IJCAI–17 Formatting Instructions

Anonymous submission

Abstract

Existing event extraction systems are supervised and often learned from expert-annotated datasets, such as ACE and ERE event extraction program. However, constructing these high-quality corpora is costly, and manually annotated dataset are limited in size and coverage of event types, which makes models learned on these datasets hard to generalize. Inspired by some Freebase schemas which share similar structures with ACE event templates, we investigate the following problems in this paper: can we generate a feasible dataset for event extraction with Freebase automatically and is it possible to extract event on this dataset. We first propose four hypotheses based on our observation and produce our dataset accordingly. Then, we design a neural network model with ILP-based post inference committing to handling two challenging problems in event extraction: multi-type events and multi-word arguments. Finally, manual evaluation demonstrates that the data we generated are feasible, and experimental results of both manual and automatic evaluation prove the effectiveness of our proposed model.

1 Introduction

2 Task Description and Dataset

2.1 Knowledge Base

2.2 Data Generation

We employ Freebase CVT instances to automatically annotate texts in Wikipedia. We regard a sentence as a positive one when it suggests an occurrence of event, or otherwise a negative sentence. The annotation strategy is based on four hypotheses, and we use the following examples to explain and motivate each hypothesis. S1 and S2 are positive sentences and their arguments are in italics and underlined.

S1: *Remedy Corp* was sold to *BMC Software* as the *Service Management Business* Unit in *2004*.

S2: *Microsoft* spent $6.3 billion buying online display advertising company *aQuantive* in *2007*.

Instances of *business.acquisition* in Freebase

mid \ property	company_acuqired	acuqiring_company	date	divisions_formed
m.07bh4j7	Remedy Corp	BMC Software	2004	Service Management Business Unit
m.05nb3y7	aQuantive	Microsoft	2007	NONE

Data generation

Event structures in our dataset

Wiki text	**S1:** *Remedy Corp* was sold to *BMC Software* as the *Service Management Business Unit* in *2004*.			
Event type	*business.acquisition*			
Arguments	company_acuqired	acuqiring_company	date	divisions_formed
	Remedy Corp	*BMC Software*	*2004*	*Service Management Business Unit*

Wiki text	**S2:** *Microsoft* spent $6.3 billion buying online display advertising company *aQuantive* in *2007*.			
Event type	*business.acquisition*			
Arguments	acuqiring_company	company_acuqired	date	divisions_formed
	Microsoft	*aQuantive*	*2007*	——

Figure 1: Examples of CVT instances in Freebase, and labeled sentences in our dataset.

S3: Microsoft hopes aQuantives Brian McAndrews can outfox Google.

S4: On April 29th, Elizabeth II and Prince Philip witnessed the marriage of Prince William.

H1: Positive sentences should contain all properties

This hypothesis indicates that if a sentence has all properties of a CVT, it is more likely to be a positive sentence. We regard the CVT as event type and extract words and phrases that match the properties of a CVT instance as involved arguments.

For example, S1 in Figure 1 contains all properties of instance $m.07bh4j7$ whose type is *business.acquisition*, thus we consider S1 as a positive sentence which expresses an event about *business.acquisition*, and *BMC Software*, *Remedy Corp*, *Serverice Management Business Unit* and *2004* should be labeled as the argument that plays the role of *acquiring_company*, *company_acquired*, *divisions_formed*, *date*, respectively.

However, in practice, we realize that *H1* is too strict that excludes a great many positive sentences like S2 in Figure 1. So we put forward the second hypothesis.

H2: Positive sentences should contain all key properties

This hypothesis is an extension of *H1*, which relaxes "all properties" constraint to "key properties". We define the CVT property that has strong relevance with the occurrence of an

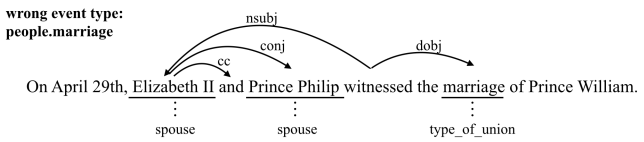

Figure 2: An illustration of dependency parse tree of S4.

event as **key property**. And **key argument** is the word or phrase that matches a key property of that CVT instance.

For example, *company_acquired* and *acquiring_company* are the key properties of CVT *business.accquisition*, with this relaxation, positive sentences like S2 need not contain all properties, but only key properties instead.

The relevance degree between a CVT cvt and one of its property pro can be measured as follows:

$$score_{cvt,pro} = log \frac{count(cvt, pro)}{count(cvt) \times count(pro)} \quad (1)$$

Formula 1 simplifies the calculation of pointwise mutual information, where $count(cvt)$ is the number of all cvt instances, $count(pro)$ is the number of pro, and $count(cvt, pro)$ is the number of cvt instances that contain pro.

H3: Key properties should include time property

This hypothesis strengthens *H2* by counting time property in key properties. We discover that for many CVTs, their key properties do not take into account time property. However, in fact, ignoring time property will produce a large number of negative sentences like S3.

This sentence does not express an explicit event about *business.accquisition* while contain all key properties of an instance, resulting in mistaking *Microsoft* for *acquiring_company*, and *Nokia* for *company_acquired*. By adding *date* to the set of key properties, S3 will be filtered. Therefore, in *H3*, we choose the property which achieves highest relevance degree among all time properties as a supplementary key property.

H4: Positive sentences should contain key properties with close syntactic distance

We introduce another factor, syntactic distance, to annotate positive sentences. Intuitively, two arguments take participant in the same event are likely to be close in syntactic structures. This factor is feasible to eliminate negative sentences, such as S4, which satisfies *H2*.

The syntactic distance can be measured by the distance of two words in dependency parsing tree. We set the maximum distance between two key arguments as 2, denoting that, for a candidate sentence, if a pair of key arguments within it violates this constraint, it is supposed to be negative. Given the dependency parsing tree in Figure 2, S4 is negative because the distance between *Prince Philip* and *marriage* is 3.

We conduct a manual evaluation on the quantity and quality of datasets generated by different hypotheses (see Section 4.2), and utilize the combinations of hypothesis *H3* and *H4* as the final strategy to data generation.

3 Model

4 Experiments

4.1 Experimental Setup

Dataset and Evaluation Methodology

We use the November 20th, 2016 English Wikipedia dump, and generate 7180 sentences, containing 7376 events and 25840 arguments as corpus. We then randomly select 6000 sentences for training and 1180 sentences as test set, and the remained 1200 sentences for validation. However, it costs too much time and labor annotating all sentences in the corpus, so we conducted both automatic evaluation and manual evaluation in the experiments. Specifically, we first manually evaluate the reliability of our test set. Next, we regard the noisy rule-generated data as gold standard and evaluate our model automatically. Finally, we manually evaluate a subset of events detected by our model and analysis the difference with results in automatic evaluation.

Evaluation Measures

We evaluated our models in terms of precision (P), recall (R), and F-measure (F) for each subtask. These performance metrics are computed according to the following standards of correctness:

- For event type classification, an event is correctly classified if its reference sentence contains all key arguments of this event type.
- For argument detection, an argument is correctly detected if its offsets, role, and related event type exactly match any reference argument within the same sentence.
- For event detection, an event is correctly detected if its type and all its key arguments match a reference event within the same sentence.

Training

In our experiments, all hyperparameters are tuned by grid search on the development set. In different stages of event extraction, we adopted different parameters. In event detection, we set the size of word embedding to 200, the size of LSTM layer to 100. In argument detection, we use the same size of word embedding, while the size of LSTM layer is 150, and the size of key argument embedding is 50. During training, we apply the generic stochastic gradient descent [Bottou, 2010] with a dropout rate as 0.5 on both the input and output layers to mitigate overfitting. Word embeddings are pretrained using skip-gram word2vec model [Mikolov *et al.*, 2013] over the whole Wikipedia dump and fine tuned during training.

4.2 Generated Dataset Evaluation

For comparison, we evaluate four dataset that utilize different hypotheses to generate positive sentences from Wikipedia. We randomly select 100 sentences in each dataset, and annotators are asked to determine whether these sentences express events intuitively.

As shown in Table 1, as the strictest hypothesis, H1 guarantees the quality and confidence of generated data, while there are merely 30K CVT instances that contains all properties

Hypothesis	H1	H2	H2+H4	H3	H3+H4
Instances	0.3M	3.6M	3.6M	1.3M	1.3M
Dataset	203	108K	12K	9241	7180
Event type	9	24	24	24	24
Correct (%)	98	22	37	81	89

Table 1: Statistic of generated dataset with different hypotheses. Instances and Dataset denotes the number of instances and sentences that satisfy each hypothesis, respectively. Event type indicates the number of different CVT types in each dataset. Correct represents the percentage of sentences which account as stating events explicitly.

of their corresponding CVT types. And by utilizing these instances, we can only obtain 203 sentences which cover 11 types of events, which is quite insufficient for further training. H2 is looser than H1, though it expands the resulting dataset, it produce a large number of incorrect sentences. This side effect demonstrates that H2 is inappropriate to be used as a soft constraint. Compared with H2, the significant improvement in the quality of sentences generated by H3 proves that CVT properties referring time information are critical to data generation. Among all hypothesesfinally, data obtained by H4 achieves highest precision, which demonstrates that our hypothesis H4 is feasible and it is an effective way to generate reliable data automatically.

4.3 Baselines

To investigate the effectiveness of our proposed model, we develop three baseline extraction systems for comparison, including traditional feature-based methods and neural network models.

For neural network method, we train a long short-term memory network that takes word embeddings as the input, and simply learns a probability distribution over all possible labels.

For feature-based methods, we apply Conditional Random Field [Lafferty *et al.*, 2001] and Maximum Entropy [Berger *et al.*, 1996] to explore a variety of elaborately features which are widely used in traditional ACE event extraction. And both two classifiers share the same feature sets.

Lexical Features

1. Unigrams and bigrams of the current word and its context within a window of size 2.
2. Unigrams and bigrams of part-of-speech tags of the current word and its context within a window of size 2.
3. Unigrams and bigrams of lemmas of the current word and its context within a window of size 2.
4. Synonym set entries in WordNet [Miller, 1995] of the current token.

Syntactic Features

1. The depth of the current words in the parse tree.
2. Dependent and governor words of the current token.
3. The path from root to leaf node of the words in the parse tree.

Entity Information

1. Unigrams and bigrams of named entity mention of the current word and its context words and its context within a window of size 2.
2. Relative distance and entity type of the nearest entity to the current token in the parse tree.
3. Relative distance and entity type of the nearest entity to the current token in the sentence.

We derive these features using Stanford CoreNLP [Manning *et al.*, 2014], and apply the implementation from the CRF++ toolkit [Kudo, 2005] and Le Zhang [1] to train CRF and max entropy classifiers, respectively.

4.4 Automatic Evaluations

Table 2 presents the overall performance of all methods on the full test set.

Comparison with baselines

Traditional feature-based models are inefficient in both event detection and argument detection. Although they can achieve high precisions in event classification and argument detection, they can only extract a limited number of events, resulting in low recalls. Neural-network-based methods performs much better than feature-based models, because they can make better use of word semantic features, especially, LSTM can capture longer range dependencies and richer contextual information instead of neighborly word features. Moreover, neural-network-based methods can avoid errors propagating from other NLP preprocessing tools like POS tagging and NER.

Effect of CRF Layer

Every model which has a CRF layer over its neural network output layer is superior to the one with a simple LSTM layer. Compared with LSTM model, LSTM-CRF achieves higher precisions and recalls in all subtasks by significantly reducing the invalid labeling sequences (e.g., `I-arg` appears right after `O`). During prediction, LSTM-CRF models take into account the constraints between neighbor labels, and co-occurrence relationships between key arguments, rather than tagging each token independently.

Effect of Post Inference

As shown in Table 2, post inference with ILP considerately-improve the overall system performance, especially in event classification. ILP treat our hypothesis about key argument as a constraint for global inference based on the output score of neural network models, and produces a gain of 7.4 in event classification, 1.8 in event detection, and 4.6 in argument detection, with respect to the F1 score.

We further investigate the effect of our heuristic method, LSTM-CRF-ILP$_{multi}$, to deal with the multi-event sentence issue. Compared with other models, LSTM-CRF-ILP$_{multi}$ selects several labeling sequences according to their objective value, and extract a number of events with comparable confidences from a sentence. As we can see from Table 2,

[1] https://github.com/lzhang10/maxent

Model	Event Classification			Argument Detection			Event Detection		
	P	R	F	P	R	F	P	R	F
CRF	96.8	9.93	18.0	64.8	6.54	11.9	29.8	3.06	5.55
MaxEnt	**97.9**	11.4	20.3	64.5	7.28	13.1	29.3	3.40	6.08
LSTM	97.2	62.4	75.1	77.1	53.9	63.5	51.0	32.8	39.9
LSTM-CRF	97.3	67.2	79.5	**78.0**	60.2	68.0	**54.4**	37.6	44.4
LSTM-CRF-ILP$_1$	93.4	81.4	86.9	74.1	71.1	72.6	49.6	43.3	46.2
LSTM-CRF-ILP$_{multi}$	93.2	**81.9**	**87.2**	74.0	**71.5**	**72.7**	49.5	**43.5**	**46.3**

Table 2: Overall system performance of automatic evaluations. (%)

this strategy may detect multiple events for a sentence, contributing to the increase of recalls, and F1 scores at the spent of a little drop of precisions.

4.5 Manual Evaluations

Manual Annotations

We randomly sample 150 unlabeled sentences from test data set. Annotators are asked to fully annotate the events and arguments to each sentence following steps below:

1. First, determine whether a given sentence is positive or negative, in other words, whether there are events in the sentence or not.
2. Second, assign event types to the positive sentences identified in first step.
3. Finally, label all related arguments and their roles according to the types of events in the positive sentences.

To make the annotation more credible, each sentence is independently annotated by two annotators, and the inter-annotator agreement is % for event types and % for arguments.

Results

Table 3 presents the average results of manual evaluations where we measure precision, recall and F1 by the same standards of correctness as automatic evaluation.

We can draw similar conclusions about the comparison of performances between different models as automatic evaluation. We demonstrate that LSTM-CRF-ILP$_{multi}$ is the most effective model in event extraction as it attains the highest F1 score in both manual and automatic evaluation.

Moreover, manual evaluation helps us to gain a deep insight of our generated data and proposed models. We further conduct automatic evaluation on this manual annotated dataset and list the top 5 event types whose F1 scores of LSTM-CRF-ILP$_{multi}$ differ greatly from automatic evaluation in Table 4.

Most of the performance differences blame on the stage of data generation. Figure 3 examples two types of errors in data generation. Some of the sentences automatic generated test set are noisy, in other words, they do not express any event while still match all key properties of certain instances. Take S5 as an example, though the phrases *the car* matches a film name, it does not indicate this film, and there is no explicit evidence expressing that an actor starring in a film. This is a bottleneck of our data generation strategy. During manual evaluation, we find 16 negative sentences and all of them are mistakenly labeled due to the same reason. Unfortunately, our model fails to identify some of these negative sentences.

S5: That night, in an apparent bid to kill Amos, the car instead runs over the sheriff, leaving Chief Deputy Wade Parent (played by James Brolin) in charge.		
(Wrong labeled in data generation)		
Event Type		film.performance
Arguments	actor	*James Brolin*
	character	*Wade Parent*
	film	*the car*
S6: Nicholas Hammond (born May 15, 1950) is an American actor and writer who is perhaps best known for his roles as Friedrich von Trapp in the film The Sound of Music, and as Peter Parker/Spider-Man on the CBS television series The Amazing Spider-Man.		
Event Type		film.performance
Arguments	actor	*Nicholas Hammond*
	character	*Friedrich von Trapp*
	film	*The Sound of Music*
(Missing in generated data)		
Event Type		tv.regular_tv_appearance
Arguments	actor	*Nicholas Hammond*
	character	*Peter Parker/Spider-Man*
	series	*The Amazing Spider-Man*

Figure 3: Example outputs of LSTM-CRF-ILP$_{multi}$.

Remarkably, our LSTM-CRF-ILP$_{multi}$ model can help find more CVT instances that not referenced in Freebase. There are two events mentioned in S6, while the arguments of the second event do not match any CVT instances in Freebase, leading to an omitting event in data generation. This phenomenon suggests that learning from distant supervision provided by Freebase, our model can help complete and update properties of Freebase instances in return.

Acknowledgments

References

[Berger *et al.*, 1996] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

[2]The full name is olympics.olympic_medal_honor in Freebase.

[3]The full name is tv.regular_tv_appearance in Freebase.

[4]The full name is film.film_regional_release_date in Freebase.

Model	Event Classification			Argument Detection			Event Detection		
	P	R	F	P	R	F	P	R	F
CRF	88.9	12.0	21.2	56.7	7.56	13.3	22.2	3.0	5.3
MaxEnt	**92.9**	9.78	17.7	61.7	6.44	11.7	28.6	3.01	5.44
LSTM	91.3	71.4	80.2	70.9	60.2	65.1	48.1	37.6	42.2
LSTM-CRF	89.3	75.2	81.6	**72.8**	64.9	68.6	**48.2**	40.1	44.1
LSTM-CRF-ILP$_1$	85.1	**85.7**	85.4	67.6	72.9	70.2	44.0	44.4	44.2
LSTM-CRF-ILP$_{multi}$	85.6	86.5	**85.5**	67.4	**73.6**	**70.4**	44.1	**45.1**	**44.6**

Table 3: Average of overall system performance of manual evaluations. (%)

Event type	P	R	F
olympics.medal_honor [2]	$\downarrow$ 25.0%	$\downarrow$ 5.0%	$\downarrow$ 13.8%
film.performance	$\downarrow$ 21.4%	$\uparrow$ 3.1%	$\downarrow$10.3%
business.acquisition	$\rightarrow$	$\downarrow$ 7.1%	$\downarrow$ 5.4%
tv.appearance [3]	$\downarrow$ 9.5%	$\uparrow$ 3.0%	$\downarrow$ 3.1%
film.release [4]	$\downarrow$ 7.7%	$\uparrow$ 5.6%	$\downarrow$ 0.55%

Table 4: Top 5 event types whose performances on event classification differ most from automatic evaluation. The model we evaluated is LSTM-CRF-ILP$_{multi}$

[Bottou, 2010] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[Kudo, 2005] Taku Kudo. Crf++: Yet another crf toolkit. *Software available at http://crfpp. sourceforge. net*, 2005.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

[Manning *et al.*, 2014] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

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

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