CMU-LTI at KBP 2016 Event Nugget Track

Zhengzhong Liu and **Jun Araki** and **Teruko Mitamura** and **Eduard Hovy**
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
{liu, junaraki, teruko, hovy}@cs.cmu.edu

Abstract

In this paper, we describe the CMU LTI team's participation in TAC KBP 2016 event nugget track. This year, we extend our feature based event detection and coreference systems to process also Chinese documents. We also conduct experiments using Neural Network based models for English event nugget detection, which can enable us building models that can be easily transfer to different languages. Our feature based English Nugget Detection and Coreference systems both rank number 2 among all the participants. The Chinese counterpart ranks first in English Nugget Detection and second in English Coreference.

1 Overview

The CMU LTI team participates in the event nugget task of TAC-KBP 2016. The event nugget evaluation task this year requires participants to perform end-to-end event nugget detection and coreference. The datasets are consisted of 3 languages (English, Chinese and Spanish) on two genre of text documents (News and discuss forum data). The CMU LTI team submitted systems for English and Chinese.

We experiment with two types of nugget extractions systems: a neural network based event extraction system and a traditional Conditional Random Field (CRF) based event extraction system which is adopted from last year. For event coreference systems, we mainly use the structured Latent Tree method, which is also adopted from our TAC 2015 system. In addition, we train a SVM classifier for realis detection. Our final submission is computed via a pipelined version of the three stages.

2 Datasets

The testing data of KBP 2016 contains two genre: Forum and News data. The TAC 2015 Engish event nugget training and test data contain both genre. However, Rich ERE datasets released for Chinese event nugget are all of forum genre. We augment the Chinese data with the Chinese training data from ACE 2005.

3 Event Nugget Detection

3.1 Preprocessing

For the feature based models, we run a set of NLP annotators to help feature extraction. We run Stanford CoreNLP pipeline on both Chinese and English datasets to conduct tokenization, parsing, Named Entity Recognition and Entity Coreference. We use Semafor (Das and Smith, 2011) and Fanse (Tratz and Hovy, 2011) parser to conduct semantic role labeling on English text. For Chinese, we run ZPar (Zhang et al., 2013) to get character level syntatic parse and use the semantic parsing module in Language Technology Platform (Che et al., 2010) to get semantic role labeling.

3.2 Conditional Random Field Models

We deploy a discriminativly CRF model to detect mention span and event. The CRF model is trained with the structured perceptron (Collins, 2002). Our final system always make use of the average weight variation as described in Collins (2002).

The feature set used in our English event nuggets this year is similar to what we use in KBP 2015, which are summarized as followed:

1. The target word itself and the direct dependent words of the target.

2. Coarse Part-of-Speech (2-character), lemma, lemma+pos and named entity tag of words in the 2-word window of the target (both side).

3. The combination of previous and next word's POS and lemma with the target word's POS and lemma.

4. Brown clusters (Sun et al., 2011), WordNet Synonym and derivative forms of the trigger.

5. Whether surrounding words match some selected WordNet senses, these senses are "Leader", "Worker", "Body Part", "Monetary System", "Possession", "Government", "Crime" and "Pathological State".

6. Closest named entity type.

7. Dependency features, including lemma, dependency type and part-of-speech of the child dependencies and head dependencies.

8. Semantic role related features includes the frame name and the argument role, named entity tag, argument head word lemma and WordNet sense (selected from the above list as well) of the arguments.

To extend our system to handle Chinese documents, we develop similar features for Chinese. Most of the features can be reused without changes in the Chinese system, which includes: window based features[1], syntactic based features, entity features, head word features and SRL features. We also use the Brown clustering features with clusters induced form Chinese Gigaword 3 [2].

In addition, Chinese tokens normally contain internal structure and each single character in the token may convey useful semantic information. We add the following character related features:

1. Whether the token contains a character.

2. The contained character and its character level Part-of-Speech.

3. The first character of the token.

4. The last character of the token.

5. Base verb structure feature as described in (Li et al., 2012): we use a feature to represent one of the base verb structure. In addition to the 6 main structures proposed by Li et al. (2012), we added 3 structures for completeness: 1) No verb character found 2) The verb character is found after 2 characters and 3) Other: any cases that are not defined above.

3.3 A trick to deal with Chinese data

During the system development, we observe that our Chinese system suffers from serious low recall despite all the features we added in. By following the training procedures, we hypothesize that the annotated Chinese data is not complete (see §7.1 for more discussion). As a result, our learning algorithm will be biased by the missed events and learn incorrect negative signals. The final model thus will be very conservative in making predictions, leading to a low recall.

The problem can only be solved by manually polishing the data, which is too expensive for us. We mitigate the problem by ignoring all training sentences that do not contain an event mention, which reduce the probability of missed annotations. On our development experiments, we found that this simple trick can directly raise our nugget detection performance by about 3%. The performance improvement also support our hypothesis that the Chinese dataset is indeed not fully annotated.

3.4 Neural Network Models

We also employ bidirectional Gated Recurrent Unit (GRU) for event nugget detection. GRU (Cho et al., 2014) is a type of recurrent neural networks (RNNs). A standard RNN takes as input a word embedding $\mathbf{x}_t$ at time step t, and iteratively computes a hidden state $\mathbf{h}_t$ as follows:

$$\mathbf{h}_t = f(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \qquad (1)$$

where f is a nonlinear activation function such as the element-wise logistic sigmoid function. It is known

[1] However since the discussion forum training data are quite noisy, we restrict the POS window to 1 instead.

[2] http://www.cs.brandeis.edu/~clp/conll15st/dataset.html

that it is difficult for RNNs to learn long-term dependencies mainly because the gradient can often explode or vanish over long sequences (Bengio et al., 1994). GRU is a RNN-based neural architecture to mitigate the problem. GRU introduces a reset gate $\mathbf{r}_t$ to control the use of previous hidden state:

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1}) \quad (2)$$

where σ is the element-wise logistic sigmoid function. GRU uses an update gate $\mathbf{z}_t$ computed by:

$$\mathbf{z}_t = \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1}) \quad (3)$$

The hidden state $\mathbf{h}_t$ is computed as follows:

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_{t-1} \quad (4)$$

where $\odot$ is the element-wise product, and $\tilde{\mathbf{h}}_t$ denotes the candidate activation given by:

$$\tilde{\mathbf{h}}_t = \sigma(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \quad (5)$$

As shown in (4), the update gate $\mathbf{z}_t$ controls how much information from the previous hidden state carries over to the current hidden state. This helps GRU to remember long-term information.

As compared to feature-based models such as CRF described in Section 3.4, neural models have an advantage that they are able to learn representations and tune parameters without relying on external tools for feature extraction.

To initialize the word embeddings, we use 300-dimensional pre-trained word embeddings[3] trained on a 6B token corpus with GloVe (Pennington et al., 2014). The embeddings are updated through back-propagation during training. We train the model on the ACE 2005 corpus[4] (Walker et al., 2006) and the TAC KBP 2015 event nugget corpus[5] (Liu et al., 2015b). The TAC KBP 2016 event nugget task defines a smaller set of event types than ACE 2005 and TAC KBP 2015. Thus, we only use event triggers or nuggets in ACE 2005 and TAC KBP 2015 which have event types compatible with the event ontology for TAC KBP 2016. We set the dimention of the hidden state to 500. We use Adam (Kingma and Ba, 2015) with the initial learning rate 0.001. To better deal with unseen words that do not appear in training data, we use a technique suggested by (Yao et al., 2013). That is, we choose a small number of words that occur only once in the training dataset, and mark them as <UNK>. The learned representation of <UNK> through training is used to represent the unseen words. Similarly, we also mark numbers as <DIGIT> to learn a single representation for numbers, following (Collobert et al., 2011).

[3] http://nlp.stanford.edu/projects/glove/

[4] https://catalog.ldc.upenn.edu/LDC2006T06

[5] http://www.nist.gov/tac/2015/KBP/Event/index.html

4 Realis Classification

Our realis classification is the same as the system described in (Liu et al., 2015a), which use a simple logistic regression model with shallow lexical features. Our in-house performance shows that such simple approach can reach a performance between 60% to 70% (varies depends on the genre and language of the document) given oracle mention span and type. However, in our submission we have an implementation problem which make our system always produce "Actual". We conduct experiments after the evaluation, and obtain expected results by using the correct Realis module.

5 Event Hopper Coreference

Similar to last year, We use the latent antecedent tree method (Fernandes et al., 2012; Björkelund and Kuhn, 2014) to conduct coreference. The features employed can be roughly classified into 3 categories:

Trigger match: exact and fuzzy match on the trigger word, uses standard linguistic features (pos, lemma, etc.) and resources like Brown Clustering and WordNet. Information from mention type and realis type are also used;

Argument match: exact and fuzzy match on the arguments, including their string, argument role and coreference information;

Discourse features: encodes sentence and mention distances.

Forum features: To encode the information flow in discuss forum data, we record whether the authors of the sentences are the same, or whether the second sentence quote the first one.

Algorithm 1 PA algorithm for latent trees

Require: Training data D, number of iterations T
Ensure: Weight vector w

$w = \vec{0}$
for $t \leftarrow 1..T$ **do**
  for $\langle M_i, \mathcal{A}_i, \tilde{\mathcal{A}}_i \rangle \in D$ **do**
    $\hat{y}_i = \arg\max_{\mathcal{Y}(\mathcal{A})} score(y)$
    if $\neg Correct(\hat{y}_i)$ **then**
      $\tilde{y}_i = \arg\max_{\mathcal{Y}(\tilde{\mathcal{A}})} score(y)$
      $\Delta = \Phi(\tilde{y}_i) - \Phi(\hat{y}_i)$
      $\tau = \frac{\Delta * w}{||\Delta||^2}$
      $w = w + \tau\Delta$
return w

We train the Latent Tree model with a passive-aggressive algorithm(Crammer et al., 2006) similar to that of Björkelund and Kuhn (2014). Our implementation is slightly difference in the Passive-Aggressive step. Our algorithm is detailed in algorithm 1, where: $\mathcal{A}$ represent the set of possible antecedents on the left; $\tilde{\mathcal{A}}$ represent the set of antecedents that are allowed by the gold standard coreference; Φ is the feature function over the tree; $\hat{y}$ represent the best decoding given current features; $\tilde{y}$ represent the current best decoding among the correct coreference structure, i.e., the latent tree. The algorithm iterativly update the weight vector in a Passive-Aggressive manner. During implementation, we found that the PA algorithm is important for the algorithm to converge well.

The feature set we selected can be applied to both Chinese and English. To migrate our English system to Chinese, we simply replace some NLP annotators with the Chinese counterpart as discussed above.

6 System Performance

In this section we present our official performance on the Event Nugget tasks:

Both our English and Chinese event detection and coreference systems produce competitive results. Our Chinese system is the first place based on all the event nugget attributes. Our English system is the second in mention type detection[6]. To our surprise, our English nugget detection performance drops about 13% (span and type) comparing to last year. However, our relative ranking is almost unchanged. We leave the investigation of the problem to future work. Since the event coreference component and coreference evaluation relies highly on the performance of nugget detection, we also observe a big drop in coreference peroformance comparing to last year.

[6]Due to the Realis component bug our combined ranking drops.

		Prec.	Recall	F1
Span	LTI1	69.82	39.54	50.49
	LTI2	63.31	30.34	41.09
Type	LTI1	61.69	34.94	44.61
	LTI2	56.33	26.87	36.39
Realis	LTI1	45.78	25.93	33.11
	LTI2	43.19	20.60	27.90
All	LTI1	40.19	22.76	29.06
	LTI2	38.59	18.41	24.92

Table 1: Performance on English Nugget Detection

	B^3	CeafE	MUC	BLANC	Aver.
LTI1	35.06	30.45	24.60	18.70	27.23
LTI2	28.89	27.13	16.85	15.09	21.99

Table 2: Performance on English Nugget Coreference

7 Discussion and Conclusion

In this paper we describe CMU LTI's participation in TAC KBP 2016 event track. Our feature based system perform pretty well on English (second place) and Chinese (first place) nugget detection. However, there are still some potential problems to be solved.

7.1 Chinese Data Annotation

We hypothesize that the Chinese datasets are not fully annotated. We take a closer look in the data and found a number of missed event nuggets. Here we list a couple examples:

(6) 支持香港同胞争取[Personnel.Elect 选举]与被[Personnel.Elect 选举]权！

(7) 司务长都是骑着二八去[TransferOwnership 买]菜去。

(8) 海豹行动是绝密，塔利班竟然可以预先得知？用个火箭就可以[Conflict.Attack打]下来，这个难度也实在是太高了吧。

In the above examples, we show several event nuggets. However, mentions annotated in red are not actually annotated in the Rich ERE datasets. Especially, in example 6, the first 选举 is annotated

		Prec.	Recall	F1
Span	LTI1	56.46	39.55	46.52
	LTI3	56.19	35.35	43.4
Type	LTI1	50.72	35.53	41.79
	LTI3	49.7	31.26	38.38
Realis	LTI1	42.7	29.92	35.18
	LTI3	43.11	27.12	33.29
All	LTI1	38.91	27.26	32.06
	LTI3	38.54	24.25	29.77

Table 3: Performance on Chinese Nugget Detection

	B^3	CeafE	MUC	BLANC	Aver.
LTI1	35.06	30.45	24.60	18.70	27.23
LTI3	28.89	27.13	16.85	15.09	21.99

Table 4: Performance on Chinese Nugget Coreference

but the second one is not. Such inconsistencies happen a lot across the dataset. When training with such data, the classifier will likely to be quite conservative on making event nugget predictions. We conduct a very simple quantitative analysis by comparing the ACE 2005 Chinese annotation against the Rich ERE Chinese annotation. Table 5 and Table 6 summarize the top 5 double-character tokens annotated in ACE and RichERE. For the most popular event mentions, Rich ERE annotated only a smaller percentage comparing to ACE.

In addition, we find that the most popular event nuggets are mostly single character in the Rich ERE datasets, such as 打(170), 说(148), 死(131), 杀(118) . In fact, in top 20 most popular event nuggets of Rich ERE, there are 17 single-character nuggets, this number is only 6 in ACE. These single character tokens are more ambiguous comparing to a double character mention (for example, 打 can represent the action of "calling someone" or "attacking someone", which corresponds to very different event type. This is because language in discuss forum posts are normally not formal. This actually challenges our event nugget systems to deal with deeper semantic problems.

7.2 Neural Modeling for Event Detection

The GRU model described in Section 3.4 has several limitations. First, in this work we employed a vanilla implementation of GRU. Recent studies make use of a new kind of features for modeling events and implement them using extra dimensions of word embeddings. One of the most promising features is sentential features (Nguyen and Grishman, 2015; Chen et al., 2015; Ghaeini et al., 2016). Second, we simply used pre-trained word embeddings in the model initialization, but instead we can create word embeddings which might be more suitable to the task of event nugget detection, by training them on a corpus similar to the TAC KBP corpus. This is a technique recently used by Liu et al. (2016) and Feng et al. (2016) in event detection for ACE 2005. Third, more rigorous hyperparameter tuning is desired. In particular, we did not use dropout (Hinton et al., 2012) in this work, and it would alleviate the issue of overfitting, leading to better generalization.

Token	Annotated	Total	%
冲突	100	119	84.03%
访问	64	90	71.11%
受伤	53	59	89.83%
死亡	46	50	92.00%
前往	44	52	84.62%

Table 5: Top 5 double character mentions in ACE

Token	Annotated	Total	%
战争	96	223	43.05%
死亡	24	33	72.73%
暗杀	22	40	55.00%
入侵	18	22	81.82%
自杀	17	33	51.52%

Table 6: Top 5 double character mentions in ERE

7.3 Towards Unsupervised Methods

Another major limitation of our current systems is that we relies highly on annotated dataset. However, creating high quality event nugget and coreference dataset is very expensive, especially on different languages. Furthermore, the datasets are normally small in size and are narrow in terms of the event types. Recently there are researches on unsupervised event discovery (Huang et al., 2016; Peng et al., 2016). This line of work will be fruitful to investigate. It will be more interesting if we can inject human preferences into the event discovery process. However, how to evaluate the found event mentions will be an inevitable problem to solve.

References

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of ACL 2014*, pages 47–57.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. LTP: a Chinese Language Technology Platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, number August, pages 13–16.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL/IJCNLP 2015*, pages 167–176.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP 2014*, pages 1724–1734.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, pages 1–8.

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*, 12:2493–2537.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. In *Journal of Machine Learning Research 7*, page 551–585.

Dipanjan Das and NA Smith. 2011. Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, volume 1, pages 1435–1444.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of ACL 2016*, pages 66–71.

Eraldo Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Proceedings of EMNLP/CoNLL 2012*, pages 41–48.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of ACL 2016*, pages 369–373.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1502.06922*.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R Voss, Jiawei Han, and Avirup Sil. 2016. Liberal Event Extraction and Event Schema Induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 258–268.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR 2015*.

Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in Chinese event extraction. *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference*, (July):1006–1016.

Zhengzhong Liu, Jun Araki, Dheeru Dua, Teruko Mitamura, and Eduard Hovy. 2015a. CMU-LTI at KBP 2015 Event Track. In *KBP TAC 2015*. NIST.

Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2015b. Overview of TAC KBP 2015 Event Nugget Track. In *Proceedings of Text Analysis Conference 2015*.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging FrameNet to improve automatic event detection. In *Proceedings of ACL 2016*, pages 2134–2143.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of ACL/IJCNLP 2015*, pages 365–371.

Haoruo Peng, Yangqi Song, and Dan Roth. 2016. Event Detection and Co-reference with Minimal Supervision. In *EMNLP 2016*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543.

A Sun, R Grishman, and S Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 521–529.

Stephen Tratz and Eduard Hovy. 2011. A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the Conference on Empirical Methods*

in Natural Language Processing, number 2010, pages 1257–1268.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia. Catalog number: LTDC2006T06.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for language understanding. In *Proceedings of INTERSPEECH 2013*, pages 2524–2528.

Meishan Zhang, Yue Zhang, Che Wanxiang, and Liu Ting. 2013. Chinese Parsing Exploiting Characters. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1326-1336, pages 125–134.