A BERT-based One-Pass Multi-Task Model for Clinical Temporal Relation Extraction

Chen Lin[1], Timothy Miller[1*], Dmitriy Dligach[2], Farig Sadeque[1], Steven Bethard[3] and Guergana Savova[1]

[*]Co-first author
[1]Boston Children's Hospital and Harvard Medical School
[2]Loyola University Chicago
[3]University of Arizona
[1]{first.last}@childrens.harvard.edu
[2]ddligach@luc.edu
[3]bethard@email.arizona.edu

Abstract

Recently BERT has achieved a state-of-the-art performance in temporal relation extraction from clinical Electronic Medical Records text. However, the current approach is inefficient as it requires multiple passes through each input sequence. We extend a recently-proposed one-pass model for relation classification to a one-pass model for relation extraction. We augment this framework by introducing global embeddings to help with long-distance relation inference, and by multi-task learning to increase model performance and generalizability. Our proposed model produces results on par with the state-of-the-art in temporal relation extraction on the THYME corpus and is much "greener" in computational cost.

1 Introduction

The analysis of many medical phenomena (e.g., disease progression, longitudinal effects of medications, treatment regimen and outcomes) heavily depends on temporal relation extraction from the clinical free text embedded in the Electronic Medical Records (EMRs). At a coarse level, a clinical event can be linked to the document creation time *(DCT)* as Document Time Relations *(DocTimeRel)*, with possible values of *BEFORE, AFTER, OVERLAP, and BEFORE_OVERLAP* (Styler IV et al., 2014). At a finer level, a narrative container (Pustejovsky and Stubbs, 2011) can temporally subsume an event as a *contains* relation. The THYME corpus (Styler IV et al., 2014) consists of EMR clinical text and is annotated with time expressions (TIMEX3), events (EVENT), and temporal relations (TLINK) using an extension of TimeML (Pustejovsky et al., 2003; Pustejovsky and Stubbs, 2011). It was used in the Clinical TempEval series (Bethard et al., 2015, 2016, 2017).

While the performance of DocTimeRel models has reached above 0.8 F1 on the THYME corpus, the CONTAINS task remains a challenge for both conventional learning approaches (Sun et al., 2013; Bethard et al., 2015, 2016, 2017) and neural models (structured perceptrons (Leeuwenberg and Moens, 2017), convolutional neural networks (CNNs) (Dligach et al., 2017; Lin et al., 2017), and Long Short-Term memory (LSTM) networks (Tourille et al., 2017; Dligach et al., 2017; Lin et al., 2018; Galvan et al., 2018)). The difficulty is that the limited labeled data is insufficient for training deep neural models for complex linguistic phenomena. Some recent work (Lin et al., 2019) has used massive pre-trained language models (BERT; Devlin et al., 2018) and their variations (Lee et al., 2019) for this task and significantly increased the CONTAINS score by taking advantage of the rich BERT representations. However, that approach has an input representation that is highly wasteful – the same sentence must be processed multiple times, once for each candidate relation pair.

Inspired by recent work in Green AI (Schwartz et al., 2019; Strubell et al., 2019), and one-pass encodings for multiple relations extraction (Wang et al., 2019), we propose a one-pass encoding mechanism for the CONTAINS relation extraction task, which can significantly increase the efficiency and scalability. The architecture is shown in Figure 1. The three novel modifications to the original one-pass relational model of Wang et al. (2019) are: (1) Unlike Wang et al. (2019), our model operates in the relation extraction setting, meaning it must distinguish between relations and non-relations, as well as classifying by relation type. (2) We introduce a pooled embedding for relational classification across long distances. Wang et al. (2019) focused on short-distance relations, but clinical CONTAINS relations often span multiple sentences, so a sequence-level embedding is necessary for such long-distance inference. (3) We use the same BERT encoding of the input instance for both

Proceedings of the BioNLP 2020 workshop, pages 70–75

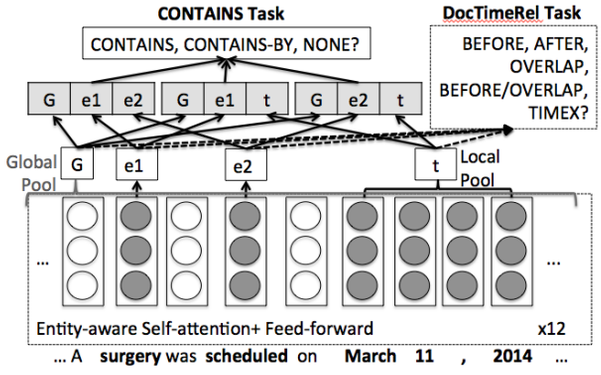

Figure 1: Model Architecture. e1, e2, and t represent entity-embeddings for "surgery", "scheduled", and "March 11, 2014" respectively. G is the pooled embedding for the entire input instance.

DocTimeRel and CONTAINS tasks, i.e. adding multi-task learning (MTL) on top of one-pass encoding. DocTimeRel and CONTAINS are related tasks. For example, if a medical event A happens BEFORE the DCT, while event B happens AFTER the DCT, it is unlikely that there is a CONTAINS relation between A and B. MTL provides an effective way to leverage useful knowledge learned in one task to benefit other tasks. What is more, MTL can potentially employ a regularization effect that alleviates overfitting to a specific task.

2 Methodology

2.1 Twin Tasks

Apache cTAKES (Savova et al., 2010)(http://ctakes.apache.org) is used for segmenting and tokenizing the THYME corpus in order to generate instances. Each instance is a sequence of tokens with the gold standard event and time expression annotations marked in the token sequences by logging their positional information. Using the entity-aware self-attention based on relative distance (Wang et al., 2019), we can encode every entity, E_i, by its BERT embedding, e_i. If an entity e_i consists of multiple tokens (many time expressions are multi-token), it is average-pooled (local pool in Figure 1) over the embedding of the corresponding tokens in the last BERT layer.

For the CONTAINS task, we create relation candidates from all pairs of entities within an input sequence. Each candidate is represented by the concatenation of three embeddings, e_i, e_j, and G, as $[G{:}e_i{:}e_j]$, where G is an average-pooled embedding over the entire sequence, and is different from the embedding of [CLS] token. The [CLS] token is the conventional token BERT inserts at the start of every input sequence and its embedding is viewed as the representation of the entire sequence. The concatenated embedding is passed to a linear classifier to predict the CONTAINS, CONTAINED-BY, or NONE relation, $\hat{r_{ij}}$, as in eq. (1).

$$P(\hat{r_{ij}}|\mathbf{x}, E_i, E_j)=softmax(W^L[G:e_i:e_j]+b) \quad (1)$$

where $W^L \in \mathbb{R}^{3d_z \times l_r}$, d_z is the dimension of the BERT embedding, $l_r = 3$ for the CONTAINS labels, b is the bias, and $\mathbf{x}$ is the input sequence.

Similarly, for the DocTimeRel (dtr) task we feed each entity's embedding, e_i, together with the global pooling G, to another linear classifier to predict the entity's five "temporal statuses": TIMEX if the entity is a time expression or the dtr type (BEFORE, AFTER, etc.) if the entity is an event:

$$P(\hat{dtr_i}|\mathbf{x}, E_i) = softmax(W^D[G:e_i]+b) \quad (2)$$

where $W^D \in \mathbb{R}^{2d_z \times l_d}$, and $l_d = 5$.

For the combined task, we define loss as:

$$L(\hat{r_{ij}}, r_{ij}) + \alpha(L(\hat{dtr_i}, dtr_i) + L(\hat{dtr_j}, dtr_j)) \quad (3)$$

where $\hat{r_{ij}}$ is the predicted relation type, $\hat{dtr_i}$ and $\hat{dtr_j}$ are the predicted temporal statuses for E_i and E_j respectively, r_{ij} is the gold relation type, and dtr_i and dtr_j are the gold temporal statuses. α is a weight to balance CONTAINS loss and dtr loss.

2.2 Window-based token sequence processing

Following Lin et al. (2019), we use a set window of tokens (Token-Window) disregarding natural sentence boundaries for generating instances. BERT may still take punctuation tokens into account. Each token sequence is limited by a set number of entities (Entity-Window) to be processed. We apply a sliding token window (windows may overlap), thus every entity gets processed. Positional information for each entity is output along the token sequence and is propagated through different layers via the entity-aware self-attention mechanism (Wang et al., 2019).

3 Experiments

3.1 Data and Settings

We adopt the THYME corpus (Styler IV et al., 2014) for model fine-tuning and evaluation. The

Model	P	R	F1
Multi-pass	0.735	0.613	0.669
Multi-pass+Silver	0.674	0.695	0.684
One-pass	0.647	0.671	0.659
One-pass+[CLS]	0.665	0.673	0.669
One-pass+Pooling	0.670	0.689	0.680
One-pass+Pooling+MTL	0.686	0.687	**0.686**

Table 1: Model performance of *CONTAINS* relation on colon cancer test set. Multi-pass baselines are from Lin et al. (2019)'s system without and with self-training using silver instances (system predictions on a unlabeled colon cancer set). We tested a one pass system with just argument embeddings; with the [CLS] token as the global context vector ([CLS]); with argument embeddings plus a globally pooled context vector (Pooling); and with global pooling as well as multi-task learning (MTL) with DocTimeRel.

Model	Single	MTL
AFTER	0.86	0.83
BEFORE	0.88	0.89
BEFORE/OVERLAP	0.63	0.56
OVERLAP	0.89	0.85
TIMEX	0.98	0.98
OVERALL	0.88	0.86

Table 2: Model performance in F1-scores of temporal statuses on colon cancer test set. Single: One-pass+Pooling for a single dtr Task; MTL: One-pass+Pooling for twin tasks: CONTAINS and dtr.

Model	P	R	F1
Lin et al. (2019)	0.473	0.700	0.565
One-pass+Pooling	0.506	0.643	0.566
One-pass+Pooling+MTL	0.545	0.624	**0.582**

Table 3: Model performance of *CONTAINS* relation on brain cancer test set.

one-pass multi-task model is fine-tuned on the THYME Colon Cancer training set with uncased BERT base model, using the code released by Wang et al. (2019)[1] as a base. The batch size is set to 4, the learning rate is selected from (1e-5, 2e-5, 3e-5, 5e-5), the Token-Window size is selected from (60, 70, 100), the Entity-Window size is selected from (8, 10, 16), the training epochs are selected from (2, 3, 4, 5), the clipping distance k (the maximum relative position to consider) is selected from (3, 4, 5), and α is selected from (0.01, 0.05). A single NVIDIA GTX Titan Xp GPU is used for the computation. The best model is selected on the Colon cancer development set and tested on the Colon cancer test set, and on THYME Brain cancer test set for portability assessment.

3.2 Results on THYME

Table 1 shows performance of our one-pass models for the CONTAINS task on the Clinical TempEval colon cancer test set. The one-pass (OP) model alone obtains an F1 score of 0.659. Adding the [CLS] token as the global context vector increases the F1 score to 0.669. Using a globally average-pooled context vectors G instead of [CLS] improves performance to 0.680, better than the multi-pass model without silver instances (Lin et al., 2019). Applying the MTL setting, the one-pass twin-task (CONTAINS and DocTimeRel) model without any silver data reaches 0.686 F1, which is on par with the multi-pass model trained with additional silver instances on the CONTAINS task, 0.684 F1 (Lin et al., 2019).

Table 2 shows the performance of our one-pass models for the DocTimeRel task on the Clinical TempEval colon cancer test set. The single-task model achieves 0.88 weighted average F1, while the MTL model compromises the performance to 0.86 F1. Of note, this result is not directly comparable to Bethard et al. (2016) results because the Clinical TempEval evaluation script does not take into account if an entity is correctly recognized as a time expression (TIMEX). There are two types of entities in the THYME annotation: events and time expressions (TIMEX). The Bethard et al. (2016) evaluation on DocTimeRel was focused on all events, and classified an event into four DocTimeRel types. Our evaluation was for all entities. For a given entity, we classify it as a TIMEX or an event; if it is an event, we classify it into four DocTimeRel types, for a total of five classes.

Table 3 shows the portability of our one-pass models on the THYME brain cancer test set. Without any tuning on brain cancer data, the MTL model with global pooling performs at 0.582 F1, which is better than the multi-pass model trained with additional silver instances (0.565 F1) reported in Lin et al. (2019), trading roughly equal amounts of precision for recall to obtain a better balance. Without MTL, the one-pass CONTAINS model with global context embeddings (One-pass+Pooling) achieves 0.566 F1 on the brain cancer test set, significantly lower than the MTL

[1] https://github.com/helloeve/mre-in-one-pass

Model	flops/inst	inst#	Ratio
OP	218,767,889	20k	**1**
OP+MTL	218,783,260	20k	**1**
Multi-pass	218,724,880	427k	23
Multi-pass+Silver	218,724,880	497k	25

Table 4: Computational complexity in flops per instance (flops/inst)×total number of instances (inst#).

model (using a Wilcoxon Signed-rank test over document-by-document comparisons, as in (Cherry et al., 2013), p-value=0.01962).

3.3 Computational Efficiency

Table 4 shows the computational burden for different models in terms of floating point operations (flops). The flops are derived from TensorFlow's profiling tool on saved model graphs. The second column is the flops per one training instance, the third column lists the number of instances for different model settings. The total computational complexity for one training epoch is thus the multiplication between column 2 and 3. The *Ratio* column is the relative ratio of total complexity using the OP total flops as the comparator.

For relation extraction, all entities within a sequence must be paired. If there are n entities in a token sequence, there are $n \times (n-1)/2$ ways to combine those entities for relational candidates. The multi-pass model would encode the same sequence $n \times (n-1)/2$ times, while the one-pass model would only encode it once and add the pairing computation on top of the BERT encoding represented in Figure 1 with very minor increase in computation per one instance (about 43K flops); and the MTL model adds another 15k flops; but they are of the same magnitude, 219K flops. The one-pass models save a lot of passes on the training instances, 20k vs. 497k, which results in a significant difference in computational load, 1 vs. 25, which could be several hours to several days difference in GPU hours. The exact number of training instances processed by the one-pass model is affected by the Token-Window and Entity-Window hyper-parameters. However, even in the worst case scenario, when the Token-Window is set to 100, and the Entity-Window is set to 8, there are 108K training instances for the one-pass model, which is still substantially fewer training instances than what are used for the multi-pass model. In addition, since the one-pass models do not run the extra steps used for generating silver instances (Lin et al., 2019), the time savings is even greater.

4 Discussion

Through table 1 row 3-5, we can see that sequence-wise embedding, either global pooling G or [CLS], is important for clinical temporal relation extraction which involves long-distance relations that may go across multiple natural sentences. Entity embeddings are good for tasks that focus on short-distance relations (such as (Gábor et al., 2018)), but may not be sufficient for picking enough context for long-distance relations.

Combining MTL with a one-pass mechanism produces a more efficient and generalizable model. With merely additional 15k flops (table 4 row 1 and 2), the model achieves high performance for both tasks. However, we found that it is hard for both tasks to get top performance. If the weight for dtr loss is increased, the dtr F1 increases at the cost of the CONTAINS scores. Even though the majority of entities in CONTAINS relations have aligned dtr values (e.g., in Figure 2(#1), both entities have matching dtr value, AFTER), some relations do have conflicted dtr values. For example, in Figure 2(#2), the dtr for *screening* is BEFORE, while *test* is a BEFORE_OVERLAP (the present perfect tense signifies *tests* happened in the past but lasts through present, hence BEFORE_OVERLAP). Even though it is a gold CONTAINS annotation, the model may be confused by an event that happened in the past (*screening*) to contain another event (*test*) that is longer than its temporal scope. Due to these conflicts, we thus pick the more challenging CONTAINS task as our priority and set α relatively low (0.01) in order to optimize the model towards the CONTAINS task, ignoring some of the dtr errors or conflicts. In the meantime, the MTL setting does help prevent the model from overfitting to one specific task, thus achieving some level of generalization. The significant 1.6% increase in F1-score on the Brain test set in table 3 demonstrates the improved generalizability.

In conclusion, we built a "green" model for a challenging problem. Deployed on a single gpu with 25 times better efficiency, it succeeded in both temporal tasks, achieved better generalizability, and suited to other pre-trained models (Liu et al., 2019; Alsentzer et al., 2019; Beltagy et al., 2019; Lan et al., 2019; Yang et al., 2019, etc.)

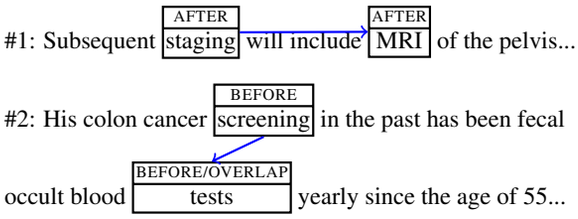

Figure 2: CONTAINS Relations with matching(#1)/conflicting(#2) DocTimeRel values.

Acknowledgments

The study was funded by R01LM10090, R01GM114355 and UG3CA243120 from the Unites States National Institutes of Health. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. We thank the anonymous reviewers for their valuable suggestions and criticism. The Titan Xp GPU used for this research was donated by the NVIDIA Corporation.

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. 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 3606–3611.

Steven Bethard, Leon Derczynski, Guergana Savova, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. Semeval-2015 task 6: Clinical tempeval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. *Proceedings of SemEval*, pages 1052–1062.

Steven Bethard, Guergana Savova, Martha Palmer, James Pustejovsky, and Marc Verhagen. 2017. Semeval-2017 task 12: Clinical tempeval. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 563–570.

Colin Cherry, Xiaodan Zhu, Joel Martin, and Berry de Bruijn. 2013. la recherche du temps perdu: extracting temporal relations from medical text in the 2012 i2b2 nlp challenge. *Journal of the American Medical Informatics Association*, 20(5):843–848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. *EACL 2017*, page 746.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

Diana Galvan, Naoaki Okazaki, Koji Matsuda, and Kentaro Inui. 2018. Investigating the challenges of temporal relation extraction from clinical text. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 55–64.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Tuur 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*.

Chen Lin, Timothy Miller, Dmitriy Dligach, Hadi Amiri, Steven Bethard, and Guergana Savova. 2018. Self-training improves recurrent neural networks performance for temporal relation extraction. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 165–176.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. *BioNLP 2017*, pages 322–327.

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.

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*.

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

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160. Association for Computational Linguistics.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green ai. *arXiv preprint arXiv:1907.10597*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

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

Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A bi-lstm approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 224–230.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1371–1377.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems 32*, pages 5754–5764.