Exploring Contextualized Neural Language Models for Temporal Dependency Parsing

Hayley Ross†[1], Jonathon Cai[2], and Bonan Min[2]

[1]Harvard University
hayleyross@g.harvard.edu
[2]Raytheon BBN Technologies
{jonathon.cai, bonan.min}@raytheon.com

Abstract

Extracting temporal relations between events and time expressions has many applications such as constructing event timelines and time-related question answering. It is a challenging problem which requires syntactic and semantic information at sentence or discourse levels, which may be captured by deep contextualized language models (LMs) such as BERT (Devlin et al., 2019). In this paper, we develop several variants of BERT-based temporal dependency parser, and show that BERT significantly improves temporal dependency parsing (Zhang and Xue, 2018a). We also present a detailed analysis on why deep contextualized neural LMs help and where they may fall short. Source code and resources are made available at https://github.com/bnmin/tdp_ranking.

1 Introduction

Temporal relation extraction has many applications such as constructing event timelines for news articles or narratives as well as time-related question answering. Recently, Zhang and Xue (2018b) presented Temporal Dependency Parsing (TDP), which organizes time expressions and events in a document to form a Temporal Dependency Tree (TDT). Given a previous step which detects time expressions and events, TDP extracts the temporal structure between them. Consider this example:

Example 1: *Kuchma and Yeltsin signed a cooperation plan on February 27, 1998. Russia and Ukraine share similar cultures, and Ukraine was ruled from Moscow for centuries. Yeltsin and Kuchma called for the ratification of the treaty, saying it would create a "strong legal foundation".*

Figure 1 shows the corresponding TDT. Compared to previous pairwise approaches for temporal relation extraction based on TimeML (Pustejovsky et al., 2003a) which require $\binom{n}{2}$ pairs of temporal relations to be annotated, TDT significantly reduces the annotation complexity while still preserving the essential temporal structure between events and temporal relations. TDP is still challenging because it requires syntactic and semantic information at sentence and discourse levels.

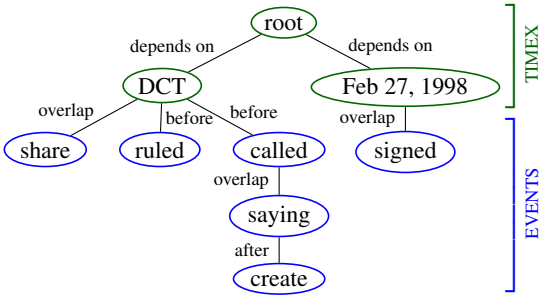

Figure 1: Temporal Dependency Tree of Example 1. DCT is Document Creation Time (March 1, 1998)

Recently, deep language models such as BERT (Devlin et al., 2019) have been shown to be successful at many NLP tasks, because (1) they provide contextualized word embeddings that are pre-trained with very large corpora, and (2) BERT in particular is shown to capture syntactic and semantic information (Tenney et al., 2019, Clark et al., 2019), which may include but is not limited to tense and temporal connectives. Such information is relevant for TDP.

In this paper, we present BERT-based TDP models, and empirical evidence demonstrating that BERT significantly improves TDP. We summarize the contributions of this paper as follows:

- We develop temporal dependency parsers that incorporate BERT, from a straightforward usage of pre-trained BERT word embeddings, to using BERT's multi-layer multi-head self-attention architecture as an encoder trained within an end-to-end system.

A previous version is available at https://arxiv.org/abs/2004.14577.

† Work done during an internship at BBN.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8548–8553,
November 16–20, 2020.

- We present experiments showing significant advantages of the BERT-based TDP models. Experiments show that BERT improves TDP performance in all models, with the best model achieving a 13 absolute F1 point improvement over our re-implementation of the neural model in (Zhang and Xue, 2018a)[1].
- We lay out a detailed analysis on BERT's strengths and limitations for this task.

We present technical details, experiments, and analysis in the rest of this paper.

2 Related Work

Much previous work has been devoted to classification and annotation of relations between events and time expressions, notably TimeML (Pustejovsky et al., 2003a) and TimeBank (Pustejovsky et al., 2003b), as well as many extensions of it (see Derczynski, 2017 for an overview). TimeML annotates all explicit relations in the text; at the extreme, TimeBank-Dense (Cassidy et al., 2014) annotates all $\binom{n}{2}$ pairs of relations. Pair-wise annotation has three problems: $O(n^2)$ complexity; the possibility of inconsistent predictions such as A before B, B before C, C before A; and forcing annotation of relations left unclear by the document.

While extracting time expressions and events is well handled (e.g. Strötgen and Gertz, 2010, Lee et al., 2014), relating them is still a challenging task. Previous research on extracting these relations (e.g. Bethard et al., 2017, Ning et al., 2017, Lin et al., 2019) almost always uses pair-wise TimeML-annotated data which has rich annotation but also inherits the above three complexity and consistency issues. To address these issues, Zhang and Xue (2018b) present a tree structure of relations between time expressions and events (TDT), along with a BiLSTM model (Zhang and Xue, 2018a) for parsing text into TDT and a crowd-sourced corpus (Zhang and Xue, 2019).

Organizing time expressions and events into a tree has a number of advantages over traditional pair-wise temporal annotation. It reduces the annotation complexity to $O(n)$ and avoids cyclic inconsistencies both in the annotation and the model output. Despite the apparent reduction in labeled edges, many additional edge labels can be deduced from the tree: in Figure 1, we can deduce e.g. that *ruled* is before *share* because *ruled* is before DCT but *share* overlaps DCT. A final advantage of TDTs is that they allow underspecification where the source document does not explicitly specify an order, such as the relation between *signed* and *called* (likely to be *overlap*, but it is not certain). Zhang and Xue (2019) is currently the only English-language TDP corpus, comprising 196 newswire articles.

In addition, this paper capitalizes on the now well-documented recent advances provided by BERT (Devlin et al., 2019). Besides offering richer contextual information, BERT in particular is shown to capture syntactic and semantic properties (Tenney et al., 2019, Clark et al., 2019) relevant to TDP, which we show yield improvements over Zhang and Xue's original model.

3 BERT-based Neural Models for Temporal Dependency Parsing

Following Zhang and Xue (2018a), we transformed temporal dependency parsing (TDP) to a ranking problem: given a child mention (event or time expression) x_i extracted by a previous system, the problem is to select the most appropriate parent mention from among the root node, DCT or an event or time expression from the window $x_{i-k}, \ldots, x_i, \ldots, x_{i+m}$ [2] around x_i, along with the relation label (*before*, *after*, *overlap*, *depends on*). That is, for each x_j in the window, the model judges the child-parent candidate pair $\langle x_i, x_j \rangle$. A Temporal Dependency Tree (TDT) is assembled with an incremental algorithm which selects, for each event and time expression in sequence in the document, the highest-ranked prediction ⟨parent, relation type⟩. The tree structure is enforced by selecting the highest probability parent which does not introduce a cycle[3].

We developed three models that share a similar overall architecture (Figure 2): the model takes a pair of mentions (child and potential parent) as input and passes each pair through an encoder which embeds the nodes and surrounding context into a dense representation. All models use the same window approach described above to source parent candidates. Following Zhang and Xue (2018a), linguistic features are concatenated onto the dense representation, which is then passed to a feed-forward

[1] We were unable to replicate the F1-score reported for this corpus in Zhang and Xue (2019). The improvement over the reported, state-of-the-art result is 8 absolute F1 points.

[2] We set $k = 10, m = 3$ in all experiments.

[3] In practice, this step to avoid cyclic edges is rare: it is required for less than 4% of the predicted edges.

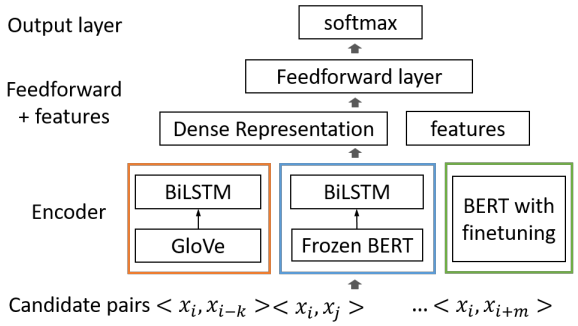

Figure 2: Model architecture for TDP with three different encoders (orange, blue, green boxes). Shown with the ⟨parent, child⟩ input pairs for a given child (event or time expression) x_i. For simplicity, we did not show $\langle x_i, root \rangle$ and $\langle x_i, DCT \rangle$, which are included as candidate pairs for all x_i.

layer and a softmax function to generate scores for each relation label for each pair.

We developed three types of encoder:

- BILSTM and BILSTM-GLOVE feed the document's word embeddings to a BiLSTM to encode the pair as well as the surrounding context. The word embeddings can be either randomly initialized (identical to Zhang and Xue, 2018a) (in BILSTM), or pre-trained from a large corpus – we used GloVe (Pennington et al., 2014) (in BILSTM-GLOVE).
- BILSTM-BERT replaces the word embeddings with frozen (pre-trained) BERT contextualized word embeddings. We used the BERT-base uncased model[4], which has been trained on English Wikipedia and the BookCorpus.
- BERT-FT: BERT's multi-layer multi-head self-attention architecture (with pre-trained weights) is used directly to encode the pairs. Its weights are fine-tuned in the end-to-end TDP training process.

All models use the same loss function and scoring as in Zhang and Xue (2018a). We present more details about the two BERT-based models below.

3.1 Model BILSTM-BERT

The first model adjusts the model architecture from Zhang and Xue (2018a) to replace its word embeddings with frozen BERT embeddings. That is, word embeddings are computed via BERT for every sentence in the document; then, these word embeddings are processed as in the original model. More details about the BiLSTM model can be found in Zhang and Xue (2018a).

[4] https://github.com/google-research/bert

3.2 Model BERT-FT

This model takes advantage of BERT's multi-layer multi-head self-attention architecture (Vaswani et al., 2017) to learn feature representations for classification. The embedding of the first token `[CLS]` is interpreted as a classification output and fine-tuned.

To represent a child-parent pair with context, BERT-FT constructs a pseudo-sentence for the (potential) parent node and a pseudo-sentence for the child node. The pair of pseudo-sentences are concatenated and separated by the `[SEP]` token, and then fed into the BERT model. Each pseudo-sentence is formed of the word(s) of the node, the node's label (TIMEX or EVENT), a separator token ':' and the sentence containing the node, as shown in Table 1.

word(s)	label	sep	sentence
February 27, 1998	TIMEX	:	*Kuchma and Yeltsin signed a cooperation plan on February 27 1998.*
called	EVENT	:	*Yeltsin and Kuchma called for the ratification . . .*

Table 1: A pair of pseudo-sentences in BERT-FT, for potential parent *February 27, 1998* and child *called* in Example 1 (The correct parent here is DCT).

4 Experiments

We use the training, development and test datasets from Zhang and Xue (2019) for all experiments (182 train / 5 development / 9 test documents, total 2084 sentences). The documents in the datasets are already annotated with events and temporal expressions. This allows us to focus on evaluating the task of constructing temporal dependency trees.

We evaluated four configurations of the encoders above. Firstly BILSTM (RE-IMPLEMENTED) re-implements Zhang and Xue (2018a)'s model[5] in TensorFlow (Abadi et al., 2016) for fair comparison. Replacing its randomly-initialized embeddings with GloVe (Pennington et al., 2014) yields BILSTM-GLOVE. We also test the models BILSTM-BERT and BERT-FT as described in Section 3.

We used Adam (Kingma and Ba, 2014) as the optimizer and performed coarse-to-fine grid search for key parameters such as learning rate and number of epochs using the dev set [6]. We observed

[5] Originally implemented in DyNet (Neubig et al., 2017).

[6] We tried all parameter configurations with learning rates in $\{0.001, 0.0001, 0.0005, 0.00025\}$ and numbers of epochs in $\{50, 75, 100\}$, and perform 5 runs for each configuration. We observed a mean F1 of 0.58 with variance=0.002 across

that when fine-tuning BERT in the BERT-FT model, a lower learning rate (0.0001) paired with more epochs (75) achieves top performance, compared to using learning rate 0.001 with 50 epochs for the BiLSTM models. We used NVIDIA Tesla P100 GPUs for training the models. On a single GPU, one epoch takes 7.5 minutes for the BERT-FT model and 0.8 minutes for the BILSTM-BERT model.

	F1 score	
Model	dev	test
Rule-based baseline (Zhang and Xue, 2019)	0.15	0.18
BiLSTM (Zhang and Xue, 2019)	0.53	0.60
BILSTM (re-impl., Zhang and Xue, 2019)	0.45	0.55
BILSTM-GLOVE	0.50	0.58
BILSTM-BERT	0.54	0.61
BERT-FT	0.59	**0.68**

Table 2: Performance of the models.

Table 2 summarizes the F1 scores[7] of our models. Results are averaged over 5 runs. We also include the rule-based baseline and the performance reported in Zhang and Xue (2019), which applies the model of Zhang and Xue (2018a) to the 2019 corpus, as a baseline[8].

BILSTM-BERT outperforms the re-implemented BILSTM model by 6 points and BILSTM-GLOVE by 3 points in F1-score, respectively. This indicates that the frozen, pre-trained BERT embeddings improve temporal relation extraction compared to either kind of non-contextualized embedding. Fine-tuning the BERT-based encoder (BERT-FT) resulted in an absolute improvement of as much as 13 absolute F1 points over the BiLSTM re-implementation, and 8 F1 points over the reported results in Zhang and Xue (2019). This demonstrates that contextualized word embeddings and the BERT architecture, pre-trained with large corpora and fine-tuned for this task, can significantly improve TDP.

We also calculated the models' accuracies on time expressions or events subdivided by their type of parent: DCT, a time expression other than DCT, or another event. Difficult categories are children of DCT and children of events. We see that the main difference between BILSTM and BILSTM-BERT is its performance on children of DCT: with BERT, it scores 0.48 instead of 0.38. Conversely BERT-FT sees improvements across the board over BILSTM, with a 0.21 increase on children of DCT, a 0.14 increase for children of other time expressions, and a 0.11 increase for children of events.

5 Analysis

Why BERT helps: A detailed manual comparison of the dependency trees produced by the different models for articles in the test set shows BERT's advantages for TDP. The following phenomena are attested by many sentences in many documents and correspond to known properties of BERT.

Firstly, unlike BILSTM, BERT-FT is able to properly relate time expressions occurring syntactically after the event, such as *Kuchma and Yeltsin signed a cooperation plan on February 27, 1998* in Example 1. (BILSTM falsely relates *signed* to the "previous" time expression DCT). This shows BERT's ability to "look forward" with its self-attention, attending to parents appearing after the child.

Secondly, BERT-FT is able to capture verb tense and use it to determine the correct relation for both DCT and chains of events. For example, it knows that present tense (*share similar cultures*) overlaps DCT, while past perfect events (*was ruled from Moscow*) happen either before DCT or before the event adjacent (salient) to them.

Thirdly, BERT-FT captures syntactic constructions with implicit temporal relations such as reported speech and gerunds (e.g. in Example 1, *Yeltsin and Kuchma called for the ratification [. . .], saying it would create . . .*, it identifies that *called* and *saying* overlap and *create* is after *saying*).

Similarly, BERT's ability to handle syntactic properties (Tenney et al., 2019, Clark et al., 2019) such as embedded clauses may allow it to detect the direction of connectives such as *since*. While all models may identify the matrix clause verb as the correct parent, BERT-FT is much more likely to choose the correct label. (BILSTM almost always chooses 'before' for DCT or 'after' for children of events, ignoring the connective.)

Lastly, both BERT-FT and BILSTM-BERT are much better than the BILSTM at identifying context changes (new "sections") and linking these events to DCT rather than to a time expression in the previous sections (evidenced by the scores on children of DCT). Because BERT's word embeddings use the sentence as context, the models using BERT may be able to "compare" the sentences and judge that they are unrelated despite being adjacent.

all configurations for all models.

[7]Following (Zhang and Xue, 2019), F1 scores are reported. For a document with n nodes, the TDP task constructs a tree of $n + 1$ edges, so F1 is essentially the same as the accuracy.

[8]We were unable to replicate the F1-score reported in Zhang and Xue (2019) despite using similar hyperparameters. Therefore, we include performances for our re-implementation and the reported score in Zhang and Xue (2019) in Table 2.

Equivalent TDP trees: In cases where BERT-FT is incorrect, it sometimes produces an equivalent or very similar tree (since relations such as *overlap* are transitive, there may be multiple equivalent trees). Future work could involve developing a more flexible scoring function to account for this.

Limitations: There are also limitations to BERT-FT. For example, it is still fooled by syntactic ambiguity. Consider this example:

Example 2: *Foreign ministers agreed to set up a panel to investigate who shot down the Rwandan president's plane on April 6, 1994.*

A human reading this sentence will infer based on world knowledge that *April 6, 1994* should be attached to the embedded clause (*who shot down*), not to the matrix clause (*agreed*), but a syntactic parser would produce both parses. BERT-FT incorrectly attaches *agreed* to *April 6, 1994*: even BERT's contextualized embeddings are not sufficient to identify the correct parse.

6 Conclusion and Future Work

We present two models that incorporate BERT into temporal dependency parsers, and observe significant gains compared to previous approaches. We present an analysis of where and how BERT helps with this challenging task.

For future research, we plan to explore other types of deep neural LMs such as Transformer-XL (Dai et al., 2019) and XLNet (Yang et al., 2019). As discussed in Section 5, we also plan to develop a more flexible scoring function which can handle equivalent trees. Finally, we plan to evaluate BERT-FT on other temporal relation datasets as part of a larger pipeline, which will include a mapping between TDTs and other temporal relation annotation schemas such as the TempEval-3 dataset (UzZaman et al., 2013).

Acknowledgments

This work was supported by DARPA/I2O and U.S. Air Force Research Laboratory Contract No. FA8650-17-C-7716 under the Causal Exploration program, and DARPA/I2O and U.S. Army Research Office Contract No. W911NF-18-C-0003 under the World Modelers program. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.

Taylor Cassidy, Bill McDowell, Nathanel Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. Technical report, Carnegie-Mellon Univ Pittsburgh PA.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Leon Derczynski. 2017. *Automatically Ordering Events and Times in Text*, volume 677 of *Studies in Computational Intelligence*. Springer.

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.

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

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.

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.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003a. TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

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

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. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Yuchen Zhang and Nianwen Xue. 2018a. Neural ranking models for temporal dependency structure parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3339–3349.

Yuchen Zhang and Nianwen Xue. 2018b. Structured interpretation of temporal relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Yuchen Zhang and Nianwen Xue. 2019. Acquiring structured temporal representation via crowdsourcing: A feasibility study. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 178–185.