An Improved Neural Baseline for Temporal Relation Extraction

Group no : 1
Project no : 4

Varun Gupta
18MA20050

Koushiki Dasgupta Chaudhuri
18MA20020

Altaf Ahmad
18MA20005

C Shanmukh Siva Sai
20EC10022

Indian Institute of Technology
Kharagpur

November 2021

Contents

1. Task Definition

Temporal relation extraction refers to the process of determining temporal relations between events, that is, whether a particular event happened before or after another one. It is one of the more difficult tasks in natural language understanding, mainly due to the unavailability of high quality training data. In 2018, [7] Ning et al. introduced a new dataset - Multi-Axis Temporal Relations for Start-points (MATRES) which had much higher annotation quality compared to previous datasets employed for TempRel extraction. In this study, we extend the work of [5] to build an improved LSTM based model for TempRel extraction which we train on the MATRES dataset.

1.1 Case Study

We first look into the different types of possible solutions and previous iterations to attempt to solve the problem at hand. The earlier attempts were based on creating models using classic learning algorithms like support vector machines, logistic regression etc. using the features which were hand-engineered for each pair of events [1]. Further approaches included mapping events into a universal timeline representation by inferring the relative temporal relation amongst events and associating each event with an absolute time interval [2]. After this further work was done for evaluating the time expressions, events and relations in contrast to gold entities in the time frame which had absolute time known for them [9]. Apart from that there were other methods which had moderate improvements over feature based methods. Some of them are - A Context-aware neural network model which stores processed temporal relations in the narrative order and retrieves them for use when the required entities are encountered. [4]; Extraction of linear time-line from a set of extracted temporal relations which directly predicts the start and end points of the events from the given text, which constitutes a linear time-line without going through the intermediate step of prediction of temporal relations [3].
Another notion of multi-axis was introduced [8] to represent the temporal structure of the text, and the identification of the fact that one of the sources of confusion was asking human annotators for TempRels across multiple axes.

1.2 Description of the Datasets

Temporal Relation (TempRel) extraction has been an important task for event understanding and timeline creation. In the dataset, first a multi-axis modeling is introduced to represent the temporal structure of events, based on which the events are anchored to different semantic axes. These event pairs represent different semantic phenomena and belong to different axes. After that the event pairs are represented using two time intervals, $[t_{start}\ ,\ t_{end}]$. The comparisons involving end points are typically more difficult than those comparing the start points.

1.2.1 Temporal Structure of Events

AN important question to solve while looking at a set of events would be to find the pair of events which are related to each other. An ideal annotator may figure out the ambiguity by themselves but it is not feasible for

requirement. So, in practice, one axis is annotated at a time : first the event is classified as anchorable or not and then every pair of anchorable events are annotated.

1.2.2 Axis Projection

In the multi-axis event modeling, the strategy chosen by the TB-Dense falls into a general approach, which means projecting events across different axes to handle the in-comparability between any two axes. Another prominent difference to earlier work was the introduction of orthogonal axes which can be used to bridge the events which have intersecting axes.

1.2.3 Interval Splitting

Instead of reducing relations between two intervals, the two time points are explicitly compared. IN this way the label set is simply $before$, $after$ and $equal$, while the expressivity remains the same. This also provides more information when the relation between the events is vague.

1.2.4 Annotation Scheme Design

Summarizing, the annotation scheme is a two step process consisting of multi-axis modeling and interval splitting. In order to handle vague relations, a majority agreement rule is applied as a post-processing step to back off a decision to vague when annotators cannot pass a majority vote.
The final dataset created is called MATRES, for Multi-Axis Temporal Relations for Start-points. This is the dataset which we will be using for the given task at hand.
In this case, the temporal relation is the abstraction of the two entities or events with the information of the

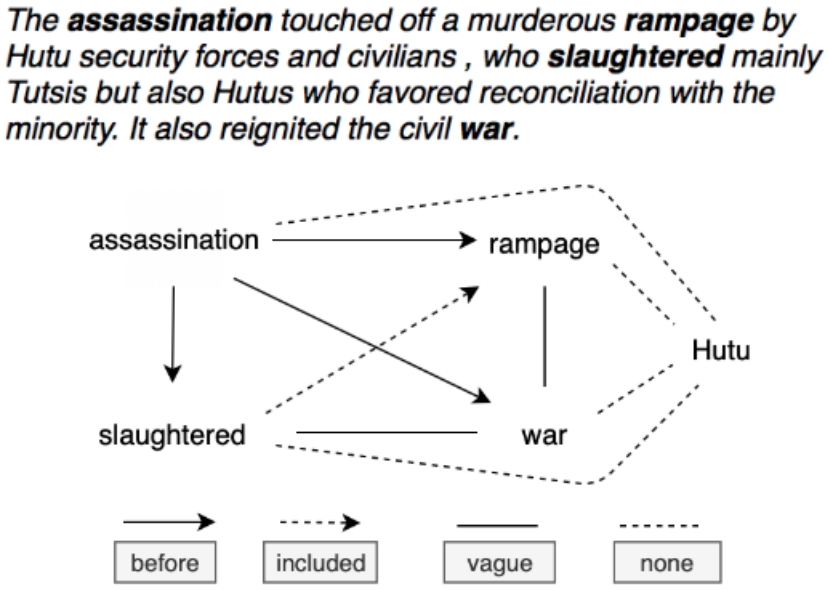

Figure 1.1: Creation of the temporal relation graph from the given paragraph

antecedent and the chronology of the events at hand.

2. Methodology

We employ LSTM networks for the temporal relation extraction task. A sequence of word embeddings is provided as an input to the LSTM. The position of events are known beforehand. The hidden states from both time steps that correspond to the location of those events are concatenated by the LSTM layer.
A Siamese network is used to fit an updated version of TEMPROB (a TempRel knowledge base extracted previously from a large corpus) and hence build a common sense encoder (CSE) for TempRel extraction. Finally, a fully connected feed-forward neural network layer is used to generate confidence scores for each label. Global inference is carried out using integer linear programming (ILP) based on the outputted confidence scores.

We extended the work in [5] through a pipeline joint model. The BiLSTM layer is shared for both event and relation tasks in this model. During training, the predictions of the event model were used to construct relation candidates to train the relation model. This strategy generates NONE pairs during training if one argument of the relation candidate is not an event. These NONE pairs helped the relation model to distinguish negative relations from positive ones.

The model was trained with gold events and relation candidates during the first several epochs. We also explored a structured joint model which is a two-stage learning procedure where the best pipleline joint model is taken and re-optimized with SSVM loss. The predicted probabilities from the event detection model are used to filter out non-events since the event model showed a very good performance. In this approach, both event and relation labels are assigned simultaneously during the global inference with ILP. Tokens with POS tags that do not appear in the training set are also filtered out.

2.1 Common Sense Encoder

Humans can often figure out temporal relations between words even without explicit connecting terms like 'while','since','after' etc. For example, we know that 'food' usually comes after 'eat'. Common sense is difficult for computers to imitate. Ning et al. [6] tried to build such a common sense database by acquiring automatically extracted TempRels from a large corpus. The resultant database was called TEMPROB and it contained observed frequencies of commonly observed tuples.

TEMPROB suffers from a lot of limitations when it comes to unseen tuples. To overcome this issue in our work, an updated version of TEMPROB was fitted with a Siamese network which is able to generalise well to unseen tuples also. Once trained, the common sense encoder part remains unchanged while training the LSTM or the feedfoward neural network part. The CSE is used only to generate the final output.

3. System Architecture

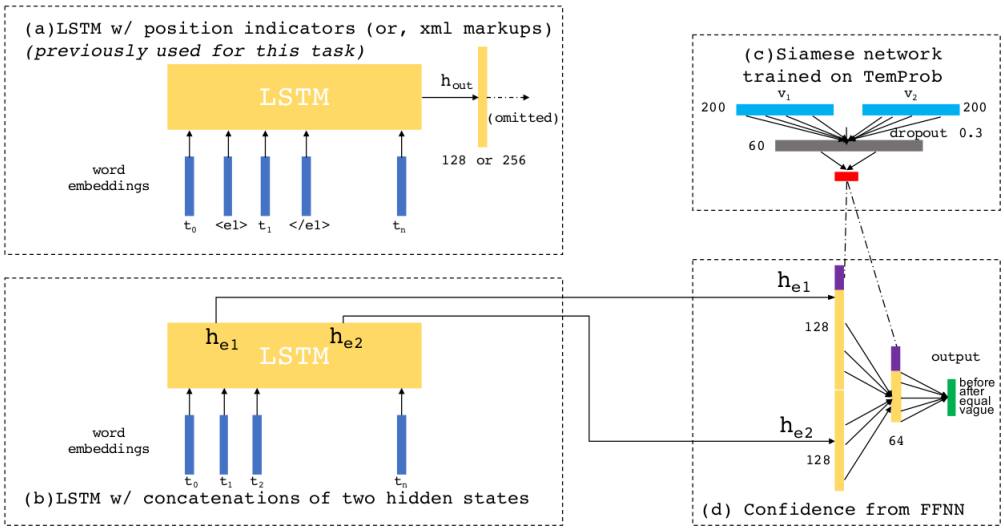

Figure 3.1: The neural network structures in the underlying paper including the siamese network to fit TemProb (adapted from [5])

A major disadvantage of feature based system architecture was that the errors occurred in feature propagates to the subsequent models. So, LSTM networks are used on the TempRel extraction problem as an end-to-end approach that only takes a sequence of word embeddings as input. The word embeddings for a particular pair of events are fed into the LSTMs and a vector representation is obtained, which is followed by a fully connected, feed-forward neural network, that generates the confidence scores for the various output levels.
The LSTMs are trained to indicate the position of the different pair of events which possess the relations amongst them. The hidden states from both the time steps are concatenated to the corresponding location of the said events. The XML markups, introduced as position indicators, indicate the event positions to the LSTM[10].
TemProb is a simple counting model and fails for unseen tuples. So, the CSE is introduced and an updated version of TemProb is fit via a Siamese network that generalizes to unseen tuples too. During training the CSE remains fixed and is only used for the output.
The pipeline joint model was trained by minimizing the cross-entropy loss. The model performances vary significantly with hidden layer dimensions of the biLSTM model, dropout ratio and entity weight in the loss function (with regulation weight fixed at 1.0). A pretrained BERT model was leveraged to compute the word embeddings and all MPL scoring functions had one hidden layer. Hyperparameters were chosen using a standard development set for TB-Dense and a random holdout-set based on an 80/20 split of training data for MATRES. An off-shelf solver provided by Gurobi optimizer was used to solve the ILP during the inference process.

4. Test-Cases and Implementation

In this section, we will first provide a brief overview regarding the temporal relation data and also the datasets used in this paper, and also touch upon the respective evaluation metrics.

4.1 Temporal Relation Data

RED and TimeBank are examples of Temporal Relation Corpora, which facilitate the research in temporal relation exchange, but the common issue in these corpora is the problem of missing annotations. Collecting densely annotated temporal relation corpora with all events and relations fully annotated is reported to be a challenging task as annotators could easily overlook facts regarding the dataset, hence making modeling and evaluation extremely difficult in previous event temporal relation research.

4.1.1 Data

The following model has been trained on the same dataset as the previous paper from which we have extended the motivation, which are TCR and MATRES dataset. The TCR dataset contains both temporal and causal relations and hence we would only require the temporal part. The MATRES dataset consists of 275 news articles from the TempEval3 workshop, with the newly annotated events and TempRels. It consists of 3 sections: Platinum(PT), TimeBank(TB), and AQUAINT(AQ). We here also respectively follow the official split, which is TB+AQ for training and PT for testing, and hence we further set aside 20% of the training data as the development set to tune learning rates ad epochs. For both the datasets, the label set includes *before*,*after*,*equal* and *vague*.

Figure 4.1: **Pipeline Model**

4.2 Implementation

Here, we describe the implementation details of the baselines and our respective model to build an end-to-end event temporal relation extraction system with an emphasis on the defined extended model, and hence we would compare and contrast on them as the respective model might be an improvement.

4.2.1 Baselines

This is the small description of the model implementation of the original paper, they evaluate their model on the TCR and MATRES dataset by leveraging convectional learning algorithms, and based on manually designed features to obtain separate models for events and temporal relation extraction as a pipeline.

4.2.2 End-to-End Event Temporal Relation Extraction

The model defined as an end-to-end system can be achieved by training event detection and relation prediction models with gold labels, separately. The BiLSTM layer is shared for both the event and relation tasks, and during training the predictions of the event model are used to construct relation candidates to train the relation model, and hence this generates the NONE pairs during training if one argument of the relation candidate is not an event. Therefore, the NONE pairs will help the relation model to distinguish negative relations from positive ones, and thus become more robust to event predictions errors

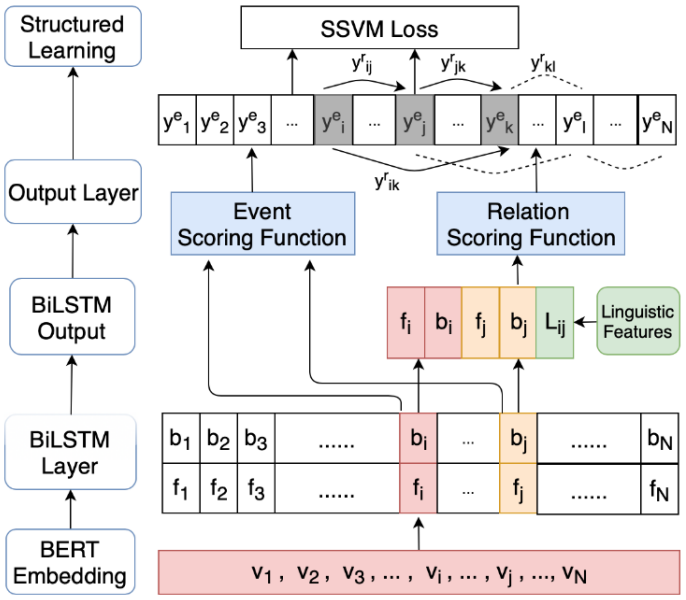

Figure 4.2: This is the deep neural network architecture example of the joint structured learning model. In the structured learning layers, the grey bars denote the tokens being predicted as events. Edge type between events follow the same notations as $y_l^e = 0$ (non- event), so all edges connecting to y_l^e are NONE. These global assignments are input to compute the SSVM loss

5. Experiment and Execution of the Code

We train this model with gold events and relation candidates during the first several epochs in order to obtain a relatively accurate event model.

5.1 Evaluation Metrics

We define the respective standard micro-average scores, and hence for the densely annotated data, the micro average metric should have the same precision, recall and F1 scores, also since the model includes NONE pairs, we follow the convention of the defined tasks and exclude them from evaluation.

5.2 Code Execution

5.2.1 Data Information

tcr-temprel.xml is the pre-processed data of TCR. Since TCR is only used as an evaluation dataset, hence we don't split it.
trainset-temprel.xml and testset-temprel.xml are the pre-processed data of MATRES, MATRES is therefore split into train and test datasets. The proposed model is hence trained on the train split of MATRES.

5.2.2 Code Structure

This section explains what the code contains and how it is analogous with the respective logic. The main python file would contain the Pipeline joint model, which would contain the respective functions of getting the Neural Network python code, which would then be trained accordingly through epochs, which would then hence get trained and eventually we would be able to use that particular trained model for predictions. The code is analogously representing that is is being trained with the candidate relations generated by event module.

The python code consists of Bert Classifier Class which is an extension of the Neural Network Class, and hence is the main architecture of the model. We respectively define the forward, train, relations functions for the model so that they would assist in training the required weights.
The code also consists of an Event Evaluator class which is used for evaluating the respective trained model with respect to the defined dataset, and hecne measure the model's accuracy scores.

6. Results

The final accuracy and F1 Scores are reported for the different cases and including the pre-trained word embeddings - GLoVe, ELMo and BERT. Further, the common sense encoder is added into the baseline Concat model.

	Emb	Acc	F1	Faware	Avg
P.I.	word2Vec	61.7	64.5	58.4	61.45
	GloVe	63.8	67.6	61.1	64.4
	ELMo	67.1	72.1	62.6	67.4
	BERT	66.7	72.8	61.1	66.9
Concat	word2Vec	63.9	68.3	59.1	63.7
	GloVe	63.2	66.2	57.1	61.65
	ELMo	65.3	71.2	62.8	67
	BERT	68.3	72.8	59.6	66.2
Concat+C.S.E	ELMo	68.9	74.3	65.8	70.1
	BERT	69.6	75.9	65.9	70.9

Here we observe that including the given CSE on top of the model increases the overall accuracy by about 5%. Further looking at the Structured joint model, the results are as follows :

	Event	Relation (G)	Relation (E)
Baselines	85.2	65.9	52.8
Pipeline Joint	87.2	-	58.5

Compared to the single-task model, the multi-task model improves F1 scores by 1.5%, while the pipeline joint model improves F1 scores by 1.3%. The pipeline joint model reaches the best end-to-end F1 score at 59.6%, which outperforms the base- line system by 6.8% and the single-task model by 2.4%.

6.1 Conclusion And Future Work

With the availability of high quality and huge datasets like MATRES and TCR and seeing the revolution of LSTMs being used in many works achieving state-of-art and especially on the MATRES and TCR datasets we replenish this to build an end-to-end temporal relation extraction model along with the siamese network which is pre-trained on TempProb dataset which facilitates a greater confidence for common sense events this achieves a greater improvement over the previous state-of-art techniques.
We are able to establish a complete end to end task to extract events and then relation between them but this leads to progression of loss from one stage to other having achieved this we can further progress by making modifying the existing architecture by using a lstm learn from both event extraction and temporal extraction and by using SSVM to simultaneously extract check for events and extract relation between events.

Bibliography

[1] Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, page 173–176, USA, 2007. Association for Computational Linguistics.

[2] Quang Xuan Do, Wei Lu, and Dan Roth. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, page 677–687, USA, 2012. Association for Computational Linguistics.

[3] Artuur Leeuwenberg and Marie-Francine Moens. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[4] Yuanliang Meng and Anna Rumshisky. Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 527–536, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[5] Qiang Ning, Sanjay Subramanian, and Dan Roth. An improved neural baseline for temporal relation extraction. *EMNLP*, 2019.

[6] Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. Improving temporal relation extraction with a globally acquired statistical resource. *arXiv preprint arXiv:1804.06020*, 2018.

[7] Qiang Ning, Hao Wu, and Dan Roth. A multi-axis annotation scheme for event temporal relations. *ACL*, 7 2018.

[8] Qiang Ning, Hao Wu, and Dan Roth. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[9] Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[10] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. 08 2015.

7. Appendix

7.1 Contributions

Varun Gupta(18MA20050)
Added the joint temporal relation extraction with shared representations on top of the base model

Koushiki Dasgupta Chaudhuri(18MA20020)
Helped in Designing the system architecture of the LSTM layer and the Siamese architecture

Altaf Ahmad(18MA20005)
Looked into the literature review and helped in the design of the experiments and the code structure.

C Shanmukh Siva Sai(20EC10022)
Compiled the final results and observations

7.2 Acknowledgments

The authors would like to thank their Natural Language Processing(CS60075) Professor Sudeshna Sarkar for giving them the opportunity to work on this term project and for motivating us and for her guidance as well as providing the necessary information to investigate NLP in a more practical setting, with more focus on the topic of temporal relations, from which we were able to write this paper.

We would also like to thank our Teacher Assistant Mr. Alapan Kuila for his assistance throughout the course and during the project work without which this could not have been completed in the designated time. We would like to thank all the other people who supported and helped in any way possible which led to the successful completion of this task.

We have put all our efforts in this project, however it wouldn't have been possible without the kind and complete support and help of many individuals and organisations. We would like to extend our sincere thanks to all of them.

7.3 Reference Solution: Code including working model

Link to the working model of code: Working Model