Cross-Document Temporal Relation Extraction with Temporal Anchoring Events

Anonymous ACL submission

Abstract

Automatically extracting a timeline on a certain topic from multiple documents has been a challenge in natural language processing, partly due to the difficulty of collecting large amounts of training data. In this work, we collect a dataset for cross-document timeline extraction from online news that gives access to metadata such as hyperlinks and publication dates. The metadata allows us to define a set of important events while linking them to time anchors, which opens the opportunity to scale up data collection. Furthermore, with this set of linked news articles, we propose a method to enhance the inference process of temporal relation prediction, by utilizing a model to link events to a set of anchoring events that are added to the inference program. We report performance of common neural models and show that our method can boost the performance of all baseline models.

1 Introduction

The problem of representing temporal knowledge and performing temporal reasoning appears in multiple disciplines, such as philosophy, linguistics, and artificial intelligence. In natural language processing, multiple aspects of temporal understanding have been explored, including but not limited to identification and normalization of temporal expressions (Strötgen and Gertz, 2010; Lee et al., 2014), temporal ordering (Chambers et al., 2014; Leeuwenberg and Moens, 2018), and temporal commonsense knowledge like typical time and frequency (Zhou et al., 2019). A fundamental task in temporal processing from natural language that is commonly studied is temporal relation (TempRel) extraction (Verhagen et al., 2007, 2010), which determines the relative order of events. Combined with the tasks of identifying relevant events and explicit temporal expressions from text, this could provide a complete picture of the temporal sequence of events (UzZaman et al., 2013).

For TempRel annotations, however, the process is known to be time-consuming and difficult, as inter-annotator agreements are usually low (UzZaman et al., 2013; Ning et al., 2018). Attempts have been made to improve the process, however the fundamental problem of annotations still exists, and this makes TempRel datasets relatively unscalable in the current state of training large deep learning models with large datasets (Devlin et al., 2019).

Furthermore, current TempRel formulations mostly focus on events that appear under the same context, usually near each other in terms of position. For example, the TimeBank dataset (Pustejovsky et al., 2003), for which several commonly used baselines are based upon, considers only relations between events and expressions that appear within adjacent paragraphs. While these densely annotated datasets are suitable for evaluating the comprehensiveness of complete temporal understanding of a single document, there has not been that many attempts made on tackling the problem of extracting TempRels across multiple documents (Minard et al., 2015; Caselli and Vossen, 2017; Reimers et al., 2018). This would be useful in constructing timelines automatically from a set of documents on a topic or a keyword, which could be more viable in real-world use cases such as professional decision-making (Vossen et al., 2016) or fact-checking (Wang, 2017; Nadeem et al., 2019). Additionally, similar to their single-document counterparts, these datasets are hard to collect and thus are small in size.

The task of cross-document TempRel extraction could be more challenging than the single-document task since a model would possibly need to perform event coreference while performing temporal grounding across documents. This is similar to many tasks nowadays that operate across multiple documents, such as open-domain knowledge extraction and question answering (Chen et al., 2017), which are much more challenging than that on a single document.

In this work, we formulate the task of cross-document TempRel extraction, and construct a dataset from online news data to evaluate TempRels between events that appear both in the same document and across documents. The use of hyperlinks and associated publication dates allow us to scale up and automatically construct a large dataset that can be used for training and evaluation. We run popular neural models as baselines with this set of data on cross-document TempRel extraction. While we report improved performance over simpler baselines using pretrained transformers, there is still a lot of progress to be made on this task. Moreover, we show that the meta data in the form of hyperlinks could be incorporated into the inference stage to improve the extraction of TempRels, by supplementing the original TempRel model with an event linking model. Motivated by open-domain tasks, events could be linked to a set of news articles, which we call *anchoring* events, and they can be added to the temporal graph and enforce additional constraints to help inference. The contributions of this work are as follows:

- We construct a dataset[1] automatically by utilizing hyperlinks and publication dates from news articles to identify events and ground them temporally, which would be scalable. We run neural network baselines on cross-document TempRel extraction using the collected dataset and show that the task is hard even using state-of-the-art pretrained transformer models.

- We use the associated links for training an event linking model, which is used to add additional constraints to the temporal graph by linking events to anchoring documents. We show that this method can boost performance on top of popular baselines.

2 Related Work

Temporal Relation Extraction There have been many attempts on the problem of classifying the temporal relation between two given events. To support temporal relation research, datasets such as TimeBANK (Pustejovsky et al., 2003) have been used as benchmarks for training and evaluating temporal information extraction systems. A number of datasets have been collected in the following years, including augmentations to TimeBANK (Verhagen et al., 2007, 2010; Bethard et al., 2007; UzZaman et al., 2013; Cassidy et al., 2014; Reimers et al., 2016), and datasets with both temporal and other types of relations (Mostafazadeh et al., 2016). These datasets are densely annotated by experts, who identify every event and temporal expression described in text in each document and assign ground truth relations to pairs of entities. They are usually low in inter-annotator agreement (Ning et al., 2018), and are limited in terms of dataset size as the data collection process is quite challenging.

In terms of modeling temporal relations, early methods (Mani et al., 2006; Chambers et al., 2007) studied the use of classical machine learning algorithms with extracted features. Following a series of TempEval workshops (Verhagen et al., 2007, 2010; UzZaman et al., 2013), a number of works on TempRel extraction have been published (Chambers et al., 2014; Leeuwenberg and Moens, 2017; Ning et al., 2017; Meng and Rumshisky, 2018). In recent studies, large neural models were explored and shown to outperform feature-based methods (Ning et al., 2019; Ballesteros et al., 2020; Lin et al., 2019). In our work, we follow this line of study and explore popular neural models, including pretrained transformer models, as baselines.

Timeline Construction More closely related to our work is the task of cross-document timeline construction, which focuses on cross-document event coreference resolution and cross-document temporal relation extraction (Minard et al., 2015). The latter topic, compared to the counterpart task without the cross-document aspect, sees less interest since the original task is already shown to be very challenging. The first to approach this task was Minard et al., who formulated it as an ordering task in which events involving a specific target entity are to be extracted from documents and ordered chronologically. A small dataset with only trial and evaluation data was collected in the challenge. A slightly larger challenge dataset on storyline extraction followed, which extended to a specific set of topics (Caselli and Vossen, 2017). More recently, Reimers et al. proposed a carefully crafted neural decision tree. In our work, we focus not only on entities or a very specific set of topics, but construct timelines in our dataset based on semantic similarity, and scale the dataset up by a magnitude.

The cross-document event coreference resolution task, on the other hand, is an extension from

[1]The data will be released publicly pending review.

the coreference resolution task which includes not only entities and noun phrases but also for event mentions that usually contain verbs (Humphreys et al., 1997; Bagga and Baldwin, 1998; Lee et al., 2012). In our work we do not directly predict event-event links but do so by linking them to anchoring article titles. Similar to (Lee et al., 2017), we use a neural model to classify links and explore using contemporary transformer models.

3 Task Description and Data Collection

Given a set of documents and a set of target events, the cross-document TempRel extraction task requires a model to order the set of events into a timeline. For this task, ideally we would like to focus on a set of documents that is most relevant to a topic, as this would be most useful for real-world applications. We cannot simply aggregate across single-document TempRel datasets by picking any two time-anchored events and ask a model to predict the relation. Furthermore, it would not make much sense to consider the TempRels between every minor event in a densely annotated dataset, as current datasets mostly restrict TempRels to the events that are close, for example in adjacent paragraphs. This could result in too many irrelevant events in the presentation of a timeline, while also complicating the construction of the timeline and harming performance.

Hence in our work we require data that is more sparsely annotated, containing only the major events in each news article while having built-in temporal annotations in order to scale the data up. News articles published by media sources present an interesting resource for our use case. First of all, these news articles usually identify important events in the text by highlighting them, and then hyperlinking them to other news articles that describe those events. Additionally, as information spreads through the internet almost instantly nowadays, news articles are usually written by reporters right after the start of events, and thus the time and dates of news articles could provide us crucial time information to the events themselves and be treated as labels. Moreover, the hyperlinks can be utilized as training signals for linking identified events to related articles that are written about them, as we will describe later in Section 4.2. Given these reasons, we collect a dataset of news articles from online media to train and evaluate our models.

We gather a total of 10,000 articles from CNN[2], dated up to June 2020. Of those 10,000 articles, 7,116 contain hyperlinked text to other news articles. Following previous work on the definition of events, we extract the head verb from each piece of hyperlinked text with NLTK (Loper and Bird, 2002) to represent an event. This gives us a total of 6,648 articles that contain at least one event. We further split the articles chronologically to get 4640/946/1062 of train/dev/test articles. The title and date of the article that is hyperlinked to the text is also extracted for each piece of hyperlinked text. We again follow previous work and focus only on the starting points of each event, as end-points has been shown to be hard to determine even for human annotators (Minard et al., 2015; Ning et al., 2018). The exact date of the hyperlinked article is used as a proxy to the exact event start time, since most news articles nowadays are published and dated on the day of the start of the event. Overall, we have 16,458 events in the 6,648 articles, an average of 2.48 events per article.

An example article from the collected dataset is shown in Figure 1. The hyperlinked text are in blue, with event verbs tagged by NLTK in bold, and the hyperlinked articles are shown on the side. In the example, important events that are relevant to the topic of air pollution during lockdowns, are highlighted and linked to related articles that are also published by CNN. Additionally, the hyperlinked articles are mostly close to the start of the event that the text is referring to, as seen in events “extended” and “began”. This supports the use of the hyperlink dates as a proxy to the highlighted events. This, however, also introduces some error, which can be seen in the event “declared”, for which the linked article does not describe that particular event, but refers to it in its text. We find these kinds of error infrequent, and the dates are generally correct. The publication dates of two given events are further used to determine the TempRel labels from the label set {*before, after, equal*}.

Finally, to construct a set of documents that are relevant for building a timeline, we take each article and retrieve the top 2 articles from the same split with TF-IDF to create a triplet, resulting in a total of 4640/946/1062 triplets for the splits respectively, the same as the number of documents. A triplet of articles is treated as a set, and given a triplet the goal is to create a timeline out of all events that

[2] https://www.cnn.com

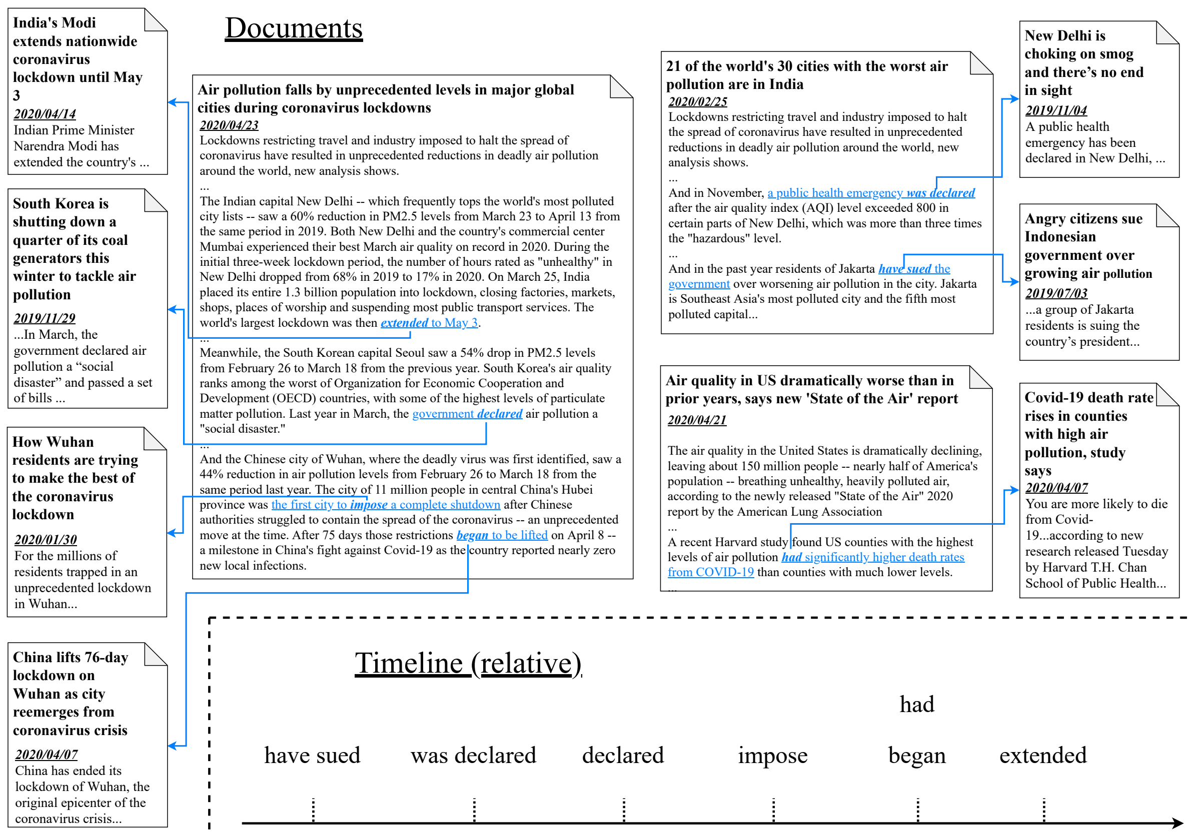

Figure 1: An example triplet from the created dataset. The documents are shown in the middle, and the hyperlinked articles containing titles and publication dates are shown on the sides. The hyperlink text are in blue, with event verbs tagged by NLTK in bold. At test time, given a triplet of documents with events highlighted but without links, the goal is to predict the relative timeline that is shown at the bottom.

	train	dev	test
#Docs (=#Triplets)	4640	946	1062
#Events	11.6k	2.2k	2.6k
#TempRels	1,059k	166k	135k
#Events / Doc	2.5	2.4	2.4
#Events / Triplet	13.6	13.9	13.0
#TempRels / Triplet	228.3	175.5	127.3

Table 1: Dataset statistics from the collected news dataset.

are in these three articles. The dataset statistics are shown in Table 1.

4 Modeling

In this section we describe the proposed method of determining cross-document TempRels with temporal anchoring events. As a refresher, given a set of documents and a set of events, the goal of this task is to order the set of events into a relative timeline. This process is usually done by constructing a temporal graph with each node in the graph representing an event, and predicting the TempRel, represented as edges, between events. We follow most previous studies and separate this process into local (L) relative predictions between two events, followed by an inference (I) stage to enforce temporal constraints. There have been works that explored global methods, however we focus on neural models in our work, which are usually incompatible with those methods due to discreteness of the inference problem.

In Section 4.1 we introduce the local prediction method we use for baseline models, in Section 4.2 we describe the model we use to link events to a set of anchoring events, and finally in Section 4.3 we describe the inference process that incorporates the anchors to construct the final temporal graph output that is globally consistent. An overview of the proposed method is shown in Figure 2.

4.1 Temporal Relation Modeling

As described earlier, in this step we are performing local TempRel predictions given a pair of events.

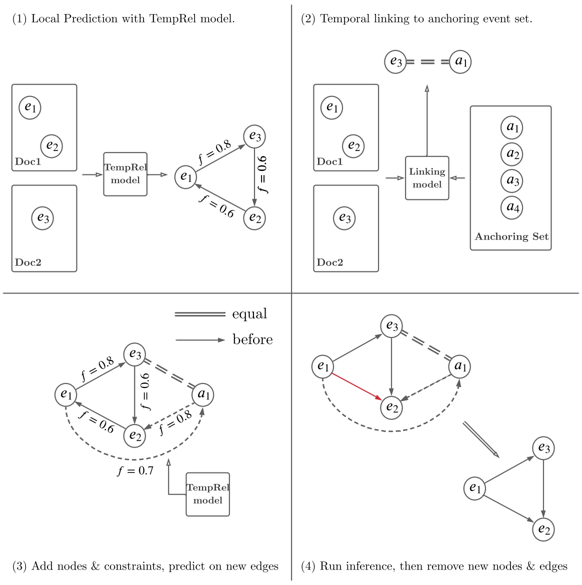

Figure 2: An illustration of the proposed method. (1) Obtain local predictions with the TempRel model. Notice that there may be inconsistencies in the temporal graph, which happens here as there exists a cycle. (2) Using the event linking model, obtain a set of "event - anchor event" links. (3) Add anchor nodes and edges to the temporal graph, while constraining the linked edges to be *equal*. (4) Run integer linear programming (ILP) inference. The extra nodes and edges give extra temporal information for the program to sort out the inconsistencies. They are then discarded at output.

We explore several neural sequence models in this work.

We follow previous work in feeding the context of the events as a sequence into the model, obtaining a representation for the particular pair of events, and finally feeding it into a fully-connected network to generate confidence scores as outputs. Specifically, the contexts for the two events are first concatenated as inputs, separated by separator tokens. The sequence goes through an encoder model into a sequence of hidden state representations. The hidden states corresponding to the tokens of each event are extracted and averaged to get an embedding vector. The two embedding vectors are finally concatenated and fed to the fully-connected layers for prediction.

4.2 Temporal Anchoring with Event Linking

At this stage, we have a model that can predict the TempRel between two given events. Normally, given a set of events, or nodes, and a set of edges to be predicted, the model could be used to predict those edges before proceeding to the inference stage.

However, given the dataset we collected, we have extra metadata we can utilize to potentially improve our predictions. Recall that each event in the original news article is hyperlinked to an article that describes the event. If we have access to a set of such documents, which we refer to as *anchoring events*, and have a model that can detect such links, we could inject extra temporal information that may be useful in the inference step. There are information that we may be able to gain based on these links. For example, suppose we have a linking model that links several events to the same underlying anchoring event, we would be able to enforce at the inference stage these events happen on the same date (*equal*). Even if we do not link multiple events to the same anchoring event, these anchoring events may still be useful when we use them as *extra* events. Note that we would need to make sure the original event and the linked event are *equal* in the graph. With the additional nodes in the graph, we can predict extra edges in the graph, and then run the inference algorithm to take the information into account. We hypothesize that this would make the system more robust, and potentially correct some of the original mistakes the local model makes. Based on the reasons laid out above, we propose to add an event linking step before inference.

In our formulation, the goal of this step is to link an event to some other events which are represented by *articles*. This differs from existing event coreference resolution problems and datasets, for which tagged events need to be partitioned into those that refer to the same underlying event. Specifically, given a tagged event and an article, our goal is to predict whether they refer to the same event. This is comparable to mention-pair models for event coreference problems.

To train such an event linking model, here again we utilize the additional proxy targets in the dataset. A hyperlink leads to the article that describes the event, and thus we choose to utilize that article along with the hyperlinked text as a pair. The *(hyperlinked text, hyperlink article title)* pairs are treated as positive examples, and as we would need negative examples for training, we randomly sample unrelated articles from the training set as negative pairs. For our classifier, again we use an neural sequence model, which takes a context-title pair concatenated as inputs. The hidden states corresponding to the event and title are averaged, separately, and the concatenation of the two mean vectors are passed through a classifier to get the prediction score.

4.3 Inference

Local predictions, for which we predict relations between each pair of events independently, could lead to inconsistencies across multiple pairs. In the view of temporal graphs, the structure should be constrained by transitivity. To enforce the global temporal consistency, we follow previous work by formulating and solving an integer linear programming (ILP) problem (Roth and Yih, 2004; Chambers and Jurafsky, 2008).

In our work, we follow the formulation described in Ning et al. (2017). To integrate anchoring events into the inference process, we add the linked anchoring events as new nodes to the graph, and then produce local predictions between each pair of events. Transitivity constraints are enforced through the optimization problem constraints. Additionally, each linked event should be labeled *equal* to the original event, so we enforce this by adding it as an extra constraint to the program. After adding all constraints to the problem, we solve the ILP with an off-the-shelf solver to obtain temporally-consistent predictions.

Specifically, let $y = \{y_1, \ldots, y_n\} \in \mathcal{Y}^n$ where $\mathcal{Y} = \{before, after, equal\}$ is the label set for TempRels. For the inference optimization problem, let $\mathcal{I}_r(i,j) \in \{0,1\}$ be the indicator function of the relation r between events i and j and $f_r(i,j)$ be the corresponding classifier output score. The ILP problem is then:

$$
\begin{aligned}
\hat{\mathcal{I}} = \arg\max_{\mathcal{I}} & \sum_{i,j\in\mathcal{E}} \sum_{r\in\mathcal{Y}} f_r(i,j)\mathcal{I}_r(i,j) \\
& + \lambda \sum_{i,j\in\mathcal{E}'} \sum_{r\in\mathcal{Y}} f_r(i,j)\mathcal{I}_r(i,j) \\
\text{s.t.} & \sum_r \mathcal{I}_r(i,j) = 1, && (1) \\
& \mathcal{I}_r(i,j) = \mathcal{I}_{\bar{r}}(j,i), && (2) \\
& \mathcal{I}_{r_1}(i,j) + \mathcal{I}_{r_2}(j,k) - \sum_{m=1}^{N} \mathcal{I}_{r_3^m}(i,k) \leq 1, && (3) \\
& \mathcal{I}_{\text{equal}}(i,j) = 1 \text{ when } i,j \text{ are linked}, && (4)
\end{aligned}
$$

for all distinct events i, j, and k, where $\mathcal{E}$ is the set of all original event pairs, $\mathcal{E}'$ is the set of all newly added event pairs due to linking, $\bar{r}$ is the reverse relation of r, and N is the number of possible relations of r_3 when r_1 and r_2 are true, and λ is a weighting factor for the newly added links. Constraints (1) enforces uniqueness, constraints (2) enforces symmetry, constraints (3) enforces transitivity, and constraints (4) enforces simultaneity of the linked events.

After solving for the objective, we can finally drop all edges $\mathcal{E}'$ connecting to the newly-added nodes while keeping the original edges $\mathcal{E}$, and output the edge predictions.

5 Experiments

5.1 Event Linking Model

Before moving on to the main results, we first describe how we trained the event linking model that is used to link to anchoring events, and evaluate how well this linking model is performing.

We use the training process of event linking described earlier on the training set. Given an event and the hyperlink title, we use the pair *(event text, hyperlink article title)* as input to predict a positive target, and sample random titles to get pairs to predict negative titles. When sampling a random title, we set a 30% probability of sampling the title from another event in the article, or otherwise sample from the entire training article pool. We use 10 negative samples in our experiments. The RoBERTa (Liu et al., 2019) base model is used as our linking model. We set the maximum length to 512 tokens, train for 10 epochs with early stopping, a learning rate of 3e-5 using a triangular schedule with warmup of 0.1, and a batch size of 256.

Here we report the event linking performance of the model that we use for the TempRel task. For a given input event, we input *(event, title)* for all titles in our article pool. We choose to evaluate in a ranking setting, by ranking the output scores of the model and selecting the articles with top k scores. The RoBERTa model achieves recall@k of 33.8, 53.5, and 61.8, respectively for $k = 1$, 5, and 10, correctly retrieving 33.8% when greedily selecting the article with the top score.

5.2 Temporal Relation Extraction Setting

We now describe the experimental setting for the cross-document TempRel extraction task. For the local prediction encoder model, we consider the following baselines:

Random & Majority We report performance of a random baseline where the unnormalized output logits are randomly sampled from a uniform distribution. For the majority baseline, the model always chooses the *after*.

LSTM (Hochreiter and Schmidhuber, 1997) We compare against a 4-layer unidirectional LSTM as baseline, with the same number of hidden units in a layer, 768, as a RoBERTa model. It is trained for 5 epochs with early stopping, a learning rate of 3e-5 using a triangular schedule with warmup of 0.2, and a batch size of 16.

RoBERTa (Liu et al., 2019) Here again the RoBERTa base model is used. We set the maximum length to 512 tokens, and use the same training hyperparameters as the LSTM model.

Longformer (Beltagy et al., 2020) We also would like to explore the effects of using longer contexts, since models would need to integrate information across documents and may require longer term dependencies to perform well. Since RoBERTa supports maximum sequence length only up to 512, we experiment with the Longformer model, a variant that combines local and global attention windows, which takes sequences with length up to 4096. We use the Longformer base model, which has the same number of layers and hidden units as RoBERTa base. The training hyperparameters are kept the same.

For baselines on TempRel prediction, all models are run on the testing set by first predicting the confidence scores for all pairs of events that appear in each triplet of articles. Inference step is then run on the output scores to get the final predictions.

We also perform inference by linking to anchoring events with the linking model obtained earlier. For the set of anchoring events, we randomly chose 10% of all article titles that appear in the training split, including titles from the training articles themselves and the article titles of the hyperlinks. We restrict each event to link to at most one article from the anchoring pool, by selecting the one with the highest confidence score in that situation. Once we obtain a set of event-to-anchor-event links, we add those to the inference step as new events, relations and constraints, but remove them when calculating the final evaluation scores. The factor λ is selected by performance on the dev set. We report scores for the baseline models with the addition of the linking step, which are denoted by "w/ linking". To solve the ILP programs, we use the Gurobi solver (Gurobi Optimization, LLC, 2021). We use PyTorch (Paszke et al., 2019) and the Transformers library (Wolf et al., 2020) for our models and experiments.

		Local Pred.		Inference	
		Acc.	F_1	Acc.	F_1
1	Random	33.5	27.1	-	-
2	Majority	54.8	23.6	-	-
3	LSTM	54.5	33.6	53.1	35.5
4	- w/ linking	-	-	53.5	37.9
5	RoBERTa	56.5	43.7	56.6	42.8
6	- w/ linking	-	-	56.9	43.4
7	Longformer	56.6	45.6	56.6	44.4
8	- w/ linking	-	-	**57.1**	**44.6**

Table 2: Results of cross-document temporal relation extraction on the collected news dataset.

The evaluation metrics we use for this task are *accuracy* and *macro F1 score*. In addition to reporting metrics on the final outputs, we also report performance when obtaining predictions directly from the outputs of the local prediction models and skipping the inference step. All experiments were performed over three runs, including the random selection of anchoring events. Reported scores are averaged over those runs.

5.3 Temporal Relation Extraction Results

In Table 2 we show the results of the cross-document temporal relation extraction task. Lines 1 and 2 show the most naive baseline results, giving very low F_1 scores. We are not able to report the performance after running the inference step, since the outputs would violate too many temporal constraints that cannot be resolved efficiently by running ILP.

Comparing lines 3, 5, and 7, we see that LSTM has a big performance gap compared to the other two pretrained transformer models, which is not completely unexpected. There is a 3.5% gap in accuracy and a 7% gap in F_1, with the latter metric usually being harder to improve, suggesting the transformer models are major improvements over the LSTM. Between the two transformer models using different context lengths, both models have almost the same accuracy score, but Longformer outperforms on F_1 by almost 2%. Since the number of parameters are similar, with the two models having the same number of layers and units except the positional embedding size, this suggests that the task possibly requires longer ranged dependencies in order to do better on temporal grounding, which is what we had hypothesized when setting up the task.

Comparing the two columns, the performance of the models based on local prediction scores versus after inference, we see that most models perform equally or worse on accuracy. LSTM benefits on F_1, but the transformers have a roughly 1% decrease in performance. The inference step sorts out the inconsistencies from the local prediction outputs, however, it comes at the sacrifice of performance.

With linking, lines 4, 6, and 8, we obtain quite an improvement on LSTM, with more than 2% increase in F_1. For the transformers, we see a smaller scale but still consistently gives the baseline models performance improvements. RoBERTa gains roughly 0.3% on accuracy and 0.6% on F_1, while Longformer gains 0.5% on accuracy and 0.2% on F_1. The local predictions are the same as their baseline counterparts (and thus the scores are not shown in the table), so the addition of these links suggest we can mitigate some of the performance drop when sorting out the conflicting output confidence scores. We can think of these linked events as a "paraphrase" or augmentation of the original events, and we use these to get extra output scores averaged with the original scores to make the system more robust, potentially correcting more mistakes that the model originally makes.

5.4 Effects of Anchoring Set Size

Since we are using anchoring event sets that we have on hand to aid inference, we would like to know how much data we need to have in order to perform well. Keeping too large of an anchoring set not only requires larger space, it also slows down the entire process as we would need to run linking scores over the entire anchoring set, and that more links would be generated and would also slow down the inference process itself.

In previous experiments, we use a set size of 10% of all titles seen in training, around 900, which is selected by performance on the dev set. Here we run the experiments with the RoBERTa model over set sizes of {1%, 5%, 10%, 20%, 50%}, and the results are shown in Figure 3. When we use 1%, we do not have many links so the performance is roughly the same. Interestingly, when we use 5%

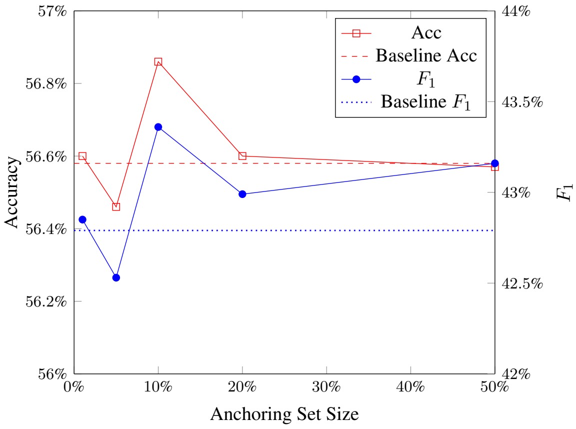

Figure 3: Performance of the RoBERTa model with different anchoring set size (as a percentage of all titles in the training set). Accuracy is on the left and F_1 is on the right. The performance for the baseline model without linking are shown as constants in the plot.

the model performances actually worsen, which may indicate that the set doesn't cover enough good anchors and the linking model links to those that hurt performance. With larger amounts of links, we would get more "nice" anchors but also introduce more noise, and at around the set size of 10% we get the best tradeoff. Finally, we note that 50% anchoring set size gives roughly the same accuracy but improves F_1 performance.

6 Conclusion and Future Work

In this work we focus on extracting timelines across documents. We construct a dataset automatically by utilizing hyperlinks and publication dates from online news articles to identify events and time anchors, making it scalable. We target the temporal relation extraction task, and propose a method using the associated links for training an event linking model, which is used to aid the inference procedure. We run neural model baselines and show that our proposed method can boost performance on top of them.

For future work, we would like to focus not only on event-event relations but also consider event-time connections. This would allow us to anchor events to absolute time or dates and be more applicable to real world tasks. This could also be extended to a complete system that detects events and performs event coreference for end-to-end operation. We also plan to further investigate the transfer of our collected data to other similar tasks or datasets, especially those that have little to none training data.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Miguel Ballesteros, Rishita Anubhai, Shuai Wang, Nima Pourdamghani, Yogarshi Vyas, Jie Ma, Parminder Bhatia, Kathleen McKeown, and Yaser Al-Onaizan. 2020. Severing the edge between before and after: Neural architectures for temporal ordering of events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5412–5417, Online. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Steven Bethard, James H. Martin, and Sara Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *International Conference on Semantic Computing (ICSC 2007)*, pages 11–18.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. 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.

Nathanael Chambers and Daniel Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706, Honolulu, Hawaii. Association for Computational Linguistics.

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

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Gurobi Optimization, LLC. 2021. Gurobi Optimizer Reference Manual.

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

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea. Association for Computational Linguistics.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1150–1158, Valencia, Spain. Association for Computational Linguistics.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246, Brussels, Belgium. Association for Computational Linguistics.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA. Association for Computational Linguistics.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760, Sydney, Australia. Association for Computational Linguistics.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 527–536, Melbourne, Australia. Association for Computational Linguistics.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Rubén Urizar. 2015. SemEval-2015 task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, Denver, Colorado. Association for Computational Linguistics.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61, San Diego, California. Association for Computational Linguistics.

Moin Nadeem, Wei Fang, Brian Xu, Mitra Mohtarami, and James Glass. 2019. FAKTA: An automatic end-to-end fact checking system. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 78–83, Minneapolis, Minnesota. Association for Computational Linguistics.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037, Copenhagen, Denmark. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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. Lancaster, UK.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the TimeBank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2195–2204, Berlin, Germany. Association for Computational Linguistics.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2018. Event time extraction with a decision tree of neural classifiers. *Transactions of the Association for Computational Linguistics*, 6:77–89.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the*

5th International Workshop on Semantic Evaluation, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

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

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

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

Piek Vossen, Rodrigo Agerri, Itziar Aldabe, Agata Cybulska, Marieke van Erp, Antske Fokkens, Egoitz Laparra, Anne-Lyse Minard, Alessio Palmero Aprosio, German Rigau, Marco Rospocher, and Roxane Segers. 2016. Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. *Special Issue Knowledge-Based Systems, Elsevier*.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.