CMU_CS_Event TAC-KBP2017 Event Argument Extraction System

Andrew Hsi
Carnegie Mellon University
Pittsburgh, PA 15213 USA
ahsi@cs.cmu.edu

Jaime Carbonell
Carnegie Mellon University
Pittsburgh, PA 15213 USA
jgc@cs.cmu.edu

Yiming Yang
Carnegie Mellon University
Pittsburgh, PA 15213 USA
yiming@cs.cmu.edu

Abstract

In this paper, we describe the CMU_CS_Event team's participation in the Event Argument Linking Task. We expand upon our cross-lingual extraction system from last year's evaluation, in particular focusing on increased integration of our existing pipeline with other high quality NLP systems from CMU. We find that making such improvements to earlier steps in the pipeline can result in substantial gains on event argument extraction performance.

1 Introduction

The CMU_CS_Event team participated this year on the Event Argument Linking Task, which requires participants to extract all mentions of event arguments, and link together arguments belonging to the same event. We submitted results from 5 runs on all three languages: English, Chinese, and Spanish.

Our core system is based on our cross-lingual event extraction system submitted to last year's TAC-KBP evaluation (Hsi et al., 2016a), which trains a single cross-lingual model that can be applied to all three languages. This year, we have expanded upon this system by focusing primarily on further improvements to earlier steps in the pipeline. The goal of such work is to reduce the effect of error propagation on our final argument results. For example, if the entity extraction component fails to detect a particular entity candidate, then we will be unable to recover any argument extractions with this entity. Similarly, failing to detect the existence of a particular event nugget will make it impossible to detect the arguments associated with this event.

The rest of this paper is organized as follows. We begin by introducing necessary terminology for the event extraction task in Section 2. We then describe our overall system architecture in Section 3. In Section 4, we show some experimental results on last year's English evaluation data, as well as our official results on the 2017 evaluation set. Finally, we offer conclusions and ideas for future work in Section 5.

2 Terminology

We begin by reviewing relevant terminology for event extraction.

- An *event* is something that happens in the world at a particular place and time.
- An *event mention* is a particular occurrence of an event in a document. An event may be mentioned multiple times within the same document, or the same event may be mentioned across a set of documents.
- An *event nugget* is a particular word or phrase that signifies the existence of an event.
- An *event argument* is an entity that fulfills some role within a particular event. The set of valid roles for an event depends on the type of event, including roles such as Agent, Place, and Time.
- An *event argument mention* is a particular textual instance of an event argument.

3 System Architecture

We use the following system architecture for event argument extraction. We begin by preprocessing the texts to obtain tokenizations, part-of-speech tags, and dependency parses. We then obtain entity candidates using two different modules, and take the union of these results as our overall entity mention output. Next, we perform event nugget detection using three separate modules, once again taking the union of the output as our final set of event nuggets. Finally, we perform event argument extraction using the entity candidates and event nuggets, followed by argument realis classification. The results from this are then used in a postprocessing step to match the specified output format for the TAC-KBP task, and to link arguments to their associated events. The overall pipeline can be seen in Figure 1.

3.1 Preprocessing

Our preprocessing step begins by running the Stanford CoreNLP tool on the input documents to obtain segmentation, tokenization, and part-of-speech tags (Manning et al., 2014). We then obtain dependency parses for English using CoreNLP, and for Chinese and Spanish using MaltParser (Nivre et al., 2007).

3.2 Entity Mention Extraction

We obtain entity mentions using two different modules. We first train a condition random field (CRF) (Lafferty et al., 2001) for each of the three languages using the Stanford Named Entity Recognizer (NER) (Finkel et al., 2005). For data, we utilize the ACE 2005 and RichERE data. We then further augment these entity extractions by taking the union of these results with the output from the CMU Entity Discovering and Linking (EDL) team's system. While a simple union may result in lower precision, in practice we find that the potential gains in recall far outweigh the loss of precision.

3.3 Event Nugget Detection

We obtain event nuggets from three different modules.

Our simplest module is to train a logistic regression classifier to make the predictions, using LIBLINEAR (Fan et al., 2008). For every word in each document, we classify the word into one of the event types in the TAC ontology, or "NONE" if the word is not an event nugget. We obtain our features for this model using the results of our preprocessing step (see Table 1 for specific features used). Word embeddings are obtained using word2vec (Mikolov et al., 2013) for all three target languages (English, Chinese, Spanish) using their respective Wikipedia dumps. A combination of ACE2005 and RichERE data is used for training.

Our remaining two modules for event nugget detection come from the CMU Event Nugget Detection and Coreference team. In the first of these two modules (Liu et al., 2015; Liu et al., 2016), a discriminatively trained CRF model is deployed to detect mention spans and events. The final module is a bidirectional long short-term memory (LSTM) with a CRF layer at the top. The LSTM is initialized with pre-trained 50-dimensional word embeddings. The CRF model is applied to English and Chinese texts, while the LSTM model is applied to English and Spanish texts. Both models are trained using RichERE data.

As in the entity mention extraction step, we once again take the union of event nugget output from these three systems as our final event nugget output.

3.4 Event Argument Extraction

Using the merged output from our three event nugget modules, we then proceed to extract event arguments. For every (event nugget, entity mention) pair within a sentence, we classify the relationship between the pair into one of the TAC argument roles, or "NONE" if no such relationship exists. We perform this classification using a logistic regression classifier trained with LIBLINEAR.

Features for our model are derived from the preprocessed texts, extracted entities, and extracted event nuggets. The specific features for this model may be seen in Figure 2.

3.5 Realis Classification

We perform realis classification on each argument found by the event argument extraction component into one of ACTUAL, GENERIC, or OTHER. To do this, we train an additional logistic regression classifier with LIBLINEAR, using similar features to the argument detection module.

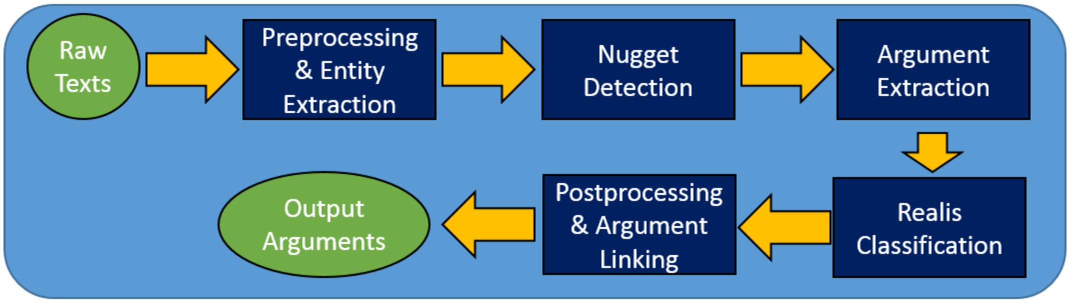

Figure 1: Architecture for our event extraction system.

Event Nugget Detection Features
Lexical features (e.g. words and lemmas within a context window)
Length of the current word
Language-specific POS tags within a context window
Universal POS tags within a context window
Word embedding vector for current word
Dependent/Governor information from dependency parsing

Table 1: Features used in the Event Nugget Detection component

Event Argument Extraction Features
Lexical features about the entity phrase
Lexical features for individual words in the entity phrase
Entity type
Event type and subtype of associated event nugget
Existence of any other candidate entities in the same sentence
Distance between the event nugget and entity
Dependent/Governor information from dependency parsing

Table 2: Features used in the Event Argument Extraction component

3.6 Postprocessing and Linking

After obtaining a final set of extracted event arguments and realis labels, we perform postprocessing to match the required output format defined by the TAC-KBP Event Argument Linking task. In addition, we link together arguments belonging to the same event, using one of three different strategies (different runs use different linking approaches).

Our baseline linking strategy is to link together all arguments that belong to the same event type. Our second linking strategy is to first use the event nugget coreference output from the CRF-based model, and link together all arguments that are attached to coreferent nuggets. Our final linking strategy is to do the same thing as the second approach, but using the coreference links from the LSTM-based model. (Note that the event nugget algorithms for Spanish do not currently create coreference links, so Spanish arguments under this strategy are only linked if they are associated with the same event nugget mention.)

3.7 Cross-lingual Training

Borrowing from our approach last year, we once again use cross-lingual training to create a single, cross-lingual model, rather than separate models for each language. This is motivated by previous success in the NLP literature for cross-lingual applications (Richman and Schone, 2008; Zeman and Resnik, 2008; Snyder et al., 2009; Chen and Ji, 2009; Cohen et al., 2011; McDonald et al., 2011; Piskorski et al., 2011; Ammar et al., 2016; Hsi et al., 2016b). Such models can be particularly useful when there is little training data available for a particular language (as is the case for Spanish event extraction), but much more data available for other resource-rich languages (e.g. English).

Our model uses a combination of language dependent and language independent features, which allows the model to capture general patterns across languages as well as specific nuances found in individual languages. Our language independent features cover information obtained by Universal POS tags (Petrov et al., 2012), nugget type information, entity type information, and Universal Dependencies (McDonald et al., 2013), while our language dependent features includes information based on individual words, language-specific part-of-speech tags, and word embeddings. The overall model is then trained by simply using all available annotated data (across all three languages) at training time.

4 Experiments

In this section we present our experimental results. We will begin first with internal experiments on the 2016 TAC KBP evaluation data, and then proceed to our results on the 2017 evaluation data.

4.1 Internal Experiments

We conducted experiments on the 2016 English evaluation data to determine the effects of improved event nugget detection on event argument extraction. We compare four approaches for event nugget detection: the logistic regression classifier, the conditional random field, the LSTM model, and the union of these three systems. Entity mention candidates were extracted using Stanford NER.

Results may be seen in Table 3. We find that the union of the three systems clearly provides the best results, with a 30.1% improvement on F1 over the best individual system. While taking a simple union of nugget results causes a drop in precision, the large increase in recall results in substantial F1 improvements.

4.2 Official Results

For the official evaluation, we ran our system on all three target languages, submitting 5 different runs:

- Run 1 – event nuggets from all three modules, entity results from Stanford NER and CMU EDL, argument linking between events of same type

- Run 2 – event nuggets from all three modules, entity results from Stanford NER and CMU EDL, argument linking based on coreference links from CRF event nugget model

- Run 3 – event nuggets from all three modules, entity results from Stanford NER and CMU EDL, argument linking based on coreference links from LSTM event nugget model

- Run 4 – event nuggets from logistic regression only, entity results from Stanford NER and CMU EDL, argument linking between events of same type

- Run 5 – event nuggets from logistic regression only, entity results from Stanford NER only, argument linking between events of same type

Results for each of our five runs on English, Chinese, and Spanish may be seen in Tables 4, 5, and 6 respectively. All presented results are under the WithRealis evaluation setting. We report the precision, recall, and F1 for argument extraction, as well as the official error-based argument scores and B^3 linking scores over bootstrapped sampled results at the median, 5% and 95% confidence intervals. (Note that the results from Runs 1, 2, and 3 are identical on all metrics except for linking scores, as the linking strategy is the only differing component among these three runs.)

On English, we find the strongest performance under Run 1, using nugget output from all three systems, entity output from both Stanford NER and CMU EDL, and linking by event type. When using only the logistic regression nugget output (Run 4), performance drops substantially on all metrics. Dropping the EDL output (Run 5) results in an additional reduction of scores. These results clearly show how important earlier steps are in the event extraction pipeline. All five runs utilize the exact same argument classifier, but F1 for argument extraction nearly doubles just from improvements to earlier stages of the pipeline.

	Precision	Recall	F1
Logistic Regression	**36.20**	5.17	9.04
CRF-based model	31.02	6.85	11.22
LSTM-based model	31.32	3.10	5.64
Merged Union	30.45	**9.60**	**14.59**

Table 3: Results on 2016 Event Argument Extraction Evaluation Data

When considering linking strategies, we find that the simple method of linking together all arguments by their event type remains a strong baseline. Our coreference-based linking strategy gives slightly lower performance on the B^3 scores compared to this approach.

On Chinese, we find similar results to English. The best performing system is seen with Run 1, and the best linking strategy remains the "link by event type" approach. Removing the additional event nugget modules and EDL both cause drops in overall performance, further indicating how important it is to have strong components in the earlier stages of the pipeline.

On Spanish, we find that the best argument scores once again come from using all event nugget and entity detection modules. Using just the logistic regression nuggets and Stanford NER entities results in less than one third of the F1 score from our best system. For argument linking, Runs 2 and 3 give the best performance. As neither the CRF-based nor the LSTM-based event nugget algorithms produce coreference links for Spanish, both runs degenerate into the same strategy of linking arguments together only when they are derived from the same event nugget mention. This ultimately gives superior performance on the Spanish data when compared to our baseline strategy of linking together all arguments of the same event type.

5 Conclusion

We submitted 5 systems this year to the Event Argument Linking Task. Our systems were based on an approach of cross-lingual training over English, Chinese, and Spanish texts, designed to leverage information from both within a particular language and across a set of languages. We have shown that improvements to earlier stages of the pipeline enable substantial gains in performance on both internal evaluation with last year's evaluation corpus as well as on this year's official evaluation scores.

Our evaluation results suggest several promising directions for future work. The improvements made to the earlier stages of our event extraction pipeline have shown the potential for major boosts in event argument performance, particularly with regards to improvements made on event nugget detection. Exploring further improvements to event nugget detection is likely to be of great benefit to event argument extraction. Another interesting area for exploration is to consider more sophisticated ways of combining modules that perform the same NLP task. For example, our current system only takes the union of event nugget output across three different event nugget algorithms. Experimenting with other ways of combining these results may enable better solutions that using a simple union.

Acknowledgments

This research was supported in part by DARPA grant FA8750-12-2-0342 funded under the DEFT program and by the National Science Foundation (NSF) under grant IIS-1546329.

References

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. In *TACL*.

Zheng Chen and Heng Ji. 2009. Can one language bootstrap the other: A case study on event extraction. In *NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *EMNLP*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. In *JMLR*.

	P	R	F1	Arg_5	Arg_M	Arg_95	Link_5	Link_M	Link_95
CMU_CS_Event1	21.99	**6.84**	**10.44**	**2.13**	**2.53**	**2.94**	**1.48**	**1.76**	**2.08**
CMU_CS_Event2	21.99	**6.84**	**10.44**	**2.13**	**2.53**	**2.94**	1.30	1.56	1.87
CMU_CS_Event3	21.99	**6.84**	**10.44**	**2.13**	**2.53**	**2.94**	1.41	1.69	2.02
CMU_CS_Event4	**32.46**	3.43	6.20	1.81	2.09	2.39	0.57	0.74	0.97
CMU_CS_Event5	30.78	2.89	5.28	1.53	1.80	2.07	0.36	0.50	0.67

Table 4: English results in official TAC KBP 2017 Evaluation

	P	R	F1	Arg_5	Arg_M	Arg_95	Link_5	Link_M	Link_95
CMU_CS_Event1	28.84	**7.82**	**12.30**	**3.49**	**4.00**	**4.51**	**1.37**	**1.71**	**2.14**
CMU_CS_Event2	28.84	**7.82**	**12.30**	**3.49**	**4.00**	**4.51**	1.29	1.57	1.90
CMU_CS_Event3	28.84	**7.82**	**12.30**	**3.49**	**4.00**	**4.51**	1.13	1.36	1.65
CMU_CS_Event4	**43.39**	3.39	6.29	2.22	2.73	3.13	0.42	0.62	0.84
CMU_CS_Event5	43.23	3.01	5.63	2.01	2.50	2.88	0.22	0.33	0.47

Table 5: Chinese results in official TAC KBP 2017 Evaluation

	P	R	PF1	Arg_5	Arg_M	Arg_95	Link_5	Link_M	Link_95

Mining wiki resources for multilingual named entity recognition. In *ACL*.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2009. Adding more languages improves unsupervised multilingual part-of-speech tagging: A bayesian non-parametric approach. In *NAACL*.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *IJCNLP*.