Severing the Edge Between Before and After: Neural Architectures for Temporal Ordering of Events

Miguel Ballesteros **Rishita Anubhai** **Shuai Wang**
Nima Pourdamghani **Yogarshi Vyas** **Jie Ma**
Parminder Bhatia **Kathleen McKeown*** **Yaser Al-Onaizan**
Amazon AI
{ballemig,ranubhai,wshui,nimpourd,yogarshi,jieman,parmib,mckeownk,onaizan}@amazon.com

Abstract

In this paper, we propose a neural architecture and a set of training methods for ordering events by predicting temporal relations. Our proposed models receive a pair of events within a span of text as input and they identify temporal relations (*Before*, *After*, *Equal*, *Vague*) between them. Given that a key challenge with this task is the scarcity of annotated data, our models rely on either pretrained representations (i.e. RoBERTa, BERT or ELMo), transfer and multi-task learning (by leveraging complementary datasets), and self-training techniques. Experiments on the MATRES dataset of English documents establish a new state-of-the-art on this task.

1 Introduction

The task of temporal ordering of events involves predicting the temporal relation between a pair of input events in a span of text (Figure 1). This task is challenging as it requires deep understanding of temporal aspects of language and the amount of annotated data is scarce.

> Albright **(e1, came)** to the State Department to **(e2, offer)** condolences.

Figure 1: Example from the MATRES dataset. The relation between **(e1, came)** and **(e2, offer)** is *Before*. Note that for the same span there may be other relation pairs.

The MATRES dataset (Ning et al., 2018) has become a de facto standard for temporal ordering of events.[1] It contains 13,577 pairs of events annotated with a temporal relation (*Before*, *After*, *Equal*, *Vague*) within 256 **English** documents (and 20 more for evaluation) from TimeBank[2] (Pustejovsky et al., 2003), AQUAINT[3] (Graff, 2002) and Platinum (UzZaman et al., 2013).

In this paper, we present a set of neural architectures for temporal ordering of events. Our main model (Section 2) is similar to the temporal ordering models designed by Goyal and Durrett (2019), Liu et al. (2019a) and Ning et al. (2019).

Our main contributions are: (1) a neural architecture that can flexibly adapt different encoders and pretrained word embedders to form a contextual pairwise argument representation. Given the scarcity of training data, (2) we explore the application of an existing framework for Scheduled Multitask-Learning (henceforth SMTL) (Kiperwasser and Ballesteros, 2018) by leveraging complementary (temporal and non temporal) information to our models; this imitates pretraining and finetuning. This consumes timex information in a different way than Goyal and Durrett (2019). (3) A self-training method that incorporates the predictions of our model and learns from them; we test it jointly with the SMTL method.

Our baseline model that uses RoBERTa (Liu et al., 2019b) already surpasses the state-of-the-art by 2 F1 points. Applying SMTL techniques affords further improvements with at least one of our auxiliary tasks. Finally, our self-training experiments, explored via SMTL as well, establishes yet another state-of-the-art yielding a total improvement of almost 4 F1 points over results from past work.

2 Our Baseline Model

Our pairwise temporal ordering model receives as input a sequence $X_{[0,n)}$ of n tokens

*Kathleen McKeown is an Amazon Scholar and a Professor at Columbia University.

[1] https://github.com/qiangning/MATRES

[2] https://catalog.ldc.upenn.edu/LDC2006T08

[3] https://catalog.ldc.upenn.edu/LDC2002T31

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5412–5417,
November 16–20, 2020.

(or subword units for BERT-like models) i.e. $\{x_0, x_1, ..., x_{n-1}\}$, representing the input text. A subsequence $span_i$ is defined by $start_i$, $end_i \in [0, n)$. Subsequences $span_1$ and $span_2$ represent the input pair of argument events $e1$ and $e2$ respectively. The goal of the model is to predict the temporal relation between $e1$ and $e2$.

First, the model embeds the input sequence into a vector representation using either static wang2vec representations (Ling et al., 2015), or contextualized representations from ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), or RoBERTa (Liu et al., 2019b). These embedded sequences are then optionally encoded with either LSTMs or Transformers. When BERT or RoBERTa is used to embed the input, we do not use any sequence encoders. The final sequence representation $H_{[0,n)}$ comprises of individual token representations i.e. $\{h_0, h_1, ..., h_{n-1}\}$.

While the goal is to predict the temporal relation between $span_1$ and $span_2$, the context around these two spans also has linguistic signals that connect the two arguments. To use this contextual information, we extract five constituent subsequences from the sequence representation $H_{[0,n)}$: (1) S_1, the subsequence before $span_1$ i.e., $H_{[0,start_1)}$, (2) S_2, the subsequence corresponding to $span_1$ i.e., $H_{[start_1,end_1)}$, (3) S_3, the subsequence between $span_1$ and $span_2$ i.e, $H_{[end_1,start_2)}$, (4) S_4, the subsequence corresponding to $span_2$ i.e., $H_{[start_2,end_2)}$ and (5) S_5, the subsequence after $span_2$, i.e. $H_{[end_2,n)}$. Each of these subsequences S_i has a variable number of tokens which are pooled to yield a fixed size representation s_i:

$$s_i = pool(S_i) \;\; \forall i \in \{1, ..., 5\} \quad (1)$$

where *pool* is the result of concatenating the output of an attention mechanism (we use the *word attention* pooling method (Yang et al., 2016) for all tokens in a given span) and mean pooling.

The final contextual pair representation c is formed by concatenating[4] the five span representations s_i with a sequence representation r. For models with BERT and RoBERTa, r is the CLS and <s> token representation respectively while for other models $r = pool(H_{[0,n)})$.

$$c = s_1 \odot s_2 \odot s_3 \odot s_4 \odot s_5 \odot r \quad (2)$$

This final contextual pair representation c is then projected with a fully connected layer followed by a softmax function to get a distribution over the output classes. The entire model is trained end-to-end using the cross entropy loss.

[4] $\odot$ is used to denote concatenation

3 Multi-task Learning

While the model described in the previous section can be directly trained using labeled training data, the amount of annotated training data for this task (in the MATRES dataset) is limited. We enrich our model with useful information from other complementary tasks via SMTL.

3.1 Method

We adapt the framework of Kiperwasser and Ballesteros (2018), where three **schedulers** are used. They follow either a constant, sigmoid or exponential curve $p(t)$, where $p(t)$ is the probability of picking a batch from the main task, t is the amount of data visited so far throughout the training process and α is a hyperparameter. The constant scheduler splits the batches randomly; at any time step, the model will be trained with sentences belonging to either the main task or the auxiliary task ($p^{const}(t) = \alpha$, $0 \leq \alpha \leq 1$) . The sigmoid scheduler allows the model to visit batches from both the auxiliary task and the main task at the beginning while the latest updates are always with batches consisting of batches from the main task ($p^{sig}(t) = \frac{1}{1+e^{-\alpha t}}$). The exponential scheduler starts by visiting only the batches from the auxiliary task while the latest updates are always from the main task ($p^{exp}(t) = 1 - e^{-\alpha t}$).

Following past work, we prepend a trained task vector to the encoder to help the model to differentiate between the main and the auxiliary tasks (Ammar et al., 2016; Johnson et al., 2017; Kiperwasser and Ballesteros, 2018, *inter alia*).

3.2 Auxiliary Datasets

We use three different **auxiliary datasets** in our SMTL setup. The first two have a different taxonomy and label set than MATRES, but have gold annotations. The last one is a silver dataset with predicted labels and same taxonomy as MATRES.

Our first dataset is the **ACE relation extraction task**.[5] We hypothesize that this task can add knowledge of different domains and of the concept of linking two spans in text given a taxonomy

[5] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-relations-guidelines-v6.2.pdf

Robert F. Angelo, who **(event, left)** Phoenix at **(timex, the beginning of October)**.

Figure 2: Example of an event-timex annotation from the Timex annotations. The relation between **(event, left)** and **(timex, the beginning of October)** is *Is_included*.

of relations. While this is not directly related to events and our farthest task in terms of similarity, the pairwise span classification is the reason we include this.

We also use a closer and complementary temporal annotation dataset, i.e. the **Timebank and Aquaint annotations involving timex relations** (timex-event, event-timex, timex-timex) (Ning et al., 2018; Goyal and Durrett, 2019).[6] We expect the model to greatly benefit from being exposed to the timex relations in an MTL framework by learning about temporality in general and by adding specificity of the event-event temporal relations from the MATRES annotations. Figure 2 shows an example of the data annotated with an event-timex relation.

We use self-training (Scudder, 1965) to generate our third dataset: **a silver dataset**. This requires an unlabeled text, a tagger to extract events from this text, and a classifier to predict temporal relations for pairs of extracted events. As our unlabeled text, we use 6,000 random documents from the CNN / Daily Mail dataset which is a collection of news articles collected between 2007 and 2015 (Hermann et al., 2015). We picked 85K segments of text within these documents that contain between 10 and 40 tokens after tokenization. We train a RoBERTa-based named entity tagger and use it to tag events in these segments.[7] This results in about 65K events. We consider all 285K pairs of events that lie within a segment as candidates for temporal ordering. Finally, we use our baseline RoBERTa temporal model to classify the temporal relation between these candidate pairs and use the top $\frac{2}{3}^{rd}$ most confident classifications based on softmax scores to get about 190K instances of silver relations.

[6] http://www.timeml.org/publications/timeMLdocs/timeml_1.2.1.html.

[7] The tagger is simply a dense layer on top of RoBERTa representation. We evaluate the tagger by using it to tag events in the MATRES validation set. The tagger reaches a F1 score of 89.5 on the MATRES development set.

4 Experiments and Results

The MATRES dataset is our primary dataset for training and validation. As in previous work, we use TimeBank and AQUAINT (256 articles) for training, 25 articles of which are selected at random for validation and Platinum (20 articles) as a held-out test set (Ning et al., 2018; Goyal and Durrett, 2019; Ning et al., 2019). Articles from TimeBank and AQUAINT at full length are about 400 tokens long on average. We believe that the document in its entirety is not required to infer the temporality between a given pair of events. Moreover, BERT style models are also often pre-trained for shorter inputs than this. For these reasons, we truncate our input text to a window of sentences[8] starting with one sentence before the first event argument up to and including one sentence after the second event argument.

We use one set of hyperparameters for all LSTM models and another set for all the Transformer models (both with and without ELMo embedder).[9] BERT and/or RoBERTa are loaded as a replacement of the Transformer parameters and they are therefore used both as embedders and encoders. We run our SMTL and self training experiments with our best baseline model on the development data: the RoBERTa model.

For the SMTL experiments, we explore the α hyperparameter, and we pick the one that produces the highest scores in our development data.

Finally, we picked our best SMTL model on the development data (see Table, this is the constant scheduler with silver data) parameters and continue training on the gold data only; we reduce the learning rate to 10^{-6}. This is because the model trained in the first step is already in a good state and we want to avoid distorting it with aggressive updates.

We compare our results (Table 1) with other top performing systems. First, we observe that among models without contextualized representations, the LSTM encoder is 2.5 F1 points better than the Transformer encoder. We observe that replacing static word representations with ELMo representations leads to significantly worse F1 with

[8] We use spacy (Honnibal and Montani, 2017) for sentence segmentation of the articles

[9] LSTM models use 2 hidden layers with 256 hidden units each, and a batch size of 64. Transformer models use 1 hidden layer with 128 hidden units, and a batch size of 24. All models are trained using Adam (Kingma and Ba, 2014) with a learning rate of 10^{-5} on an NVIDIA V100 16GB GPU.

Experiment	Acc	F1
LSTM	64.4 ± 0.36	69.1 ± 0.39
+ Elmo	60.0 ± 2.89	64.8 ± 3.00
Transformer	61.9 ± 0.93	66.4 ± 0.99
+ Elmo	62.2 ± 1.3	66.9 ± 1.35
BERT base	71.5 ± 0.63	77.2 ± 0.74
RoBERTa base	73.5 ± 1.03	78.9 ± 1.16
+ SMTL (ACE) constant (0.6)	72.5 ± 0.69	78.5 ± 0.84
+ SMTL (ACE) exponent (0.5)	71.5 ± 1.81	77.4 ± 1.19
+ SMTL (ACE) sigmoid (0.5)	70.0 ± 1.81	76.4 ± 0.89
+ SMTL (Timex) constant (0.9)	73.4 ± 1.81	79.3 ± 0.64
+ SMTL (Timex) exponent (0.7)	73.7 ± 0.74	79.4 ± -0.46
+ SMTL (Timex) sigmoid (0.8)	74.2 ± 0.74	79.8 ± 0.70
+ SMTL (silver data) constant (0.05)	73.8 ± 0.74	80.3 ± 0.51
+ SMTL (silver data) sigmoid (0.2)	74.0 ± 0.73	80.1 ± 0.72
+ SMTL (silver data) exponent (0.1)	73.9 ± 0.64	79.6 ± 0.52
Self-training: fine-tune on gold	**75.5** ± 0.39	**81.6** ± 0.26
Ning et al. (2018)	61.6	66.6
Goyal and Durrett (2019)[10]	68.6	74.2
Ning et al. (2019)	71.7	76.7

Table 1: Results, including comparison with the best systems on the MATRES test set (Platinum). Results highlighted in bold are the best in each metric. We report average (and standard deviation) of accuracy and F1 over 5 runs with different random seeds. Given that it does not carry temporal information, we treat the relation VAGUE as a *no relation* for the F1 results as in Ning et al. (2019). For the SMTL experiments, the selected α value is shown between parentheses.

the LSTM encoder, but marginally improves upon the F1 of the Transformer encoder. We attribute this difference to the non-complementary nature of LSTM and ELMo representations, as ELMo is also LSTM-based, and thus the ELMo+LSTM combination might need more training data in order to extract meaningful signals.

Importantly, however, our base model that uses pretrained RoBERTa surpasses the previous state-of-the-art (Ning et al., 2019) which uses BERT. Our BERT models yield very similar results to them. The main differences are that they do not finetune BERT along with the updates to the model, while we do and also, we model the context around the argument spans explicitly as part of S_1, S_3 and S_5. The reason why RoBERTa is better than BERT in this case is likely due to the fact that it has been trained longer, over more data, and over longer sequences. This matters because our temporal ordering model usually takes into account a long span in which both events occur.

The SMTL experiments show that the auxiliary task with timex annotations provides non-negligible improvements of almost 1 F1 point on top of our RoBERTa model. Learning from the timex annotations makes our model more aware of time relations and thus, better at ordering events in time. The sigmoid and exponent schedulers perform better than the constant scheduler, suggesting that the model needs to first learn about temporality, and then learn to be more specialized on predicting temporal ordering relations later. We believe this timex multi-tasking setup to be an implicit yet effective way to teach our model about timexes in general without timex embeddings used in (Goyal and Durrett, 2019). When we use the ACE relation extraction dataset as an auxiliary task, none of the schedulers produce improvements while the sigmoid and exponent scheduler fare significantly worse. This result suggests that if the tasks differ too much, SMTL might not be a helpful strategy.

The self-training experiments (including SMTL with silver data) show that the silver data helps to reach better performance with constant being the best scheduler. Furthermore, fine-tuning of the best model (according to development set score, which in this case it is the same as test set score) on the gold data gives us another boost in performance establishing a new state of the art in the task that is 2.7 F1 points better than our RoBERTa baseline, and almost 4 points better than the previous published results.

5 Conclusions and Future Work

This paper presents neural architectures for ordering events in time. It establishes a new state-of-the-art on the task through pretraining, leveraging complementary tasks through SMTL and self-training techniques.

For the future, instead of using the RoBERTa baseline model for the self-training experiments, we could run several iterations by retraining on the data produced by our best self-trained model(s); this could be a good avenue for further improvements. In addition we plan to extend our work by moving to other languages beyond English (we currently have not tried this due to lack of data) using cross-lingual models, (Subburathinam et al., 2019), applying other architectures like CNNs (Nguyen and Grishman, 2015), incorporating tree structure in our models (Miwa and Bansal, 2016) and/or by handling jointly performing event recognition and temporal ordering (Li and Ji, 2014; Katiyar and Cardie, 2017).

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

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.

Tanya Goyal and Greg Durrett. 2019. Embedding time expressions for deep temporal ordering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4400–4406, Florence, Italy. Association for Computational Linguistics.

David Graff. 2002. *The aquaint corpus of English news text*. Linguistic Data Consortium.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of Word2Vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado. Association for Computational Linguistics.

Sijia Liu, Liwei Wang, Vipin Chaudhary, and Hongfang Liu. 2019a. Attention neural model for temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 134–139, 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. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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.

H Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.