different components, that consume a highly simplified representation of the input.

Although there has not been much work applying deep learning techniques to TemporalIE, some relevant work has been done on a similar (but typically more local) task of relation extraction. Convolutional neural networks (Zeng et al., 2014) and recurrent neural networks both have been used for argument relation classification and similar tasks (Zhang and Wang, 2015; Xu et al., 2015; Vu et al., 2016). We take inspiration from some of this work, including specifically the approach proposed by Xu et al. (2015) which uses syntactic dependencies.

3 Dataset

We used QA-TempEval (SemEval 2015 Task 5)[2] data and evaluation methods in our experiments. The training set contains 276 annotated TimeML files, mostly news articles from major agencies or Wikinews from late 1990s to early 2000s. This data contains annotations for events, temporal expressions (referred to as TIMEXes), and temporal relations (referred to as TLINKs). The test set contains unannotated files in three genres: 10 news articles composed in 2014, 10 Wikipedia articles about world history, and 8 blogs entries from early 2000s.

In QA-TempEval, evaluation is done via a QA toolkit which contains yes/no questions about temporal relations between two events or an event and a temporal expression. QA evaluation is not available for most of the training data except for 25 files, for which 79 questions are available. We used used this subset of the training data for validation. The test set contains unannotated files, so QA is the only way to measure the performance. The total of 294 questions is available for the test data, see Table 6.

We also use TimeBank-Dense dataset, which contains a subset of the documents in QA-TempEval. In TimeBank-Dense, all entity pairs in the same sentence or in consecutive sentences are labeled. If there is no information about the relation between two entities, it is labeled as "vague". We follow the experimental setup in (Chambers et al., 2014), which splits the corpus into training/validation/test sets of 22, 5, and 9 documents, respectively.

4 TIMEX and Event Extraction

The first task in our TemporalIE pipeline (TEA) is to identify time expressions (TIMEXes) and events in text. We utilized the HeidelTime package (Strötgen and Gertz, 2013) to identify TIMEXes. We trained a neural network model to identify event mentions. Contrary to common practice in TemporalIE, our models do not rely on event attributes, and thus we did not attempt to identify them.

Feature	Explanation
is_main_verb	whether the token is the main verb of a sentence
is_predicate	whether the token is the predicate of a phrase
is_verb	whether the token is a verb
is_noun	whether the token is a noun

Table 1: Token features for event extraction

We perform tokenization, part-of-speech tagging, and dependency parsing using NewsReader (Agerri et al., 2014). Every token is represented with a set of features derived from preprocessing. Syntactic dependencies are not used for event extraction, but are used later in the pipeline for

[2] http://alt.qcri.org/semeval2015/task5/

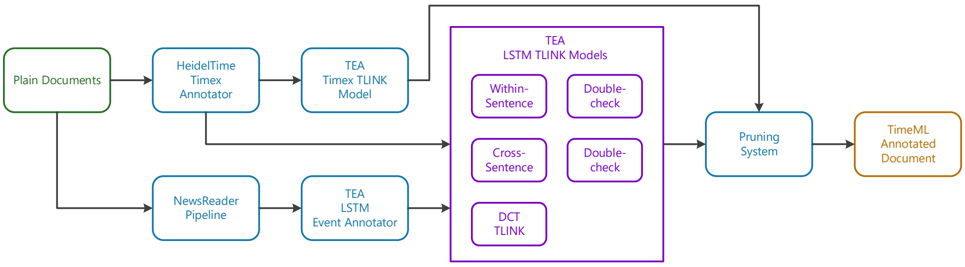

Figure 1: System overview for our temporal extraction annotator (TEA) system

TLINK classification. The features used to identify events are listed in Table 1.

The event extraction model uses long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), an RNN architecture well-suited for sequential data. The extraction model has two components, as shown on the right of Figure 2. One component is an LSTM layer which takes word embeddings as input. The other component takes 4 token-level features as input. These components produce hidden representations which are concatenated, and fed into an output layer which performs binary classification. For each token, we use four tokens on each side to represent the surrounding context. The resulting sequence of nine word embeddings is then used as input to an LSTM layer. If a word is near the edge of a sentence, zero padding is applied. We only use the token-level features of the target token, and ignore those from the context words. The 4 features are all binary, as shown in Table 1. Since the vast majority of event mentions in the training data are single words, we only mark single words as event mentions.

5 TLINK Classification

Our temporal relation (TLINK) classifier consists of four components: an LSTM-based model for intra-sentence entity relations, an LSTM-based model for cross-sentence relations, another LSTM-based model for relations with document creation time, and a rule-based component for TIMEX pairs. The four models perform TLINK classifications independently, and the combined results are fed into a pruning module to remove the conflicting TLINKs. The three LSTM-based components use the same streamlined architecture over token sequences recovered from shortest dependency paths between entity pairs.

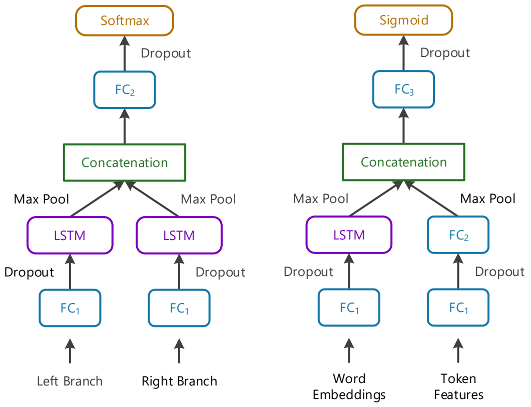

Figure 2: Model architecture. Left: intra-sentence and cross-sentence model. Right: Event extraction model.

5.1 Intra-Sentence Model

A TLINK extraction model should be able to learn the patterns that correspond to specific temporal relations, such as specific temporal prepositional phrases and clauses with temporal conjunctions. This suggests such models may benefit from encoding syntactic relations, rather than linear sequences of lexical items.

We use the shortest path between entities in a dependency tree to capture the essential context. Using the NewsReader pipeline, we identify the shortest path, and use the word embeddings for all tokens in the path as input to a neural network. Similar to previous work in relation extraction (Xu et al., 2015), we use two branches, where the left branch processes the path from the source entity to the least common ancestor (LCA), and the right branch processes the path from the target entity to the LCA. However, our TLINK extraction model uses only word embeddings as input, not POS tags, grammatical relations themselves, or WordNet hypernyms.

For example, for the sentence "Their marriage ended before the war", given an event pair (*marriage*, *war*), the left branch of the model will receive the sequence (*marriage, ended*), while the right branch will receive (*war, before, ended*). The LSTM layer processes the appropriate sequence of word embeddings in each branch. This is followed by a separate max pooling layer for each branch, so for each LSTM unit, the maximum value over the time steps is used, not the final step value. During the early stages of model design, we observed that this max pooling approach (also used in Xu et al. (2015)) resulted in a slight improvement in performance. Finally, the results from the max pooling layers of both branches are concatenated and fed to a hidden layer, followed by a softmax to yield a probability distribution over the classes. The model architecture is shown in Figure 2 (left). We also augment the training data by flipping every pair, i.e. if $(e_1, e_2) \rightarrow$ BEFORE, $(e_2, e_1) \rightarrow$ AFTER is also included.

5.2 Cross-Sentence Model

TLINKs between the entities in consecutive sentences can often be identified without any external context or prior knowledge. For example, the order of events may be indicated by discourse connectives, or the events may follow natural order, potentially encoded in their word embeddings.

To recover such relations, we use a model similar to the one used for intra-sentence relations, as described in Section5.1. Since there is no common root between entities in different sentences, we use the path between an entity and the sentence root to construct input data. A sentence root is often the main verb, or a conjunction.

5.3 Relations to DCT

The document creation time (DCT) naturally serves as the "current time". In this section, we discuss how to identify temporal relations between an event and DCT. The assumption here is that an event mention and its local context can often suffice for DCT TLINKs. For example, English has inflected verbs for tense in finite clauses, and uses auxiliaries to express aspects.

The model we use is again similar to the one in Section5.2. Although one branch would suffice in this case, we use two branches in our implementation. One branch processes the path from a given entity to the sentence root, and the other branch processes the same path in reverse, from the root to the entity.

5.4 Relations between TIMEXes

Time expressions explicitly signify a time point or an interval of time. Without the TIMEX entities serving as "hubs", many events would be isolated from each other. We use rule-based techniques to identify temporal relations between TIMEX pairs that have been identified and normalized by HeidelTime. The relation between the DCT and other time expressions is just a special case of TIMEX-to-TIMEX TLINK and is handled with rules as well.

DATE value	Calculation	Representation

2017-08-04	START = 2017 + 7/12 + 3/365
	= 2017.591
	END = START
		(2017.591, 2017.591)

2017-SU
(Summer 2017)
	START = 2017 + 5/12 = 2017.416
	END = 2017 + 8/12 = 2017.666
		(2017.416, 2017.666)

Table 2: Examples of DATE values and their tuple representations

In the present implementation, we focus on the DATE class of TIMEX tags, which is prevalent in the newswire text. The TIME class tags which contain more information are converted to DATE. Every DATE value is mapped to a tuple of real values (*start*, *end*). The "value" attribute of TIMEX tags follows the ISO-8601 standard, so the mapping is straightforward. Table 2 provides some examples. We set the minimum time interval to be a day. Practically, such a treatment suffices for our data. After mapping DATE values to tuples of real numbers, we can define 5 relations between TIMEX entities $T_1 = (start_1, end_1)$ and $T_2 = (start_2, end_2)$ as follows:

$$T_1 \times T_2 \rightarrow \begin{cases} \text{BEFORE} & \text{if } end_1 < start_2 \\ \text{AFTER} & \text{if } start_1 > end_2 \\ \text{INCLUDES} & \text{if } start_1 < start_2 \\ & \text{ and } end_1 > end_2 \\ \text{IS\_INCLUDED} & \text{if } start_1 > start_2 \\ & \text{ and } end_1 < end_2 \\ \text{SIMULTANEOUS} & \text{if } start_1 = start_2 \\ & \text{ and } end_1 = end_2 \end{cases} \tag{1}$$

The TLINKs from training data contain more types of relations than the five described in Equation 1. However relations such as IBEFORE ("immediately before"), IAFTER("immediately after") and IDENTITY are only used on event pairs, not TIMEX pairs. The QA system also does not target questions on TIMEX pairs. The purpose here is to use the TIMEX relations to link the otherwise isolated events.

6 Double-checking

A major difficulty we have is that the TLINKs for intra-sentence, cross-sentence, and DCT relations in the training data are not comprehensive. Often, the temporal relation between two entities is clear, but the training data provides no TLINK annotation. We downsampled the NO-LINK class in training in order to address both the class imbalance and the fact that TimeML-style annotation is de-facto sparse, with only a fraction of positive instances annotated.

In addition to that, we introduce a technique to boost the recall of positive classes (not NO-LINK) and to reduce the misclassification between the opposite classes. Since entity pairs are always classified in both orders, if both orders produce a TLINK relation, rather than NO-LINK, we adopt the label with a higher probability score, as assigned by the softmax classifier. We call this technique "double-checking". It serves to reduce the errors that are fundamentally harmful (e.g. BEFORE misclassified as AFTER, and vice versa). We also allow a positive class to have the "veto power" against NO-LINK class. For instance, if our model predicts $(e_1, e_2) \rightarrow$ AFTER but NO-LINK reversely, we adopt the former.

NO-LINK ratio	Recall	Recall	BEFORE	AFTER
	BEFORE	AFTER	as AFTER	as BEFORE
0.5	0.451	0.445	0.075	0.092
0.1	0.643	0.666	0.145	0.159
0.1 + double-check	0.721	0.721	0.125	0.125

Table 3: Effects of downsampling and double-checking on intra-sentence results. 0.5 NO-LINK ratio means that NO-LINKs are downsampled to a half of the number of all positive instances combined. BEFORE as AFTER shows the fraction of BEFORE misclassified as AFTER, and vice versa.

Table 3 shows the effects of double-checking and downsampling the NO-LINK cases on the intra-sentence model. Double-checking technique not only further boosts recall, but also reduces the misclassification between the opposite classes.

7 Pruning TLINKs

The four TLINK classification models in Section 5 deal with different kinds of TLINKs, so their output does not overlap. Nevertheless temporal relations are transitive in nature, so the deduced relations from given TLINKs can be in conflict.

Most conflicts arise from two types of relations, namely BEFORE/AFTER and INCLUDES/IS_INCLUDED. Naturally, we can convert TLINKs of opposite relations and put them all together. If we use a directed graph to represent the BEFORE relations between all entities, it should be acyclic. Sun (2014) proposed a strategy that "prefers the edges that can be inferred by other edges in the graph and remove the ones that are least so". Another strategy is to use the results from separate classifiers or "sieves" to rank TLINKs according to their confidence (Mani et al., 2007; Chambers et al., 2014). High-ranking results overwrite low-ranking ones.

We follow the same idea of purging the weak TLINKs. Given a directed graph, our approach is to remove the edges to break cycles, so that the sum of weights from the removed edges is minimal. This problem is actually an extension of the minimum feedback arc set problem and is NP-hard (Karp, 1972). We therefore adopt a heuristic-based approach, applied separately to the graphs induced by BEFORE/AFTER and INCLUDES/IS_INCLUDED relations.[3] The softmax layer provides a probability score for each relation class, which represents the strength of a link. TLINKs between TIMEX pairs are generated by rules, so we assume them to be reliable and assign them a score of 1. Although INCLUDES/IS_INCLUDED edges can generate conflicts in a BEFORE/AFTER graph as well, we currently do not resolve such conflicts because they are relatively rare. We also do not use SIMULTANEOUS/IDENTITY relations to merge nodes, because we found that it leads to very unstable results.

For a given relation (e.g., BEFORE), we incrementally build a directed graph with all edges representing that relation. We first initialize the graph with TIMEX-to-TIMEX relations. Event vertices are then added to this graph in a random order. For each event, we add all edges associated with it. If this creates a cycle, the edges are removed one by one until there is no cycle, keeping track of the sum of the scores associated with removed edges. We choose the order in which the edges are removed to minimize that value.[4] The algorithm is shown above.

In practice, the vertices do not have a high de-

[3] We found that ENDS and BEGINS TLINKs are too infrequent to warrant a separate treatment.

[4] By removing an edge, we mean resetting the relation to NO-LINK. Another possibility may be to set the relation associated with the edge to the one with the second highest probability score, however this may create additional cycles.

```
$X \leftarrow$ EVENTS;
$V \leftarrow$ TIMEXes;
$E \leftarrow$ TIMEX pairs;
Initialize $G \leftarrow < V, E >$;
for $x \in X$ do
    $V' \leftarrow V + \{x\}$;
    $C \leftarrow \{(x, v) \cup (v, x) | v \in V\}$ ;
    $E' \leftarrow E \cup C$ ;
    $G' \leftarrow < V', E' >$ ;
    if cycle_exists(G') then
        for $C_i \in \pi(C)$ do
            $score_i = 0$;
            while $C_i \neq \phi$ & $cycle\_exists(G \cup C_i)$
              do
                $c \leftarrow C_i.pop()$;
                $score_i += weight(c)$;
            end
        end
    end
    $G \leftarrow G \cup C_i$ s.t. $i = argmin(score_i)$;
end
```

Algorithm 1: Algorithm to prune edges. $\pi(C)$ denotes some permutations of C, where C is a list of weighted edges.

gree for a given relation, so permuting the candidates $N \times (N - 1)$ times (i.e., not fully), where N is the number of candidates, produces only a negligible slowdown. We also make sure to try the greedy approach, dropping the edges with the smallest weights first.

8 Model Settings

In this section, we describe the model settings used in our experiments. All models requiring word embeddings use 300-dimensional word2vec vectors trained on Google News corpus (3 billion running words).[5] Our models are written in Keras on top of Theano.

TIMEX and Event Annotation The LSTM layer of the event extraction model contains 128 LSTM units. The hidden layer on top of that has 30 neurons. The input layer corresponding to the 4 token features is connected with a hidden layer with 3 neurons. The combined hidden layer is then connected with a single-neuron output layer. We set a dropout rate 0.5 on input layer, and another drop out rate 0.5 on the hidden layer before output.

As mentioned earlier, we do not attempt to tag event attributes. Since the vast majority of tokens are outside of event mention boundaries, we set higher weights for the positive class. In order to answer questions about temporal relations, it is not particularly harmful to introduce spurious events, but missing an event makes it impossible to answer any question related to it. Therefore we intentionally boost the recall while sacrificing precision. Table 4 shows the performance of our event extraction, as well as the performance of HeidelTime TIMEX tagging. For events, partial overlap of mention boundaries is considered an error.

Annotation	Prec	Rec	F1
TIMEX	0.838	0.850	0.844
Event	0.729	0.963	0.830

Table 4: TIMEX and event evaluation on validation set.

Intra-Sentence Model We identify 12 classes of temporal relations, plus a NO-LINK class. For training, we downsampled NO-LINK class to 10% of the number of positive instances. Our system does not attempt to resolve coreference. For the purpose of identifying temporal relations, SIMULTANEOUS and IDENTITY links capture the same relation of simultaneity, which allowed us to combine them. The LSTM layer of the intra-sentence model contains 256 LSTM units on each branch. The hidden layer on top of that has 100 neurons. We set a dropout rate 0.6 on input layer, and another drop out rate 0.5 on the hidden layer before output.

Cross-Sentence Model The training and evaluation procedures are very similar to what we did for intra-sentence models, and the hyperparameters for the neural networks are the same. Now the vast majority of entity pairs have no TLINKs explicitly marked in training data. Unlike the intra-sentence scenario, however, a NO-LINK label is truly adequate in most cases. We found that downsampling NO-LINK instances to match the number of all positive instances (ratio=1) yields desirable results. Since positive instances are very sparse in both the training and validation data, the ratio should not be too low, so as not to risk overfitting.

DCT Model We use the same hyperparameters for the DCT model as for the intra-sentence and cross-sentence models. Again, the training files do not sufficiently annotate TLINKs with DCT even if the relations are clear, so there are many false negatives. We downsample the NO-LINK instances so that they are 4 times the number of positive instances.

[5] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

system	coverage	prec	rec	f1
human-fold1-original	0.43	0.91	0.38	0.54
human-fold1-timlinks	0.52	0.93	0.47	**0.62**
TIPSem-fold1-original	0.35	0.57	0.22	0.32
TIPSem-fold1-timex	0.53	0.69	0.38	0.50
orig. validation data	0.37	**0.93**	0.34	0.50
orig. tags TEA tlinks	0.81	0.58	0.47	0.52
TEA-initial	0.78	0.60	0.47	0.52
TEA-double-check	**0.89**	0.60	**0.53**	0.56
TEA-prune	0.82	0.58	0.48	0.53
TEA-flat	0.81	0.44	0.35	0.39
TEA-Dense	0.68	0.70	0.48	0.57
TEA-final	0.84	0.64	**0.53**	**0.58**

Table 5: QA results on validation data. There are **79** questions in total. The 4 systems on the top of the table are provided with the toolkit. The systems starting with "human-" are annotated by human experts. TEA-final utilizes both double-check and pruning. TEA-flat uses the flat context. TEA-Dense is trained on TimeBank-Dense.

9 Experiments

In this section, we first describe the model selection experiments on QA-TempEval validation data, selectively highlighting results of interest. We then present the results obtained with the optimized model on the QA-TempEval task and on TimeBank-Dense.

9.1 Model Selection Experiments

As mentioned before, "gold" TLINKs are sparse, so we cannot merely rely on the F1 scores on validation data to do model selection. Instead, we used the QA toolkit. The toolkit contains 79 yes-no questions about temporal relations between entities in the validation data. Originally, only 6 questions have "no" as the correct answer, and 1 question is listed as "unknown". After investigating the questions and answers, however, we found some errors and typos[6]. After fixing the errors, there are 7 no-questions and 72 yes-questions in total. All evaluations are performed on the fixed data.

The evaluation tool draws answers from the annotations only. If an entity (event or TIMEX) involved in a question is not annotated, or the TLINK cannot be found, the question will then be counted as not answered. There is no way for participants to give an answer directly, other than delivering the annotations. The program generates Timegraphs to infer relations from the annotated TLINKs. As a result, relations without explicit TLINK labels can still be used if they can be inferred from the annotations. The QA toolkit uses the following evaluation measures:

$$\text{coverage} = \frac{\#\text{answered}}{\#\text{questions}},\ \text{precision} = \frac{\#\text{correct}}{\#\text{answered}}$$

$$\text{recall} = \frac{\#\text{correct}}{\#\text{questions}},\ \text{f1} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$$

Table 5 shows the results produced by different models on the validation data. The results of the four systems above the first horizontal line are provided by the task organizer. Among them, the top two use annotations provided by human experts. As we can see, the precision is very high, both above 0.90. Our models cannot reach that precision. In spite of the lower precision, automated systems can have much higher coverages i.e. answer a lot more questions.

As a starting point, we evaluated the validation files in their original form, and the results are shown as "orig. validation data" of Table 5. The precision was good, but with very low coverage. This supports our claim that the TLINKs provided by the training/validation files are not complete. We also tried using the event and TIMEX tags from the validation data, but performing TLINK classification with our system. As shown with "orig. tags TEA tlinks" in the table, now the coverage rises to 64 (or 0.81), and the overall F1 score reaches 0.52. The TEA-initial system uses our own annotators. The performance is similar, with a slight improvement in precision. This result shows our event and TIMEX tags work well, and are not inferior to the ones provided by the training data.

The double-checking technique boosts the coverage a lot, probably because we allow positive results to veto NO-LINKs. Combining double-checking with the pruning technique yields the best results, with F1 score 0.58, answering 42 out of 79 questions correctly.

In order to validate the choice of the dependency path-based context, we also experimented with a conventional flat context window, using the same hyperparameters. Every entity is represented by a 11-word window, with the entity mention in the middle. If two entities are near each other, their windows are cut short before reaching the other entity. Using the flat context instead of dependency paths yields a much weaker performance.

[6] Question 24 from XIE19980821.0077.tml should be answered with "yes", but the answer key contains a typo "is". Question 34 from APW19980219.0476.tml has BEFORE that should be replaced with AFTER. Question 29 from XIE19980821.0077.tml has "unknown" in the answer key, but after reading the article, we believe the correct answer is "no".

This confirms our hypothesis that syntactic dependencies represent temporal relations better than word windows. However, it should be noted that we did not separately optimize the models for the flat context setting. The large performance drop we saw from switching to flat context did not warrant performing a separate parameter search.

We also wanted to check whether a comprehensive annotation of TLINKs in the training data can improve model performance on the QA task. We therefore trained our model on TimeBank-Dense data and evaluated it with QA (see the TEA-Dense line in Table 5). Interestingly, the performance is nearly as good as our top model, although TimeBank-Dense only uses five major classes of relations. For one thing, it shows that our system may perform equally after being trained on sparsely labeled data and on densely labeled data, judged from the QA evaluation tool. If this is true, excessively annotated data may not be necessary in some tasks.

	doc	**words**	**quest**	**yes**	**no**	**dist-**	**dist+**
news	10	6920	99	93	6	40	59
wiki	10	14842	130	117	13	58	72
blogs	8	2053	65	65	0	30	35
total	28	23815	294	275	19	128	166

Table 6: Test data statistics. Adapted from Table 1 in Llorens et al. (2015a).

9.2 QA-TempEval Experiments

We use the QA toolkit provided by the QA-TempEval organizers to evaluate our system on the test data. The documents in test data are not annotated at all, so the event tags, TIMEX tags, and TLINKs are all created by our system.

Table 6 shows the the statistics of test data. As we can see, the vast majority of the questions in the test set should be answered with *yes*. Generally speaking, it is much more difficult to validate a specific relation (answer *yes*) than to reject it (answer *no*) when we have as many as 12 types of relations in addition to the vague NO-LINK class. **dist-** means questions involving entities that are in the same sentence or in consecutive sentences. **dist+** means the entities are farther away.

The QA-TempEval task organizers used two evaluation methods. The first method is exactly the same as the one we used on validation data. The second method used a so-called Time Expression Reasoner (TREFL) to add relations between TIMEXes, and evaluated the augmented results. The goal of such an extra run is to "analyze how a general time expression reasoner could improve results". Our model already includes a component to handle TIMEX relations, so we will compare our results with other systems' in both methods.

News Genre (99 questions)

system	**prec**	**rec**	**f1**	**% answd**	**# correct**
hlt-fbk-ev1-trel1	0.59	0.17	0.27	29	17
hlt-fbk-ev1-trel2	0.43	0.23	0.30	55	23
hlt-fbk-ev2-trel1	0.56	0.20	0.30	36	20
hlt-fbk-ev2-trel2	0.43	0.29	0.35	69	29
ClearTK	0.60	0.06	0.11	10	6
CAEVO	0.59	0.17	0.27	29	17
TIPSemB	0.50	0.16	0.24	32	16
TIPSem	0.52	0.11	0.18	21	11
TEA	**0.61**	**0.44**	**0.51**	**73**	**44**

Wikipedia Genre (130 questions)

system	**prec**	**rec**	**f1**	**% answd**	**# correct**
hlt-fbk-ev1-trel1	0.55	0.16	0.25	29	21
hlt-fbk-ev1-trel2	0.52	0.22	0.35	50	34
hlt-fbk-ev2-trel1	0.58	0.17	0.26	29	22
hlt-fbk-ev2-trel2	0.62	0.36	0.46	58	47
ClearTK	0.60	0.05	0.09	8	6
CAEVO	0.59	0.17	0.26	28	22
TIPSemB	0.52	0.13	0.21	25	17
TIPSem	**0.74**	0.19	0.30	26	25
TEA	0.62	**0.44**	**0.51**	**71**	**57**

Blog Genre (65 questions)

system	**prec**	**rec**	**f1**	**% answd**	**# correct**
hlt-fbk-ev1-trel1	**0.57**	0.18	**0.28**	32	12
hlt-fbk-ev1-trel2	0.43	0.18	0.26	43	12
hlt-fbk-ev2-trel1	0.47	0.14	0.21	29	9
hlt-fbk-ev2-trel2	0.34	**0.20**	0.25	**58**	**13**
ClearTK	0.56	0.08	0.14	14	5
CAEVO	0.48	0.18	0.27	38	12
TIPSemB	0.31	0.08	0.12	25	5
TIPSem	0.45	0.14	0.21	31	9
TEA	0.43	**0.20**	0.27	46	**13**

Table 7: QA evaluation on test data without TREFL

The results are shown in Table 7. We give the results for the hlt-fbk systems that were submitted by the top team. Among them, hlt-fbk-ev2-trel2 was the overall winner of TempEval task in 2015. ClearTK, CAEVO, TIPSEMB and TIPSem were some off-the-shelf systems provided by the task organizers for reference. These systems were not optimized for the task (Llorens et al., 2015a).

For news and Wikipedia genres, our system outperforms all other systems by a large margin. For blogs genre, however, the advantage of our system is unclear. Recall that our training set contains news articles only. While the trained model works well on Wikipedia dataset too, blog dataset is fundamentally different in the following ways: (1) each blog article is very short, (2) the style of writing in blogs is much more informal, with non-standard spelling and punctuation, and (3) blogs

All Genres (294 questions)

system	prec	rec	f1	% awd	# corr
hlt-fbk-ev2-trel2	0.49	0.30	0.37	62	89
hlt-fbk-ev2-trel2-TREFL	0.51	0.34	0.40	**67**	99
TEA	**0.59**	**0.39**	**0.47**	66	**114**
TEA-TREFL	0.58	0.38	0.46	66	111

Table 8: Test results over all genres.

are written in first person, and the content is usually personal stories and feelings.

Interestingly, the comparison between different hlt-fbk submissions suggests that resolving event coreference (implemented by hlt-fbk-ev2-trel2) substantially improves system performance for the news and Wikipedia genres. However, although our system does not attempt to handle event coreference explicitly, it easily outperforms the hlt-fbk-ev2-trel2 system in the genres where coreference seems to matter the most.

Evaluation with TREFL The extra evaluation with TREFL has a post-processing step that adds TLINKs between TIMEX entities. Our model already employs such a strategy, so this post-processing does not help. In fact, it drags down the scores a little. Table 8 summarizes the results over all genres before and after applying TREFL. For comparison, we include the top 2015 system, hlt-fbk-ev2-trel2. As we can see, TEA generally shows substantially higher scores.

9.3 TimeBank-Dense Experiments

We trained and evaluated the same system on TimeBank-Dense to see how it performs on a similar task with a different set of labels and another method of evaluation. In this experiment, we used the event and TIMEX tags from test data, as Mirza and Tonelli (2016).

Since all the NO-LINK (vague) relations are labeled, downsampling was not necessary. We did use double-checking in the final conflict resolution, but without giving positive cases the veto power over NO-LINK. Because NO-LINK relations dominate, especially for cross-sentence pairs, we set class weights to be inversely proportional to the class frequencies during training. We also reduced input batch size to counteract class imbalance.

We ran two sets of experiments. One used the uniform configurations for all the neural network models, similar to our experiments with QA-TempEval. The other tuned the hyperparameters for each component model (number of neurons, dropout rates, and early stop) separately.

system	ClearTK	NavyT	CAEVO	CATENA	TEA-Dense	
					uniform	**tuned**
F1	0.447	0.453	0.507	0.511	0.505	0.519

Table 9: TEA results on TimeBank-Dense. ClearTK, NavyT, and CAEVO are systems from Chambers et al. (2014). CATENA is from Mirza and Tonelli (2016)

The results from TimeBank-Dense are shown in Talble 9. Even though TimeBank-Dense has a very different methodology for both annotation and evaluation, our “out-of-the-box” model which uses uniform configurations across different components obtains F1 0.505, compared to the best F1 of 0.511 in previous work. Our best result of 0.519 is obtained by tuning hyperparameters on intra-sentence, cross-sentence, and DCT models independently.

For the QA-TempEval task, we intentionally tagged a lot of events, and let the pruning algorithm resolve potential conflicts. In the TimeBank-Dense experiment, however, we only used the provided event tags, which are sparser than what we have in QA-TempEval. The system may have lost some leverage that way.

10 Conclusion

We have proposed a new method for extraction of temporal relations which takes a relatively simple LSTM-based architecture, using shortest dependency paths as input, and re-deploys it in a set of subtasks needed for extraction of temporal relations from text. We also introduce two techniques that leverage confidence scores produced by different system components to substantially improve the results of TLINK classification: (1) a “double-checking” technique which reverses pairs in classification, thus boosting the recall of positives and reducing misclassifications among opposite classes and (2) an efficient pruning algorithm to resolve TLINK conflicts. In a QA-based evaluation, our proposed method outperforms state-of-the-art methods by a large margin. We also obtain state-of-the art results in an intrinsic evaluation on a very different TimeBank-Dense dataset, proving generalizability of the proposed model.

Acknowledgments

This project is funded in part by an NSF CAREER Award to A. Rumshisky (IIS-1652742). We would like to thank Connor Cooper and Kevin Wacome for their contributions to the early stages of this work.

References

Rodrigo Agerri, Josu Bermudez, and German Rigau. 2014. Ixa pipeline: Efficient and ready to use multilingual nlp tools. In *Proc. of the 9th Language Resources and Evaluation Conference (LREC2014)*, pages 26–31.

Steven Bethard, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2015. Semeval-2015 task 6: Clinical tempeval. In *Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 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.

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

Richard Karp. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations, Proc. Sympos.*, pages 85–103.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015a. Semeval-2015 task 5: Qa tempeval - evaluating temporal information understanding with question answering. In *Proc. of the 9th International Workshop on Semantic Evaluation*, pages 46–54. Association for Computational Linguistics.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015b. Semeval-2015 task 5: Qa tempeval-evaluating temporal information understanding with question answering. In *Proc. of the International Workshop on Semantic Evaluation (SemEval-2015)*.

Inderjeet Mani, Ben Wellner, Marc Verhagen, and James Pustejovsky. 2007. Three approaches to learning tlinks in timeml. *Technical Report CS-07–268, Computer Science Department*.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, Rubén Urizar, and Fondazione Bruno Kessler. 2015. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *Proc. of the International Workshop on Semantic Evaluation (SemEval-2015)*.

P Mirza and S Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *The 26th International Conference on Computational Linguistics*, pages 64–75. Association for Computational Linguistics.

Paramita Mirza. 2016. Extracting temporal and causal relations between events. *CoRR*, abs/1604.08120.

Paramita Mirza and Anne-Lyse Minard. 2015. Hlt-fbk: a complete temporal processing system for qa tempeval. In *Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 801–805. Association for Computational Linguistics.

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

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Weiyi Sun. 2014. *Time Well Tell: Temporal Reasoning in Clinical Narratives*. PhD dissertation. Department of Informatics, University at Albany, SUNY.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

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*.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proc. 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 *Proc. of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. *CoRR*, abs/1605.07333.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proc. of EMNLP 2015*, pages 1785–1794. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proc. of COLING 2014*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *CoRR*, abs/1508.01006.