Representations of Time Expressions for Temporal Relation Extraction with Convolutional Neural Networks

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

[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

Token sequences are often used as the input for Convolutional Neural Networks (CNNs) in natural language processing. However, they might not be an ideal representation for time expressions, which are long, highly varied, and semantically complex. We describe a method for representing time expressions with single pseudo-tokens for CNNs. With this method, we establish a new state-of-the-art result for a clinical temporal relation extraction task.

1 Introduction

Convolutional Neural Networks (CNNs) utilize convolving filters and pooling layers for exploring and subsampling a feature space, and show excellent results in tasks such as semantic parsing (Yih et al., 2014), search query retrieval (Shen et al., 2014), sentence modeling (Kalchbrenner et al., 2014), and many other natural language processing (NLP) tasks (Collobert et al., 2011).

Token sequences are often used as the input for a CNN model in NLP. Each token is represented as a vector. Such vectors could be either word embeddings trained on the fly (Kalchbrenner et al., 2014), pre-trained on a corpus (Pennington et al., 2014; Mikolov et al., 2013), or one-hot vectors that index the token into a vocabulary (Johnson and Zhang, 2014). CNN filters then act as n-grams over continuous representations. Subsequent network layers learn to combine these n-gram filters to detect patterns in the input sequence.

This token vector sequence representation has worked for many NLP tasks, but has not been well-studied for temporal relation extraction. Time expressions are complex linguistic expressions that are challenging to represent because of their length and variety. For example, for the time expressions in the THYME (Styler IV et al., 2014) colon cancer training corpus, there are 3,833 occurrences of 2,014 unique expressions of which 1,624 (80.6%) are multi-token, 1,104 span three or more tokens, and some span as many as 10 tokens. CNNs, which represent meaning through fragments of word sequences, might struggle to compose these fragments to represent the meaning of time expressions. For example, can a CNN properly generalize that *May 7* as a date is closer to *April 30* than *May 20*? Can it embed years like *2012* and *2040* to recognize that the former was in the past, while the latter is in the future? Time normalization systems can handle such phenomena, but they are complex and language-specific, and often require significant manual effort to re-engineer for a new domain (Strötgen and Gertz, 2013; Bethard, 2013).

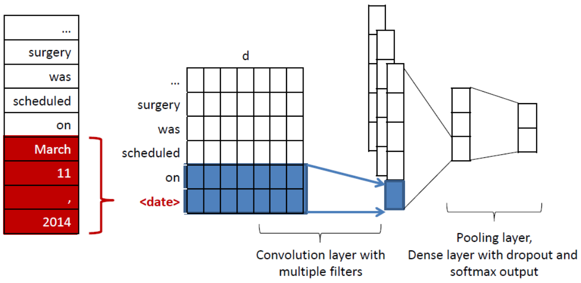
...
surgery
was
scheduled
on
March
11
,
2014
d
...
surgery
was
scheduled
on
<date>
Convolution layer with multiple filters
Pooling layer, Dense layer with dropout and softmax output

Figure 1: CNN with encoded timex

Fortunately, not all tasks require full time normalization, so if the CNN can at least embed a meaningful subset of the time expression semantics, it may still be helpful in such tasks. An open question then, is how to best feed time expressions to the CNN so that it can usefully generalize over them as part of its solution to a larger task.

We propose representing time expressions as single pseudo-tokens, with single vector representations (as in Figure 1), that encode easily extractable information about the time expression that is valuable for the task of temporal relation extraction. The benefits are two-fold: 1) Only minimal linguistic preprocessing is required: off-the-shelf time expression identifiers are available with low over-

EVENT TIME
A *surgery* *is scheduled on* *Mar 11*.
⇓
1: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ mar 11 ⟨/t⟩ .
2: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ ⟨timex⟩ ⟨/t⟩ .
3: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ ⟨date⟩ ⟨/t⟩ .
4: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ ⟨nn_cd⟩ ⟨/t⟩ .
5: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ ⟨date_nn_cd⟩ ⟨/t⟩ .
6: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ ⟨index_721⟩ ⟨/t⟩ .
7: a ⟨e⟩ surgery ⟨/e⟩ is scheduled on ⟨t⟩ mar 11 ⟨date⟩ ⟨/t⟩ .
8: ⟨o⟩ ⟨o⟩ ⟨o⟩ ⟨o⟩ ⟨o⟩ ⟨b⟩ ⟨i⟩ ⟨o⟩ ⟨o⟩
9: ⟨o⟩ ⟨o⟩ ⟨o⟩ ⟨o⟩ ⟨o⟩ ⟨b_date⟩ ⟨i_date⟩ ⟨o⟩ ⟨o⟩
10: a ⟨e1⟩ surgery ⟨/e1⟩ is ⟨e2⟩ scheduled ⟨/e2⟩ on .

Figure 2: Representations of an input sequence

head and high accuracy (Miller et al., 2015). 2) CNN filters are more effective because they operate over the time expression as one unit. The filter process can thus focus on the informative surrounding context to catch generalizable patterns instead of being trapped within lengthy time expressions.

We explored a variety of one-tag representations for time expressions, from very specific to very general. We also experimented with other ways to inject temporal information into the CNN models and compared them with our one-tag representations. We picked a challenging learning task where time expressions are critical cues for evaluating our proposed representation: clinical temporal relation extraction. The identification of temporal relations in medical text has been drawing growing attention because of its potential to dramatically increase the understanding of many medical phenomena such as disease progression, longitudinal effects of medications, a patient's clinical course, and its many clinical applications such as question answering (Das and Musen, 1995; Kahn et al., 1990), clinical outcomes prediction (Schmidt et al., 2005), and the recognition of temporal patterns and timelines (Zhou and Hripcsak, 2007; Lin et al., 2014).

Through experiments, we not only demonstrate the usefulness of one-tag representations for time expressions, but also establish a new state-of-the-art result for clinical temporal relation extraction.

2 Methods

We trained two CNN-based classifiers for recognizing two types of within-sentence temporal relations, event-event and event-time relations, as they usually call for different temporal cues (Lin et al., 2016a). The input to our classifiers was manually annotated (gold) events and time expressions during both training and testing stages. That way we isolated the task of time expression representation for temporal relation extraction from the tasks of event and time expression recognition. We adopted the same xml-tag marked-up token sequence representation and model setup as (Dligach et al., 2017). Figure 2(1) illustrates the marked-up token sequence for an event-time instance, in which the event is marked by ⟨e⟩ and ⟨/e⟩ and the time expression is marked by ⟨t⟩ and ⟨/t⟩. Event-event instances are handled similarly, e.g. *a ⟨e1⟩ surgery ⟨/e1⟩ is ⟨e2⟩ scheduled ⟨/e2⟩ on march 11.*

We tried different ways of representing a time expression as a one-token tag. The most coarse option would be to represent all time expressions with one universal tag, ⟨timex⟩, as in Figure 2(2). For more granular options, we experimented with these additional representations: 1) The time class[1] of a time expression, as in Figure 2(3), where the time expression, *Mar 11*, is represented by its class, ⟨date⟩. 2) The Penn Treebank POS tags of the tokens in a time expression, as in Figure 2(4), where the time expression, *Mar 11*, is represented by concatenating two POS tags, ⟨nn_cd⟩. 3) The combination of time class and POS tags, as in Figure 2(5), where the time expression is represented by ⟨date_nn_cd⟩. 4) A fine-grained representation that assigns an index to each unique time expression, as in Figure 2(6), where the time expression is represented by ⟨index_721⟩, the index used every time the time expression *Mar 11* appears. For event-event relations, where time expressions are not part of the relational arguments, we tried removing the time expressions altogether, as in Figure 2(10), where *Mar 11* has been removed.

To show the contribution of one-tag representations versus adding new information to the system, we explored incorporating temporal information by adding time-class tags to the original token sequences (Figure 2(7)) and adding BIO tags with/without time classes for time expression (Figure 2(8,9)) alongside the original token sequences.

We used the same CNN architecture as the CNN used in (Dligach et al., 2017), and focused on extracting the *contains* relation. The word embeddings were randomly initialized[2] and

[1] We used the standard clinical domain classification (Styler IV et al., 2014), where the classes are date (e.g., *next Friday*, *this month*), time (e.g. *3:00 pm*), duration (e.g., *five years*), quantifier (e.g. *twice*, *four times*), prepostexp (e.g., *preoperative*, *post-surgery*), and set (e.g., *twice monthly*).

[2] Our preliminary experiments showed better results for randomly-initialized embeddings than several pre-trained embeddings. One-hot vectors were too slow for processing.

Model	Event-time relations			Event-event relations		
	P	R	F1	P	R	F1
THYME system	0.583	0.810	0.678	0.569	0.574	**0.572**
1. CNN tokens	0.660	0.775	0.713	0.566	0.522	0.543
2. CNN <timex>	0.697	0.710	0.703	0.681	0.397	0.501
3. CNN time class tags	0.705	0.759	**0.731**	0.582	0.495	0.535
4. CNN POS tags	0.727	0.710	0.719	0.619	0.462	0.529
5. CNN time class+ POS tags	0.709	0.747	0.727	0.553	0.521	0.537
6. CNN indexed time expressions	0.692	0.727	0.709	0.645	0.429	0.516
7. CNN token + time class tags	0.749	0.626	0.682	0.437	0.589	0.502
8. CNN token + BIO tags	0.691	0.708	0.700	0.570	0.423	0.486
9. CNN token + BIO-time class tags	0.713	0.726	0.719	0.428	0.542	0.478
10. CNN remove all time expressions	n/a	n/a	n/a	0.635	0.446	0.524

Table 1: Event-time and event-event *contains* relation on the dev set (all notes included)

learned through training. For the combined token and BIO sequence input, we used two embedding/convolutional branches: one for the token sequence, and one for the BIO sequence; the resulting vectors were concatenated into the same dense, dropout and final softmax layers. All models were implemented in Keras 1.0.4 (Chollet, 2015) with Theano (Theano Development Team, 2016) backend. Models were trained with a batch size of 50, a dropout rate of 0.25, RMSprop optimizer, and a learning rate of 0.0001, on a GTX Titan X GPU. Our code will be made publicly available.

3 Evaluation Methodology and Results

We tested our new representations of time expressions on the THYME corpus (Styler IV et al., 2014). We followed the evaluation setup of Clinical TempEval 2016 (Bethard et al., 2016). The THYME corpus contains a colon cancer set and a brain cancer set. The colon cancer set was our main focus. Models were trained on the colon cancer training set, hyper-parameters were tuned on the colon cancer development set. Finally, the best models were re-trained using the best hyper-parameters on the combined training and development sets, tested and compared on the colon cancer test set.

As a secondary validation set, we also considered the brain cancer portion of the THYME corpus. The models were re-trained on the brain cancer training and development sets (using the best hyper-parameters found for colon cancer) and tested on the brain cancer test set.

For results on the test sets, we used the official Clinical TempEval evaluation scripts (with closure-enhanced precision, recall, and F1-score).

Table 1 shows performance on the colon development set for the THYME system and the various methods of representing time expressions to CNN models. The order of representation settings is identical to that in Figure 2. For event-time relations, all our neural models outperformed the state-of-the-art THYME system's F1. Three one-tag temporal representations with moderate granularity, time class (Table 1(3)), POS tags (Table 1(4)), and time class plus POS tags (Table 1(5)), performed better than the token sequence CNN baseline (Table 1(1)), with the time class tag representation achieving the highest score (Table 1(3)). CNNs were better able to leverage time class information in our tag-based representation (Table 1(3)), than adding time class information to the original token sequence (Table 1(7)) or adding a separate time-class neural embedding (Table 1(9)).

For event-event relations, none of the neural models performed as well as the state-of-the-art THYME system. The CNN token-based model had similar performance as some of the one-tag temporal representations (Table 1(3,4,5)). Removing the time expression entirely (Table 1(10)) did not hurt performance much, confirming that time expressions were not critical cues for within-sentence event-event relation reasoning (Xu et al., 2013). Thus, on the colon test set, we evaluated the contribution of encoding time expressions on the event-time CNN model only. For the event-event part, we used the THYME event-event system, so that our results were directly comparable with the outcomes of Clinical TempEval 2016 (Bethard et al., 2016) and the performance of the THYME system (Lin et al., 2016a,b). As for the Brain cancer data, we

Corpus	Model	*contains* relations			
		P	R	F1	p-value
Colon cancer	Top Clinical TempEval 2016 system	0.588	0.559	0.573	
	THYME system	0.669	0.534	0.594	
	CNN (tokens) event-time + THYME event-event	0.654	0.576	0.612	
	CNN (encode) event-time + THYME event-event	0.662	0.585	**0.621**	0.03
Brain cancer	CNN (tokens) event-time	0.765	0.371	0.500	
	CNN (encode) event-time	0.726	0.429	**0.539**	0.0002

Table 2: Performance on both Colon and Brain test sets with the Clinical TempEval evaluation.

only evaluated on the event-time CNN models, so that we could directly assess the contribution of encoding time expressions as time class tags.

The top 4 rows of Table 2 show performance on the colon cancer test set for the best model from Clinical TempEval 2016, the THYME system, our CNN model with tokens only, and our CNN model where time expressions are encoded with time class tags. (To allow comparison with prior work, the event-time relation predictions made by our CNN models were coupled with the event-event relation predictions from the THYME system.) The bottom two rows of Table 2 show performance on the brain cancer test set. On both colon and brain corpora, the encoded CNN model outperformed the regular CNN model significantly, based on a Wilcoxon signed-rank test over document-by-document comparisons, as in (Cherry et al., 2013).

4 Discussion

The CNN filters in the first layers are designed to detect the presence of highly discriminative patterns. For the event-time relation extraction task, one such pattern signaling a *contains* relation is "*on Mar 11, 2014*" as in Figure 1. However, a more generalizable pattern should be – "*on DATE*". Our time-class tag representation provided such information and contributed towards generalizability. A size-two filter can easily capture such a useful pattern, instead of picking up less generalizable patterns like "on March" or "11 ," (shown in Figure 1). For a time-sensitive learning task, especially the event-time relation extraction, our time encoding technique has been proved effective on two corpora. We hypothesize the contribution is from generalizability and efficient filter computation.

Our method did not work for event-event relations because time expressions are not critical cues for such relations. CNN models as a whole did not outperform the conventional THYME event-event system, as confirmed by Dligach et al. (2017). Event-event relations have lower inter-annotator agreement and usually leverage more of the syntactic information and event properties (Xu et al., 2013), which are not perfectly captured by token sequences. The class imbalance issues are more severe for event-event relations than for event-time relations as well (Dligach et al., 2017). These likely lead to a lower performance for event-event CNNs. In the future, we will investigate methods to improve the event-event model including incorporating syntactic information and event properties into a deep neural framework, and positive instance augmentation Yu and Jiang (2016).

Word embeddings trained by conventional methods such as word2vec and GloVe did not prove to be useful in our preliminary experiments. This is likely due to (1) lack of sufficiently large publicly available domain-specific corpora, and (2) inability of the conventional methods to capture the semantic properties of events that are key for the relation extraction task (such as event durations).

Currently, when we combined our encoded CNN-based event-time model with the THYME event-event model, we achieved the state-of-the-art performance (0.621F) on the colon cancer data. The best 2016 Clinical TempEval system achieved 0.573F (Bethard et al. (2016); row 1 of Table 2), the result of the THYME system was 0.594F (Lin et al. (2016b); row 2 of Table 2), while our best combined model reached 0.621F, significantly higher (p=0.03) than the 0.612F of the combination of a regular CNN event-time model and the THYME event-event model. Note that the number of gold event-time *contains* relation instances is similar to the number of gold event-event *contains* relations (Lin et al., 2016a). Having a better event-time model indeed made the difference.

The conventional machine learning world has focused on heavy feature engineering, while the new deep learning world has called for minimalistic pre-processing as input to powerful learners. We propose a new direction to combine the best of both worlds – infusing some knowledge into the learner input. For CNN models, multi-word time expressions are imperfectly represented in the token sequence representation. With a little engineering, we can encapsulate the time expressions in one tag with different granularities. Our experiments show that this small change still takes minimum linguistic preprocessing but delivers a significant performance boost for a temporal relation extraction task. There are other multi-token named entities (locations, organizations, etc.) where it may be hard to generalize over their multiple tokens. We believe our encoding strategy is likely to benefit tasks where critical linguistic information resides in phrases or multi-word units.

Acknowledgments

The study was funded by R01LM10090 (THYME), R01GM103859 (iPGx), and U24CA184407 (DeepPhe). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The Titan X GPU used for this research was donated by the NVIDIA Corporation.

References

Steven Bethard. 2013. A synchronous context free grammar for time normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 821–826. http://www.aclweb.org/anthology/D13-1078.

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.

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. https://doi.org/10.1136/amiajnl-2013-001624.

François Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Amar K Das and Mark A Musen. 1995. A comparison of the temporal expressiveness of three database query methods. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, page 331.

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

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058* .

Michael G Kahn, Larry M Fagan, and Samson Tu. 1990. Extensions to the time-oriented database model to support temporal reasoning in medical expert systems. *Methods of information in medicine* 30(1):4–14.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* .

Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K Savova. 2016a. Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association* 23(2):387–395.

Chen Lin, Elizabeth W Karlson, Dmitriy Dligach, Monica P Ramirez, Timothy A Miller, Huan Mo, Natalie S Braggs, Andrew Cagan, Vivian Gainer, Joshua C Denny, and Guergana K Savova. 2014. Automatic identification of methotrexate-induced liver toxicity in patients with rheumatoid arthritis from the electronic medical record. *Journal of the American Medical Informatics Association* https://doi.org/10.1136/amiajnl-2014-002642.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016b. Improving temporal relation extraction with training instance augmentation. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. Association for Computational Linguistics, pages 108–113.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Timothy A Miller, Steven Bethard, Dmitriy Dligach, Chen Lin, and Guergana K Savova. 2015. Extracting time expressions from clinical text. In *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015)*. Association for Computational Linguistics, pages 81–91.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–43.

Reinhold Schmidt, Stefan Ropele, Christian Enzinger, Katja Petrovic, Stephen Smith, Helena Schmidt, Paul M Matthews, and Franz Fazekas. 2005. White matter lesion progression, brain atrophy, and cognitive decline: the austrian stroke prevention study. *Annals of neurology* 58(4):610–616.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, pages 373–374.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation* 47(2):269–298. https://doi.org/10.1007/s10579-012-9179-y.

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.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688. http://arxiv.org/abs/1605.02688.

Yan Xu, Yining Wang, Tianren Liu, Junichi Tsujii, I Eric, and Chao Chang. 2013. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association* 20(5):849–858.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *ACL (2)*. Citeseer, pages 643–648.

Jianfei Yu and Jing Jiang. 2016. Pairwise relation classification with mirror instances and a combined convolutional neural network. *Proceedings of COLING 2016* .

Li Zhou and George Hripcsak. 2007. Temporal reasoning with medical dataa review with emphasis on medical natural language processing. *Journal of biomedical informatics* 40(2):183–202.