Exploring Text Representations for Generative Temporal Relation Extraction

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

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

Abstract

Sequence-to-sequence models are appealing because they allow both encoder and decoder to be shared across many tasks by formulating those tasks as text-to-text problems. Despite recently reported successes of such models, we find that engineering input/output representations for such text-to-text models is challenging. On the Clinical TempEval 2016 relation extraction task, the most natural choice of output representations, where relations are spelled out in simple predicate logic statements, did not lead to good performance. We explore a variety of input/output representations, with the most successful prompting one event at a time, and achieving results competitive with standard pairwise temporal relation extraction systems.

1 Introduction

Extracting temporal information from texts is critical in the medical domain for prognostication models, studying disease progression, and understanding longitudinal effects of medications and treatments. The standard route for extracting temporal information is by casting it as a relation task between time expressions and medical events. This relation extraction task is approached by forming relation candidates by pairing potential relation arguments and training a classifier to determine whether a relation exists between them. This pairwise approach is taken by a state-of-the-art temporal relation extraction system (Lin et al., 2019), which uses a pretrained language model such as BERT (Devlin et al., 2019) for representing the training examples.

The goal of this paper is to investigate a generative approach to relation extraction as an alternative to the traditional pairwise method. We investigate whether it is possible for a sequence-to-sequence (seq2seq) model such as T5 (Raffel et al., 2020), BART (Lewis et al., 2020), and SciFi (Phan et al., 2021) to ingest a chunk of clinical text, often containing multiple sentences, and generate human-readable output containing all relation instances in the input. This goal proved to be more ambitious than we anticipated, but ultimately we succeeded in designing input/output representations that were competitive with state-of-the-art.

Using generative models for relation extraction has received little attention and no work exists on using these models for temporal relation extraction. Paolini et al. (2021) use natural language to encode sentence-level relations but mapping the output text to the input arguments is not trivial and requires an alignment algorithm. Huang et al. (2021) formulate relation extraction as a template generation problem but their approach requires a complex cross-attention guided copy mechanism. We explore sentence- as well as cross-sentence relations and encode relations in a structured and human-readable form in which the relation arguments can be easily mapped to the reference entities in the input.

In our experiments, we use SemEval-2016 Task 12: Clinical TempEval data (Bethard et al., 2016), which annotated time expressions, events, and temporal relations, specifically the CONTAINS relation that links times and events to their narrative containers (Pustejovsky and Stubbs, 2011). For example, in Table 1 the time expression *postop* in the second sentence contains the event *chemotherapy*.

Proceedings of the 4th Clinical Natural Language Processing Workshop, pages 109 - 113
July 14, 2022 ©2022 Association for Computational Linguistics

2 Methods

2.1 Input and output representation variants

While a natural input/output representation would have been to keep everything fully in the realm of words (e.g., the NATURAL row in table 1), this would have made reconstructing the character offsets of these relations difficult. For example, if the system produced *1998 contains tumor* for an input where the surface form *tumor* appeared multiple times (a common occurrence in clinical data), we would not be able to determine which *tumor* event to link to the date.

Thus, we focused on representations where we could deterministically recover the character offsets of the events and times being related. We took as input chunks of text, typically spanning multiple sentences to capture cross-sentence relations. We appended a slash character and an integer index to each event and time expression to disambiguate surface forms that occured multiple times in the text. We also marked all reference events and time expressions with special tags to make the candidates for relation arguments transparent to the model. Examples of such input formatting can be found in the bottom three rows of table 1.

Given this setup, our original goal was a seq2seq model that would take as input the formatted text and generate all temporal relations as output. Our first input/output representation encoded the relations as predicate logic statements with *contains* as the predicate, event/time indices as the arguments, and predicates sorted by the position of the first argument (table 1, RELATIONS variant). The sorting is necessary to introduce a notion of order into an otherwise order-less relation extraction problem, i.e., to transform a set prediction problem into a sequence prediction problem.

Our second input/output representation encoded the temporal relations as classifications over each event or time, where the model must predict a temporal container for each event and each time, generating the underscore character if no container is found (table 1, CONTAINERS variant). Preliminary error analysis had indicated that models based on the RELATIONS variant struggled to decide when to produce or omit an argument, and the CONTAINERS variant removed that choice.

Our final input/output representation was similar to CONTAINERS, but rather than asking the model to predict all temporal containers, it prompted the model with a focus event or time and asked only for the temporal container for that. We achived this by attaching the index of the focus event or time at the end of the formatted input text after a vertical bar separator character, and using as output only the index of the container event or time or underscore to indicate no relation (table 1, 1-CONTAINER variant). Thus, for every chunk of text, the number of examples that we generate equals the total number of events and times in the chunk.

Note that traditional pairwise relation extraction models, require $O(n^2)$ examples to encode the relations, where n is the total number of events and times in the chunk. Our RELATIONS and CONTAINERS representations require m training examples, where m is the number of chunks ($m << n$) and our 1-CONTAINER representation requires n examples, thus potentially reducing training time and memory requirements.

2.2 Models

For seq2seq models, we compare BART, T5, and SciFive (a clinical version of T5). The models are trained to receive a chunk of text and generate output as described in section 2.1.

Chunk size plays an important role in input/output representations: short chunks have fewer relation instances and seq2seq models have less trouble predicting them correctly. But short chunks miss long-distance relations, which often span multiple sentences. Longer chunks are harder for seq2seq models, but they capture more long-distance relations. This tension plays a role in the performance of our models and we treat chunk size as a hyperparameter that we tune on the development set.

Additionally, we observe that our 1-CONTAINER variant does not require a full auto-regressive decoder since models only need to generate a single integer (the index of the container). We thus study two encoder-only models. In BERT-softmax, we follow the standard text classification approach with BERT: add a randomly-intitialized softmax layer on top of the last layer's contextualized represenatation of the `[cls]` token, where the sofmax is over all items in the vocabulary. In BERT-similarity, we instead compute similarity (the dot product) between the `[cls]` token and all word piece embeddings in the vocabulary, apply softmax to the similarity scores, and select the item with the largest score. Note that the classification layer of BERT-softmax must be trained from scratch, while

Variant	Input	Output
Natural	2001: Left breast lumpectomy followed by radiation therapy. She received no postop chemotherapy. She was given tamoxifen for five years and then Femara.	2001 contains lumpectomy. postop contains chemotherapy. five years contains tamoxifen.
Relations	**task**: relext; **text**: <t> 2001/0 </t>: Left breast <e> lumpectomy/3 </e> followed by radiation <e> therapy/4 </e>. She received no <t> postop/1 </t> <e> chemotherapy/5 </e>. She was given <e> tamoxifen/6 </e> for <t> five years/2 </t> and then <e> Femara/7 </e>.	contains(0; 3) contains(1; 5) contains(2; 6)
Containers	**task**: relext; **text**: <t> 2001/0 </t>: Left breast <e> lumpectomy/3 </e> followed by radiation <e> therapy/4 </e>. She received no <t> postop/1 </t> <e> chemotherapy/5 </e>. She was given <e> tamoxifen/6 </e> for <t> five years/2 </t> and then <e> Femara/7 </e>.	contains(0; _) contains(3; 0) contains(4; _) contains(1; _) contains(5; 1) contains(6; 2) contains(2; _) contains(7; _)
1-Container	**task**: relext; **text**: <t> 2001/0 </t>: Left breast <e> lumpectomy/3 </e> followed by radiation <e> therapy/4 </e>. She received no <t> postop/1 </t> <e> chemotherapy/5 </e>. She was given <e> tamoxifen/6 </e> for <t> five years/2 </t> and then <e> Femara/7 </e>. **|** 3	0

Table 1: Sample input/output (I/O) representation variants. Bold text indicates task prompt conventions. Note that the 1-Container variant shows only one relation; seven more instances would be required to represent classifications for all eight input events and times.

BERT-similarity does not require any layer to be trained from scratch.

2.3 Experiments

We use BART (facebook/bart-base), T5 (t5-base), SciFive (razent/SciFive-base-Pubmed_PMC), and BERT (bert-base-uncased) from the HuggingFace model hub[1]. Our code is based on the HuggingFace Transformers library (Wolf et al., 2020) and will be released publically upon publication. We use AdamW optimizer and tune its learning rate and weight decay as well as other model hyperparameters such as chunk size, beam size, and the number of epochs on the official Clinical TempEval development set. After tuning the models, we retrained on the training and development sets combined. We report the results on the Clinical TempEval test set using the official evaluation script.

We compare to three baselines from Lin et al. (2019). BERT-T and BioBERT are standard pairwise relation extraction BERT-based ('bert-base' and 'biobert', respectively) models that generate relation candidates by pairing all events and times in a 60-token chunk of text and train a three-way classifier to predict whether a relation exists between them. The negative class represents the no-relation scenario. The positive class is split into two labels, Contains, and Contained-By, depending on the order of the arguments. BERT-TS augments the aforementioned BERT system with high-confidence 'silver' instances obtained through self-training. The BioBERT-based system is currently the state-of-the-art on this dataset.

Chunks: We apply simple preprocessing to the TempEval data to generate the inputs and outputs for our models as follows: (1) we split the corpus into sections (e.g. medications, family history), which are marked with standardized section headers; (2) we split sections into sentences using a simple regular expression; (3) we form chunks by concatenating adacent sentences up to the *chunk_size* hyperparameter. A sample chunk is shown in table 1.

[1] https://huggingface.co/models

N	Model	I/O Representation	Chunk	P	R	F1
1	BERT-T (Lin et al., 2019)	Pairwise	n/a	0.735	0.613	0.669
2	BERT-TS (Lin et al., 2019)	Pairwise	n/a	0.670	0.697	0.683
3	BioBERT (Lin et al., 2019)	Pairwise	n/a	0.674	0.695	0.684
4	BERT-softmax	1-CONTAINER	50	0.714	0.530	0.608
5	BERT-similarity	1-CONTAINER	50	0.712	0.540	0.615
6	BART	RELATIONS	50	0.709	0.231	0.348
7	BART	CONTAINERS	75	0.480	0.266	0.342
8	BART	1-CONTAINER	175	0.651	0.671	0.661
9	T5	RELATIONS	50	0.675	0.570	0.618
10	T5	CONTAINERS	75	0.684	0.625	0.654
11	T5	1-CONTAINER	75	0.718	0.632	0.672
12	T5	1-CONTAINER	175	0.717	0.675	**0.696**
13	SciFive	RELATIONS	50	0.669	0.503	0.574
14	SciFive	CONTAINERS	75	0.657	0.609	0.632
15	SciFive	1-CONTAINER	175	0.691	0.683	0.687

Table 2: Generative relation extraction and baseline performance on Clinical TempEval test set using reference relation arguments (events and times). Top three systems include current SOTA (line 3) on this dataset.

3 Results and Discussion

Only one input/output variant was competitive with baseline systems: the 1-CONTAINER variant (table 2, lines 12 and 15) performed at least as well or better than all three baselines (lines 1-3). T5's good performance is notable since it is more comparable with BERT-T (line 1), which, unlike the other two baselines did not have acccess to additional training examples (BERT-TS) or in-domain data (BioBERT). On the other hand, suprisingly, SciFive did not have an advantage over T5 despite having been pretrained on in-domain data.

Our encoder-only systems (lines 4 and 5) performed much worse than the comparable 1-CONTAINER variant for the seq2seq models. This is likely due to the lack of a full pretrained decoder, although the similarity-based variant (line 5) mitigated that disadvantage a little.

BART performed worse than the other seq2seq models across all input/output variants although its performance could potentially be improved by a much more extensive hyperparameter search. We leave an exploration into why its "out-of-the-box" performance was inferior for future work.

Chunk size issues: The number of reference relations can grow quadratically with the size of the input as the number of potential relation arguments in the input grows (e.g. it is possible for a time expression to contain multiple events). Because of this, the CONTAINERS input/output variant had a problem on the output side: we observed that the seq2seq maximum length limit (512 word pieces) was not enough to accomodate all relation instances for chunk sizes above 75-100 word pieces. Our 1-CONTAINER input/output variant mitigates that problem by essentially trading the output size for a larger number of training examples, resulting in the best performance (line 12). However, the 1-CONTAINER variant (line 11) is still better when we set the chunk size to the same value as the best CONTAINERS variant (line 10). This hints at a fundamental advantage of this type of model over a full seq2seq model. We hypothesize that this is due to a difficulty on the part of seq2seq models to produce structured outputs such as predicate logic statements.

4 Conclusion

Engineering input/output representations for seq2seq models proved difficult as obvious choices of output representations, such as explicit relations encoded as predicate logic statements led to poor performance. By exploring alternative input/output representations, we were able to improve performance. Our 1-CONTAINER input/output variant with a T5 model was competitive with or better than the current state-of-the-art without requiring additional training data. This is likely due to several factors. First, predicting one relation at a time allowed the model to mitigate the limitation on the maximum length of the output and capture long-distance relations, which was more challenging for the other variants. Second, it required generating only a single word, which is more like the text generation tasks the seq2seq models were trained on than generating predicate logic expressions like the other variants required. Future research may want to explore different pretraining objectives for seq2seq models that would be more appropriate when downstream tasks require generating structured output.

Acknowledgements

Research reported in this publication was supported by the National Library Of Medicine of the National Institutes of Health under Award Numbers R01LM012973 and R01LM010090. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California. Association for Computational Linguistics.

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

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

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