Neural Temporal Relation Extraction

Dmitriy Dligach[1], Timothy Miller[2], Chen Lin[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]ddligach@luc.edu
[2]{first.last}@childrens.harvard.edu
[3]bethard@email.arizona.edu

Abstract

We experiment with neural architectures for temporal relation extraction and establish a new state-of-the-art for several scenarios. We find that neural models with only tokens as input outperform state-of-the-art hand-engineered feature-based models, that convolutional neural networks outperform LSTM models, and that encoding relation arguments with XML tags outperforms a traditional position-based encoding.

1 Introduction

Investigating drug adverse effects, disease progression, and clinical outcomes is inconceivable without forming some representation of the temporal structure of electronic health records. Temporal relation extraction has emerged as the most viable route to building timelines that tie each medical event to the time of its occurrence. This connection between times and events can be captured as a *contains* relation which is the most frequent temporal relation type in clinical data (Styler IV et al., 2014). Consider the sentence: *Patient was diagnosed with a rectal cancer in May of 2010*. It can be said that the temporal expression *May of 2010* in this sentence *contains* the *cancer* event. The same relation can exist between two events: *During the surgery the patient experienced severe tachycardia*. Here, the *surgery* event *contains* the *tachycardia* event.

The vast majority of systems in temporal information extraction challenges, such as the i2b2 (Sun et al., 2013) and Clinical TempEval tasks (Bethard et al., 2015; Bethard et al., 2016), used classifiers with a large number of manually engineered features. This is not ideal, as most NLP components used for feature extraction experience a significant accuracy drop when applied to out-of-domain data (Wu et al., 2014; McClosky et al., 2010; Daumé III, 2009; Blitzer et al., 2006), propagating the error to the downstream components and ultimately leading to significant performance degradation. In this work, we propose a novel temporal relation extraction framework that requires minimal linguistic pre-processing and can operate on raw tokens.

We experiment with two neural architectures for temporal relation extraction: a convolutional neural network (CNN) (LeCun et al., 1998) and a long short-term memory neural network (LSTM) (Hochreiter and Schmidhuber, 1997). Little work exists on using these methods for relation extraction; to the best of our knowledge no work exists on using LSTM models for relation extraction or CNN models for *temporal* information extraction. Zeng et al. (2014) and Nguyen and Grishman (2015) employ CNNs for non-temporal relation extraction and show that CNNs can be effective for relation classification and perform as well as token-based baselines for relation extraction. Our experiments, on the other hand, show that neural relation extraction models can compete with a complex feature-based state-of-the-art relation extraction system.

Another important difference that sets our work apart is our representation of the argument positions: previous work used token position features (embedded in a 50-dimensional space) to encode the relative distance of the words in the sentence to the relation arguments (Nguyen and Grishman, 2015; Zeng et al., 2014). We propose a much simpler method for encoding relation argument positions and show that it works better in our experiments. We introduce special tokens (e.g. `<e1>` and `</e1>`) to mark the positions of the arguments in a sentence, effectively annotating the relation arguments with XML tags. The sentences augmented with this markup become the input to a neural network. This approach makes it possible to use the same representations for CNN and LSTM models.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 746–751,
Valencia, Spain, April 3-7, 2017.

Our contributions are the following: we introduce a simple method for encoding relation argument positions and show that CNNs and LSTMs can be successfully used for temporal relation extraction, establishing a new state-of-the-art result. Our best performing model has no input other than word tokens, in contrast to previous state-of-the-art systems that require elaborate linguistic preprocessing and many hand-engineered features. Finally, we show that a neural model can be remarkably effective at extracting temporal relations when provided with only part-of-speech tags of words, rather than words themselves. This approach is promising for the scenarios where reliance on word tokens is undesirable (e.g. domain adaptation).

2 Methods

2.1 Input representation

All proposed models operate on a $n \times d$ matrix representing the context of a temporal relation. This matrix is formed by concatenating n word embeddings of d dimensions. Word embeddings can either be initialized randomly or use the output of a tool like word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Similar representations have been used for various sentence modeling tasks (Kim, 2014; Kalchbrenner et al., 2014).

We adapt this input representation for relation extraction by augmenting the input token sequences with markup of the relation arguments. For example, the markup *Patient was* `<e>` *diagnosed* `</e>` *with a rectal cancer in* `<t>` *may of 2010* `</t>` indicates that the model is to predict a relation between the event *diagnosis* and the time *May of 2010*. Event-event relations are handled similarly, e.g.: *During the* `<e1>` *surgery* `</e1>` *the patient experienced severe* `<e1>` *tachycardia* `</e2>`.

The directionality of the temporal relation is modeled as a three-way classification task: *contains* vs. *contains*$^{-1}$ vs. *none*. For event-time relations, *contains* indicates that the time contains the event, and *contains*$^{-1}$ indicates the reverse. For event-event relations, *contains* indicates that the first event in the text *contains* the second event, and *contains*$^{-1}$ indicates the reverse. For both event-event and event-time relations, *none* indicates that no relation exists between the arguments.

In addition to training on token sequences, we experiment with training on sequences of part-of-speech (POS) tags. Under this scenario, the input to the network is again an $n \times d$ matrix, but it now embeds the POS tags in the d dimensional space.

2.2 Models

We experiment with two neural architectures for temporal relation extraction: (1) a convolutional neural network (CNN), and (2) a long short-term memory neural network (LSTM). Both models start by feeding the input word sequences into an embedding layer, which we configure to learn the embeddings from scratch. In the CNN-based model, the embedding layer is followed by a convolution layer that applies convolving filters of various sizes to extract n-gram-like features, which are then pooled by a max-pooling layer. In the LSTM-based model, the embedding layer is fed into a standard LSTM recurrent layer. The output of either the max-pooling layer (for the CNN) or the last unit in the recurrent layer (for the LSTM) is fed into a fully connected dense layer, which is followed by the final softmax layer outputting the probability distribution over the three possible classes for the input.

We build a separate model for event-time and event-event relations, and for each model we try several input variants: token sequences, POS sequences, and token/POS sequence combination. The latter model involves building two separate neural network branches: the first receives tokens as features, while the second receives POS tags; the two branches are merged and fed into the softmax layer, acting in effect as an ensemble classifier.

3 Evaluation

3.1 Datasets

We evaluated the proposed methods on a publicly available clinical corpus (Styler IV et al., 2014) that was the basis for the Clinical TempEval shared tasks (Bethard et al., 2015; Bethard et al., 2016). The gold standard annotations include time expressions, events (both medical and general), and temporal relations. We used the standard split established by Clinical TempEval 2016, using the development set for evaluating models and tuning model parameters, and evaluating our best event-event and event-time models on the test set. Following Clinical TempEval, we focus only on the *contains* relation, which was the most common relation and had the highest inter-annotator agreement.

3.2 Experiments

We compare the performance of our neural models to the *THYME* system (Lin et al., 2016a),

Model	Argument representation	Event-time relations			Event-event relations		
		P	R	F1	P	R	F1
THYME full system	n/a	0.583	0.810	0.678	0.569	0.574	0.572
THYME tokens only	n/a	0.564	0.786	0.657	0.562	0.539	0.550
CNN tokens	position embeddings	0.647	0.627	0.637	0.580	0.324	0.416
CNN tokens	XML tags	0.660	0.775	0.713	0.566	0.522	0.543
CNN pos tags	XML tags	0.707	0.708	0.707	0.630	0.204	0.309
LSTM tokens	XML tags	0.691	0.626	0.657	0.610	0.418	0.496
LSTM pos tags	XML tags	0.754	0.657	0.702	0.603	0.212	0.313
CNN token + pos tags	XML tags	0.727	0.681	0.703	0.653	0.435	0.522
LSTM token + pos tags	XML tags	0.698	0.660	0.679	0.572	0.458	0.508

Table 1: Event-time and event-event *contains* relation on dev set.

Model	Event-time relations			Event-event relations		
	P	R	F1	P	R	F1
THYME system	0.244	0.819	0.376	0.206	0.681	0.317
CNN tokens	0.268	0.768	0.398	0.309	0.538	0.393

Table 2: Event-time and event-event *contains* relations with medical arguments on dev set

which is based on hand-engineered linguistic features and support vector machine classifiers, and achieved the highest performance on the Clinical TempEval 2015 test set (Lin et al., 2016b). This system is available as part of cTAKES (`http://ctakes.apache.org`) and performs both event-event and event-time relation classification. We discard all non-*contains* relation instances from the data, re-train this system, and re-evaluate it on the official Clinical TempEval 2016 dev and test sets.

We train two versions of the the THYME system: (1) a version based on the full set of features including token features, dependency path features, ontology (UMLS) based features, gold event and time properties, and others; (2) token only features. Our neural models include CNN and LSTM architectures trained on sequences of tokens, sequences of POS tags, and a combination of the two. For comparison, we also include a token-based CNN model that uses position embeddings (Nguyen and Grishman, 2015; Zeng et al., 2014) rather than XML markup used in the rest of our neural models.

SemEval data includes gold annotations of both medical (e.g. *colonoscopy*, *tachycardia*) and general (e.g. *discussed*, *reported*) events. Relations between medical events are the most important for clinical applications, but also present a special challenge as the accuracy of their extraction is currently low. To evaluate our models on the relations between clinical events, we filtered out all general events (and relations associated with them) using a UMLS dictionary. UMLS (Bodenreider, 2004) is a comprehensive ontology of clinical terminology (somewhat analogous to WordNet (Miller, 1995)) that includes most clinical terms and thus can be used as a lookup resource for clinical vocabulary. Similar evaluation was used in (Lin et al., 2016b).

We implemented all neural models in Keras 1.0.4 (Chollet, 2015) with the Theano (Theano Development Team, 2016) backend. The code will be made publicly available. All models were trained with batch size of 50, dense layer dropout rate of 0.25, and RMSprop optimizer. The words were represented using 300-dimensional embeddings initialized randomly. The training was performed using GeForce GTX Titan X GPU provided by NVIDIA Corporation.

The CNN models used 200 filters each for filter sizes 2, 3, 4, and 5, and a learning rate of 0.0001. The LSTM models had 128 hidden units and a learning rate of 0.001. The number of hidden fully connected units was 300.

These settings are identical or similar to those used in neural sentence modeling work (Nguyen and Grishman, 2015; Zhang and Wallace, 2015; Kim, 2014) and were validated on the SemEval development set. We tuned the number of training epochs by starting from 3 and increasing until validation accuracy began to decrease. Once the parameter tuning was finalized, we evaluated our best event-event and event-time models on the held-out test set.

Model	Event-time relations			Event-event relations		
	P	R	F1	P	R	F1
THYME system (all events)	0.577	0.845	0.685	0.595	0.572	0.584
CNN tokens (all events)	0.683	0.717	0.700	0.688	0.412	0.515
THYME system (medical events only)	0.230	0.851	0.362	0.215	0.703	0.330
CNN tokens (medical events only)	0.272	0.714	0.394	0.300	0.519	0.380

Table 3: Event-time and event-event *contains* relations on test set

3.3 Results

Table 1 shows the evaluation of different model types and feature sets on the dev set. For both event-time and event-event relations, the best-performing neural model was the CNN with only tokens as features. For event-time relations, all our neural models except the token-based LSTM outperformed the state-of-the-art THYME system, and all models performed as well or better than the THYME tokens-only baseline. For event-event relations, none of the neural models performed as well as the state-of-the-art THYME system, and only the CNN token-based model came close to the performance of the THYME tokens-only baseline. The CNN with position embeddings (CNN tokens / position embeddings row) performed worse then when arguments were marked with XML tags (CNN tokens / XML tags row). CNNs with position embeddings have considerably more parameters and are harder to train; this likely explains the performance drop comparing to the models where the arguments are marked with XML tags.

Table 2 shows the performance of the THYME system and our best neural model (CNN tokens with XML tags) on the modified data that only contains relations between medical events. The neural models outperform the feature-based system in both cases.

Finally, Table 3 shows the performance of the state-of-the-art THYME system and the best neural systems on the test set. For event-time relation extraction, our neural models establish a new state-of-the art, and when focusing on only medical events our neural models outperform the state-of-the-art on both event-time and event-event relations.

4 Discussion

Of all the neural architectures we experimented with, the token-based CNN demonstrated the best performance across all experimental conditions. And in all scenarios but one (event-event relations, all events), this model with only token input outperformed the feature-based THYME system which includes not only tokens and part of speech tags, but syntactic tree features and gold event and time properties. Intriguingly, for event-time relations, the part-of-speech-based CNN also outperformed the feature-based THYME system (and was very close to the performance of the token-based CNN), suggesting that part-of-speech alone is enough to make accurate predictions in this task, when coupled with the modeling power of a neural network.

We also found that CNN models outperformed LSTM models for our relation extraction tasks, despite the intuition that LSTMs, by modeling the entire word sequence, should be a better model of natural language data. In practice, the local predictors of class membership obtained by the CNN seem to provide stronger cues to the classifier than the vectorized representation of the entire sequence formed by the LSTM.

Despite the structural similarities between event-time relation classification and event-event relation classification, the neural models fell short of traditional feature-based models for event-event relations, reaching only up to the level of a traditional feature-based model that has access only to the tokens (the same input as the neural models). This suggests that the neural models for event-event relations are not able to generalize over the token input as well as they were for event-time relations. This may be due in part to the difficulty of the task: even for feature-based models, event-event classification performance is about 10 points lower than event-time classification performance. But it may also be due to class imbalance issues, as there are many more *none* relations in the event-event task: the positive to negative ratio is 1:15 for event-event, but only 1:3 for event-time. The THYME system for event-event relations is tuned with class-specific weights that help it deal with class imbalance, and without these class-specific weights, its performance drops more than 10 points in F1. Our neural models do not yet include any equivalent

for addressing class imbalance, so this may be a source of the problem. The fact that the event-event CNN system beats the feature-based system when tested on only medical events supports this view: after non-medical events are removed from the sentence, the imbalance problem is alleviated (a medical event is more likely to be involved in a relation), which likely allows the CNN model to generalize better. Addressing this class imbalance problem is an interesting avenue for future work. Additionally, we plan to investigate the applicability of the proposed neural models for general (non-temporal) relation extraction.

Acknowledgments

This work was partially funded by the US National Institutes of Health (U24CA184407; R01 LM 10090; R01GM114355). The Titan X GPU used for this research was donated by the NVIDIA Corporation.

References

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.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

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

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

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, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016b. Improving temporal relation extraction with training instance augmentation. *ACL 2016*, page 108.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 28–36. Association for Computational Linguistics.

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.

George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 39–48.

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

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.

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

Stephen Wu, Timothy Miller, James Masanz, Matt Coarr, Scott Halgrim, David Carrell, and Cheryl Clark. 2014. Negation's not solved: generalizability versus optimizability in clinical natural language processing. *PloS one*, 9(11):e112774.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.