Can BERT Improve Temporal Dependency Parsing?

Hayley Ross*[1], **Jonathon Cai**[2], **Bonan Min**[2]
[1]Brandeis University, Waltham, MA
[2]Raytheon BBN Technologies, Cambridge, Massachusetts

Abstract

Extracting temporal relations between events and time expressions has many applications such as constructing event timelines and time-related question answering. It is also a challenging problem which requires syntactic and semantic information at sentence or discourse level, which are captured by deep language models such as BERT (Devlin et al., 2019). In this paper, we developed several variants of BERT-based temporal dependency parser, and show that BERT significantly improves temporal dependency parsing (Zhang and Xue, 2018a). Source code and trained models will be made available at github.com.

1 Introduction

Temporal relation extraction is important for many applications including constructing event timelines for news articles and narratives and 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). Consider the following example:

Example 1: *Kuchma and Yeltsin signed a cooperation plan Friday. 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 binary, pairwise approaches for temporal relation extraction such as (Cassidy et al., 2014), a TDT is much more concise but preserves the same (if not more) information. However, TDP is challenging because the task requires both syntactic and semantic information at the sentence and discourse level.

* Work done during an internship at BBN.

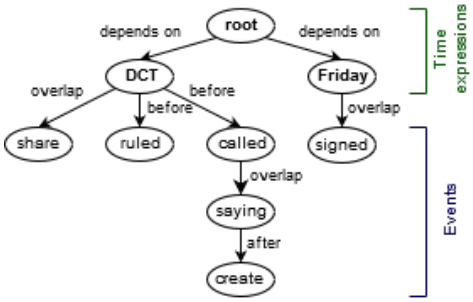

Figure 1: Temporal Dependency Parse Tree (TDT) of **Example 1**.

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 temporal dependency parsing.

In this paper, we investigate the potential for applying BERT to this task. We developed two models that incorporate BERT into a temporal dependency parsing, starting from a straightforward usage of pre-trained BERT word embeddings, to using BERT as an encoder and training it within an end-to-end TDP system. Experiments showed 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, 2019)[1]. We present technical details, experiments, and analysis in the rest of this paper.

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

2 Related Work

Much previous work has been devoted to classification of relations between events and time expressions, notably TimeML (Pustejovsky et al., 2003a) and the associated TimeBank corpus (Pustejovsky et al., 2003b), as well as the TimeBank-Dense corpus (Cassidy et al., 2014) which annotates all $\binom{n}{2}$ pairs of relations, resulting in an $O(n^2)$ complexity. Another problem with annotating every pair is that it creates the possibility of inconsistent predictions such as A before B, B before C, C before A. To address these issues, (Zhang and Xue, 2018b) present a structured schema of relations between time expressions and events to form a tree. In this structure, all time expressions are children of either the root (if they are absolute time expressions), of the special time expression node Document Creation Time (DCT), or children of other time expressions. All events are children of either a time expression or another event. The children and their parents are connected through a relation such as *before*, *after*, *overlap*, or *depends on*. By organizing time expressions and events into a tree, this reduces the annotation complexity to $O(n)$.

This paper builds primarily on the chain of work done by (Zhang and Xue, 2018b), (Zhang and Xue, 2018a) and (Zhang and Xue, 2019), which presents an English corpus annotated with this schema as well as a simple baseline and first neural architecture. (Zhang and Xue, 2018a) uses a BiLSTM model with simple attention and randomly initialized word embeddings trained only on the data in the corpus. This paper capitalizes on recent advances in deep models and pre-trained word embeddings such as ELMo (Peters et al., 2018), ULMFit (Howard and Ruder, 2018) and 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 the original model.

3 BERT-based Models

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, the problem is to select the most appropriate parent mention (an event, time expression, DCT or the root node), selected among $x_1, x_2, \ldots, x_{i+m}$, $i-1$ mentions before x_i and m mentions appearing immediately after x_i, along with the relation label (*overlap*, *before*, *after*, *depends on*). A Temporal Dependency Tree (TDT) is assembled by selecting the highest-ranked prediction ⟨parent, relation type⟩ for each event or time expression in a document (while ensuring that no cycles are created).

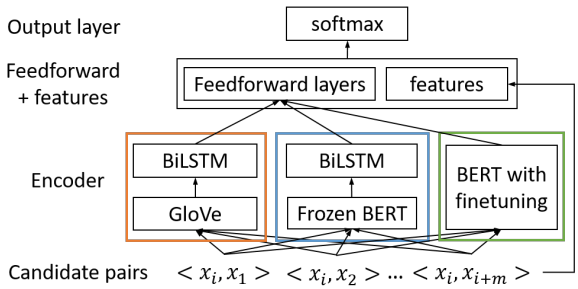

Figure 2: Architectures of Temporal Dependency Parsers. x_i is the child mention, while $x_1, x_2, \ldots, x_{i+m}$ are its possible parents with the first $i-1$ mentions before x_i and the last m mentions appearing after x_i. The orange, blue and green boxes represent the three encoders, respectively.

As shown in Figure 2, we developed three models that share a similar overall architecture: the model takes a pair of mentions (child and parent) as input and passes each pair through an encoder which embeds the nodes and surrounding context into a dense representation. Hand-crafted features are concatenated onto the dense representation, which is then passed to either one or two feed-forward layers to generate scores for each relation label for each pair.

Three types of models are developed:

- **BiLSTM with non-contextualized word embeddings** which uses word embeddings (one vector per word) for each input word in the corpus, then uses a BiLSTM to encode the pair as well as the surrounding context. The word embeddings can be either randomly initialized (then the model is the same as the neural model described in (Zhang and Xue, 2018a)), or can be pre-trained from a large corpus, e.g. GloVe (Pennington et al., 2014).
- **BiLSTM with frozen BERT embeddings** in which the word embeddings are replaced with frozen (pre-trained) BERT contextualized word embeddings.
- **BERT with fine-tuning**: The BERT / Transformer (Vaswani et al., 2017) model (with pre-trained weights) is used directly to encode the pairs. The weights are fine-tuned

word(s)	label	sep	sentence
Friday	TIMEX	:	*Kuchma and Yeltsin signed a cooperation plan Friday*
called	EVENT	:	*Yeltsin and Kuchma called for the ratification . . .*

Table 1: "Sentence" inputs to BERT in **BERT as encoder**, for potential parent *Friday* and child *called* in **Example 1**. (The correct parent for *called* is DCT.)

within the end-to-end mention ranking training process.

All models use the same loss function and scoring as in (Zhang and Xue, 2018a).

3.1 Model 1: BiLSTM with Frozen BERT

The first model simply 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 by a BiLSTM. The BiLSTM output is passed to an attention mechanism (which handles events / time expressions with multiple words), then combined with the hand-crafted features (listed in Table 2) and passed to a feed-forward network with one hidden layer, which ranks each relation label for each (possible) parent / child pair.

3.2 Model 2: BERT with Fine-Tuning

This model uses the classification capabilities of BERT / Transformer, where 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 as encoder** constructs a "sentence" for the (potential) parent node and a "sentence" for the child node under consideration. These are passed to BERT in that order and concatenated with the `[SEP]` token as is standard for BERT. Each "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 for the child *called* and potential parent *Friday* in **Example 1**.

3.3 Additional Features

We used several hand-crafted binary and scalar features (Table 2) in all models, expanding on the features in (Zhang and Xue, 2018a).

Node distance features
parent is previous node in document
parent is before child in same sentence
parent is before child, more than one sentence away
parent is after child
parent and child are in same sentence
scaled distance between nodes
scaled distance between sentences
Time expression / event label features
child is time expression and parent is root
child and parent both time expressions
child is event and parent is DCT
parent is padding node[2]

Table 2: Binary and scalar features used in all models.

4 Experiments

We use the training, dev and test data from (Zhang and Xue, 2019) for all experiments. We evaluated the following models:

- **BiLSTM (re-implemented)**: We re-implement (Zhang and Xue, 2018a)'s model[3] in TensorFlow for fair comparison. Similar to (Zhang and Xue, 2018a), randomly initialized word embeddings are used in the first layer. We also replace the randomly-initialized word embeddings with GloVe (Pennington et al., 2014), resulting in the model **BiLSTM with GloVe**.
- **BiLSTM with frozen BERT**: This model shares the same architecture with the above, except that we use pre-trained BERT model to look up frozen, contextualized word embeddings for each word.
- **BERT as encoder**: We use BERT as the encoder (Figure 2) with the input data taking the form described in Section 3.2. During training, the BERT parameters are fine-tuned in the end-to-end system.

We used Adam 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. We observed that when fine-tuning BERT in the **BERT as encoder** model, a lower learning rate (0.0001) paired with more epochs (75) achieves higher performance, compared to using learning rate 0.001 with 50 epochs for the BiLSTM models.

[2]Window $x_1, x_2, \ldots, x_{i+m}$ is of fixed size, so it must be padded near the start or end of a document

[3]The original model was implemented in DyNet (Neubig et al., 2017).

Model	F1-score
Baseline (Zhang and Xue, 2019)	0.18
BiLSTM (re-implemented)	0.55
BiLSTM (Zhang and Xue, 2019)	0.60
BiLSTM with GloVe	0.58
BiLSTM with frozen BERT	0.61
BERT as encoder	**0.68**

Table 3: Performance of various models.

Table 3 summarizes the performance of our models. We also include the rule-based baseline and the model in (Zhang and Xue, 2019)[4] as a baseline.

BiLSTM with frozen BERT outperforms the re-implemented baseline **BiLSTM** model by 6 points and **BiLSTM with 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 as encoder**) 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 accuracies for each model 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. By this breakdown, we see that the main difference between the **BiLSTM** and the **BiLSTM with frozen BERT** is its performance on children of DCT: with BERT, it scores 0.48 instead of 0.38. Conversely, for **BERT as encoder**, we see improvements across the board, with a 0.21 increase on children of DCT over the BiLSTM, a 0.14 increase for children of other time expressions, and a 0.11 increase for children of events.

5 Analysis

Why BERT helps: Comparing the temporal dependency trees produced by the models for the test set, we see that these improvements correspond to the phenomena below.

Firstly, unlike the BiLSTM, **BERT as encoder** is able to properly relate time expressions occurring syntactically after the event, such as the classic news construction *Kuchma and Yeltsin signed a cooperation plan [on] Friday* in **Example 1**.. (The BiLSTM relates *signed* to the "previous" time expression DCT). This shows BERT's ability to "look forward", attending to information indicating a parent appearing after the child.

Secondly, with BERT the model is able to capture verb tense, and use it to determine the correct label in almost all cases, both for DCT and for chains of events. It knows that present tense sentences (*share similar cultures*) overlap DCT, while past perfect events (*was ruled from Moscow*) happen either before DCT or before the event immediately adjacent (salient) to them. Similarly, progressive tense (*saying*) may indicate parallel events i.e. overlap.

Thirdly, the BERT model capture syntax related to time. They are particularly adept at progressive and reported speech constructions such as *Yeltsin and Kuchma called for the ratification of the treaty, saying [that] it would create* ... where 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) may allow it to detect in which direction adverbs such as *since* should be applied to the events. This means that while both the BiLSTM and the BERT model may identify the correct parent, **BERT as encoder** is much more likely to choose the correct label whereas the BiLSTM model almost always chooses either before for DCT or after for children of events.

Last, both BERT-based models 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 reported above on children of DCT). This may be because BERT is able to compare the sentences and judge that they are unrelated despite being adjacent, whereas BiLSTM models with non-contextualized embeddings cannot.

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

Limitations: There are also limitations to **BERT as encoder**. For example, it is still fooled

[4]We were unable to replicate the F1-score reported in (Zhang and Xue, 2019) despite using similar hyperparameters. Therefore, we include performances for both of our re-implementation and the reported score in (Zhang and Xue, 2019) in Table 3.

by syntactic ambiguity. Consider:

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 subclause *who shot down* . . . , not to the matrix clause (*agreed*), but a syntactic parser would produce both parses. **BERT as encoder** attaches *agreed* to *April 6, 1994*: evidently BERT cannot identify which matrix verb is more salient either.

6 Conclusion and Future Work

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

For future research, we plan to explore the interaction between the representation learnt by BERT and the hand-crafted features added at the final layer, as well as develop a more flexible scoring function which can handle equivalent trees.

Acknowledgements

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

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. *arXiv preprint arXiv:1906.04341*.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

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*.

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.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

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.

Yuchen Zhang and Nianwen Xue. 2018a. Neural ranking models for temporal dependency structure parsing. *arXiv preprint arXiv:1809.00370*.

Yuchen Zhang and Nianwen Xue. 2018b. Structured interpretation of temporal relations. *arXiv preprint arXiv:1808.07599*.

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.