Incorporating Event Type Priori and Argument Relationship for Event Extraction

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Event extraction is a particularly challenging task of information extraction. Most previous work rely on patterns for event type priori, or use a bunch of local and global features for the identification and classification process of triggers and arguments. However, event patterns suffer the low recall issue since the real world events have a large variety of representations. In addition, previous work only considers the features of each argument while ignoring the relation between arguments. In this paper, we propose a **R**egularization-**B**ased model to make full use of the relation between arguments and trained a SVM model to make better use of **E**vent **T**ype priori (RBET method). Experiments show that we achieved a rather good result than the current state-of-art work.

1 Introduction

Event extraction has become a popular research topic in the area of information extraction. ACE 2005 defines the event extraction task[1] as three sub-tasks: identifying the trigger of an event, identifying the arguments of the event and distinguishing their corresponding roles. As an example in Figure 1, there is an "Attack" event triggered by "tear through" with four arguments, each argument has one role.

Previous work can be classified into two folds: (1) pipelined models using event patterns as well as local features (2) the joint models using both local and global features

The first kind of approaches (Grishman et al., 2005; Ji and Grishman, 2008; Liao and Grishman, 2010; Huang and Riloff, 2012) heavily rely on event patterns. Specifically, event patterns are used as event type priori for identifying event trigger and arguments. However, event patterns suffer the low recall issue since the real world events have a large variety of representations. We have carefully studied a pipeline system namely JET[2] which was proposed by Grishman et al. (2005) and followed by Ji and Grishman (2008), Liao and Grishman (2010) and Huang and Riloff (2012). The experimental results show that the recall of trigger identification and classification is relatively low, about 50%∼60%. The main reason is that most of the missed triggers cannot be matched by any pattern. The missing match problem of event patterns will lead errors of the downstream modules (e.g. argument identification and classification). To solve the problem, we trained a classifier to assign the event a type without solely relying on event patterns.

The other kind of approach (Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013; Lu and Roth, 2012) tried to identify both event trigger and argument in joint ways by incorporating global features (or distributional features). The joint models reduce the cascading errors in the pipeline system. However, both the pipeline systems and joint models identify each candidate argument separately without considering the relation among arguments. The relation here means that how possible that two candidate arguments belong to the same event. Ignoring the relation of arguments mainly lead to the following two issues: (1) Some arguments are not identified, although they have relations with other identified arguments. As the example shown in Figure 1, the underlined arguments are the ground truth, and the arguments circled by dashed line are the identification result of JET. We can see that JET missed "a waiting shed". The entity "a waiting shed" has common dependency governor with "a powerful bomb", so when the latter entity is identified

[1] http://www.itl.nist.gov/iad/mig/tests/ace/2005/

[2] http://cs.nyu.edu/grishman/jet/jet.html

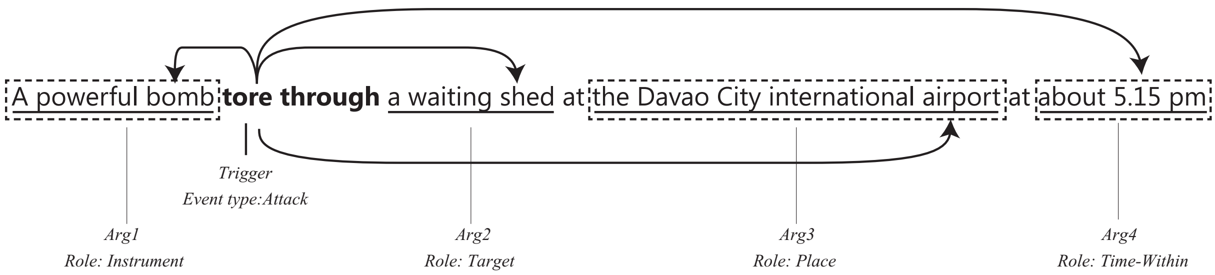


Figure 1: Event example: There is an event trigger by "tear through" with four arguments

as argument, the former should be more likely to be identified. (2) Some arguments are incorrectly identified even they are irrelevant to the event. We have carefully studied the events in which the arguments are wrongly identified. Among the 641 arguments identified by JET on ACE 2005 testing corpus, 106 arguments were incorrectly identified. The connections between the correctly identified arguments and the incorrectly identified arguments are very weak. This means that they cannot be identified simultaneously if argument relations can be leveraged.

In summary, we first train an event type classifier to predict the event type to address the missing match problem of event patterns. Second, we use a regularization method to model the relation arguments to improve the argument identification. Our approach is named RBET.

The contribution of this paper is as follows:

- We implemented an event type classifier for incorporating event type priori to event extraction. If no patterns can be matched, the classifier can predict the event type based on the local features in the sentence, which helps to identify and classify the arguments.
- We proposed a regularization method in order to make full use of the relation between candidate arguments. The regularization method improves the performance of argument identification.

2 Related Work

Event extraction is very important in the knowledge mining field. An event consists of a trigger and several arguments. To extract an event is to identify the trigger and arguments from raw text, then label the arguments with correct role, and finally classify the event into the correct event type. Previous works can be classified to two kinds of methods: (1) pipelined model using event patterns as well as local features (2) methods using local and global features.

Many classical work focus on local text, and use pattern based method for event type priori (Kim and Moldovan, 1993; Riloff and others, 1993; Soderland et al., 1995; Huffman, 1996; Freitag, 1998b; Ciravegna and others, 2001; Califf and Mooney, 2003; Riloff, 1996; Riloff et al., 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005; Grishman et al., 2005; Ji and Grishman, 2008; Liao and Grishman, 2010; Huang and Riloff, 2012), others use local feature based classification method (Freitag, 1998a; Chieu and Ng, 2002; Finn and Kushmerick, 2004; Li et al., 2005; Yu et al., 2005). Moreover, a variety of techniques have been explored for weakly supervised training (pattern-based and rule-based) of event extraction systems (Riloff, 1996; Riloff et al., 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005; Patwardhan and Riloff, 2007; Chambers and Jurafsky, 2011). In some of these systems, human should help to delete some nonsense patterns or rules. (Shinyama and Sekine, 2006; Sekine, 2006) are unsupervised methods to extract patterns from open domain texts. However, pattern is not always enough although some method (Huang and Riloff, 2012; Liu and Strzalkowski, 2012) use bootstrapping to get more patterns.

Some other methods (Gu and Cercone, 2006; Patwardhan and Riloff, 2009) considered broader context when deciding the role fillers. Other systems take the whole discourse feature into consideration, such as (Maslennikov and Chua, 2007; Liao and Grishman, 2010; Hong et al., 2011; Huang and Riloff, 2011). Ji and Grishman (2008) even considers the topic-related documents, so they proposed the cross-document method. Liao and Grishman (2010; Hong et al. (2011) use a series of global features (for example, the occurence of one event type lead to the occurence of another) to improve the role assignment and event classification performance.

Joint models (Li et al., 2013; Lu and Roth, 2012) is also an important contribution. Li et al. (2013) makes full use of the local feature and global feature to get a better result. The semi-CRF based method (Lu and Roth, 2012) trains separate models for each event type, which requires a lot of training data. However, all of these above methods considers arguments separately while ignoring the relation between arguments.

In addition, Ritter et al. (2012), Zhou et al. (2014) and Zhou et al. (2015) used unsupervised method to extract events from Twitter.

In summary, most of the above works are strongly depended on patterns which will bring severe loss of recall. In addition, the arguments are considered independently while the relation between arguments is also important for argument identification. We implement an event type classifier and designed a regularization method to solve the two problems.

3 ACE Event Extraction Task

Automatic Content Extraction (ACE) is an event extraction task. It annotates 8 types and 33 subtypes of events. ACE defines the following terminologies:

- Entity: an object or a set of objects in one of the semantic categories of interest
- Entity mention: a reference to an entity, usually a noun phrase (NP)
- Event trigger: the main word which most clearly expresses an event occurrence
- Event arguments: the entity mentions that are involved in an event
- Argument roles: the relation of arguments to the event where they participate, there are 35 roles in total
- Event mention: a phrase or sentence within which an event is described, including trigger and arguments

Given an English document, an event extraction system should identify event triggers with their subtypes and their arguments from each sentence. An example is shown in Figure 1, there is an "Attack" event triggered by "tear through" with four arguments, each argument has a role type such as "Instrument", "Target", etc.

For the evaluation metric, we follow the previous works (Ji and Grishman, 2008; Liao and Grishman, 2010; Li et al., 2013). We use the following criteria to determine the correctness of the predicted event mentions.

- A trigger is considered to be correct if and only if its event type and offsets can match the reference trigger;
- An argument is correctly identified if and only if its event type and offsets can match any of the reference arguments;
- An argument is correctly identified and classified if and only if its event type, offsets and role can match any of the reference arguments.

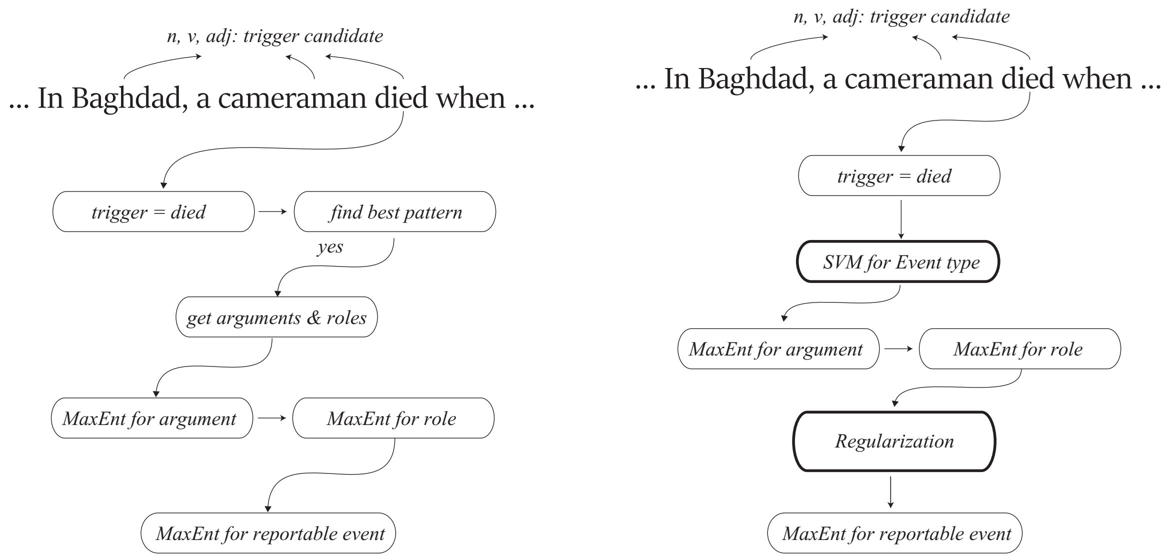
4 Baseline: JET Extractor for Events

Many previous works take JET as their baseline system, including Ji and Grishman (2008), Liao and Grishman (2010), Li et al. (2013). JET extracts events independently for each sentence. This system combines pattern matching with statistical model together. For each event mention in the training corpus of ACE, the patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. After that, three Maximum Entropy classifier are trained using the local features as well as the pattern match features:

- Argument Classifier: to distinguish arguments from non-arguments
- Role Classifier: to classify arguments by argument role
- Reportable-Event Classifier: to determine whether there is a reportable event mention according to the trigger, event type, and a set of arguments

The features the three classifier used are listed in Table 1.

Figure 2(a) shows the whole test procedure. In the test procedure, each sentence is scanned for nouns, verbs and adjectives as trigger candidates. When a trigger candidate is found, the system tries to match the context of the trigger against the set of patterns associated with that trigger. If this pattern matching process is successful, the best pattern will assign some of the mentions in the sentence as the arguments of a potential event mention. For the remaining mentions in the sentence, the argument classifier is applied; for any argument pass-



(a) The flow chart of JET

(b) The flow chart of our approach

Figure 2: The left is the flow chart of JET, the right is the flow chart of our approach, the thick line block is our contribution

	Origin	Added
Arg	Trigger, EvType, MentionType, argument, EvTypeAndArg, EvTypeAndMentionType, prevToken, prevTokenAndType, chunkPathAndType synPathEvType	Can Match best pattern
Role	Trigger, EvType, MentionType, argument, EvTypeAndArg, EvTypeAndMentionType, prevToken, prevTokenAndType, chunkPathAndType synPathEvType	Can Match best pattern
Event	Trigger, Pattern match score	EvType TriggerPOS

Table 1: The Maximum Entropy Classifier features

ing that classifier, a role is assigned to it using the role classifier. Finally, once all arguments have been assigned, the reportable-event classifier is applied to decide whether this event mention should be reported.

5 Motivations

5.1 Error Analysis

We have carefully studied the JET system. After carefully analysis the errors of JET, we found that they can be attributed to two types of errors. First, the pattern set is not always enough for so large variety of events, so that the event type priori cannot be fully utilized, which could lead to severe recall loss. Second, the arguments are considered independently, while the relation between candidate arguments is also very important to the

		Trigger Classification	Argument Classification	Role Classification
JET	P	67.56	46.45	41.02
	R	53.54	37.15	32.81
	F	59.74	41.29	36.46

Table 2: Overall performance of JET on blind test data

argument identification.

5.1.1 Trigger Error

The performance of JET is shown in Table 2. According to the table, we found that the recall of "Trigger Classification" is only 53.54%, which means about 46.46% event mentions are lost. We randomly sample 40 documents in ACE corpus and use the origin JET system to extract events for statistical analysis. According to statistics, of all the lost event mentions, about 96.3% are due to the missing of corresponding event pattern, while the last 3.7% have a matched pattern but the event type is wrong.

The error in the trigger identification and classification can affect the argument identification and classification severely. In the statistical result, there are in total 230 argument identification error. Among them, 124 are due to the trigger error, which took about 53.9%.

For the 96.3% triggers which do not have corresponding patterns, their event type cannot be correctly assigned. Then, the arguments cannot be correctly identified and classified. Hence, for those triggers without any matched patterns, the event type should be correctly predicted, and then this event type may help to identify and classify arguments together with other features. This method may supply the patters in some extent.

5.1.2 Argument Error

Apart from the 124 trigger-caused argument identification error, the remaining 106 errors can be split into two types of error.

- Error 1: An error argument has been identified
- Error 2: The right argument has not been identified

Intuitively, we can observe that if two entities belong to the arguments of the same event, there are always some obvious relations between them. For example, "Police have arrested four people in connection with the killings", the argument "police" and "four people" share the same father in the dependency parse tree. Certainly, the actual situation is much more complicated than this, but it is sure that the arguments of one event have relation with each other.

Corresponding to the two kinds of errors, we should capture two kinds of argument relationship. (1) Two arguments should occur in the same event; (2) Two argument cannot occur in the same event. For the first relationship, if one argument is identified, then the other is more likely to be identified. For the second relationship, if one argument is identified, then the other cannot be identified. Intuitively, with these two kinds of relationship, the argument identify performance will improve.

6 Approach Overview

Based on the above analysis, we decide to make two improvements: (1) make better use of event type priori: if there is no corresponding pattern, instead of cast it away immediately, we use a classifier to assign the current trigger an event type and assign some of the entity mentions as its arguments, (2) use a regularization method to make full use of the relation of arguments.

The thick line blocks in Figure 2(b) depicts our improvements. Whether or not any pattern can be matched, we identify and classify the arguments directly by the maximum entropy classifier instead of by patterns first. The best matched pattern and the event type we got from the classifier can be taken as a feature in the following maximum entropy classifiers. After the outputs of argument and role classifier are calculated, we make use of the argument relationship to regularize for a better result. In addition, we add some new features to the origin maximum entropy classifier as is shown in Table 1 in order to make the origin classifier more suitable for our improvement.

6.1 Incorporating Event Type Priori

Event type is very important priori information for argument identification and role classification, not only the event type is an important feature, the corresponding event schema can also help the argument identification and role classification a lot. We would like to assign an event type to the given trigger based on the sentence it is located.

We use WORD2VEC[3] for an embedding representation of common words, the word vectors are trained on the default "text8" training text data. Each word vector is 200-dim.

The feature of the event type classifier is a 400-dim vector, which is concatenated by two word vectors. The first is the trigger's word vector. The second is denoted as $Aver(NP)$ and can be calculated by Eq 1, which represents the information of all the candidate arguments.

$$Aver(NP) = \frac{1}{n}\sum_i Vec(Head(NP_i)) \quad (1)$$

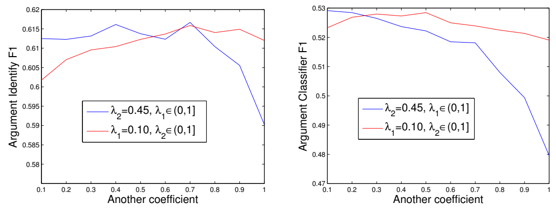
Here, the $Vec(\cdot)$ means a word's vector representation. $Head(\cdot)$ means the head word of an NP (for example, "a little girl" has a head word of "girl").

We trained a SVM classifier using the training documents, the classifying precision on the test documents achieved 80.6%. We have carefully checked the wrong cases. We found that the meanings of some event types are rather similar like "Transfer-Ownership" and "Transfer-Money", which caused many errors. Hence, we did not continue to improve this classifier.

6.2 Capture the Relationship Between Arguments

In this section, we will capture two kinds of relation between arguments described in Section 5.1.2. For a trigger, if there are n candidate arguments, we set a $n \times n$ matrix C to represent the relationship between arguments. If $C_{i,j} = 1$, then argument i and argument j should belong to

[3] http://code.google.com/p/word2vec/



(a) Arg Identify (b) Arg Classify

Figure 3: The trend graph when fix one coefficient and change another

the same event. If $C_{i,j} = -1$, then argument i and argument j cannot belong to the same event.

We use a n-dim vector X to represent the identification result of arguments. Each entry of X is 0 or 1, 0 represents "noArg", 1 represents "arg". The evaluation of X should be calculated by Eq 2. The larger $E(X)$ is, the better X will be.

$$E(X) = \lambda_1 X^T C X + \lambda_2 P_{sum}^{arg} + (1 - \lambda_1 - \lambda_2) P_{sum}^{role} \quad (2)$$

Here, $X^T C X$ means to add up all the relationship value if the two arguments are identified. Hence, the more the identified arguments are related, the larger the value $X^T C X$ is. P_{sum}^{arg} is the sum of all chosen arguments probability. The probability here is the output of the arguments' maximum entropy classifier. P_{sum}^{role} is the sum of all the classified roles' probability. The probability here is the output of the roles' maximum entropy classifier. We tuned the coefficients λ_1 and λ_2 on the development set, and finally we set $\lambda_1 = 0.10$ and $\lambda_2 = 0.45$. Figure 3 shows the variation of argument identification's F1 measure and argument classification's F1 measure when fix one parameter and change another. Note that the third coefficient $1 - \lambda_1 - \lambda_2$ must be positive, which is the reason why the curve decreases sharply when λ_2 is fixed and $\lambda_1 > 0.65$.

Eq 2 means that, while we should identify and classify the candidate arguments with larger probability, the argument relationship should also as much as possible. The arguments should also follow the following constraints. These constraints together with Eq 2 can make the argument identification and classification help each other for a better result.

- Each entity can only take one role
- Each role can belong to one or more entities
- The role assignment must follow the event schema of the corresponding type

We use Beam Search method to search for the optimal assignment X.

6.2.1 Training the Argument Relationship Matrix

The argument relationship matrix is very important in the regularization process. We trained a maximum entropy classifier to predict the connection between two entities. The entity pairs in the ground truth events are used for our training data. After a large amount of trials, we choose the following features:

- TRIGGER: the trigger of the event
- ENTITY DISTANCE: the distance between the two candidate arguments in the sentence
- Whether the two candidate arguments occur on the same side of the trigger
- PARENT DEPENDENCY DISTANCE: the distance between the two candidate arguments' parents in the dependency parse tree
- PARENT POS: if the two candidate arguments share the same parent, take the common parent's POS tag as a feature
- Whether the two candidate arguments occur on the same side of the common parent if the two candidate arguments share the same parent

For an entity pair, if both of the entities belong to the same event's arguments, we take it as positive example. For each positive example, we randomly exchange one of the entities with an irrelevant entity (an irrelevant entity is in the same sentence with the event, but it is not the event's argument) to get a negative example. In the testing procedure, we predict the relationship between entity i and entity j using the maximum entropy classifier. Since in many cases, the relations may be very obscure and difficult to be figure out. So we only capture a portion of them. We set two thresholds, if the output of the maximum entropy classifier is larger than 0.8, we set $C_{i,j} = 1$, if the output is lower than 0.2, we set $C_{i,j} = -1$.

7 Experiments

7.1 Data

We utilize ACE 2005 data sets as our testbed. As is consistent with previous work, we randomly select 10 newswire texts from ACE 2005 training corpora as our development set, and then conduct blind test on a separate set of 40 ACE 2005

Abbreviation	Illustration
JET	The within-one-sentence baseline of Grishman et al. (2005)
CD	Cross-Document (Ji and Grishman, 2008)
CEV	Cross-Event (Liao and Grishman, 2010)
CEN	Cross-Entity (Hong et al., 2011)
Joint	Joint model in (Li et al., 2013)
SP	Unsupervised Structured Preference method in (Lu and Roth, 2012)
semi-CRF	Supervised CRF in (Lu and Roth, 2012)
RBET	Our approach with both event type classifier and regularization
RBET-Regu	Our approach without regularization
RBET-ET	Our approach without event type classifier

Table 3: Baseline illustration

Method	Trigger F1	Arg id F1	Arg id+cl F1
JET	59.7	42.5	36.6
CD	67.3	46.2	42.6
Joint	65.6	-	41.8
RBET	66.0	**55.4**	**43.8**
RBET-Regu	66.0	51.8	42.7
RBET-ET	64.8	54.6	43.0

Table 5: Overall Performance with predicted entities, timex, and values

newswire texts. The rest of ACE training corpus is used as the training data.

7.2 Baseline System

The abbreviation and simple illustration of the baselines as well as our approach are listed in Table 3. The JET system is the within-one-sentence baseline proposed by (Grishman et al., 2005). The CD, CEV and CEN are all extension systems of the within-one-sentence baseline. The joint model (Li et al., 2013) is the currently state-of-art method. Among these methods, CEV and CEN make use of the gold-standard entities, timex, and values annotated in the corpus as the argument candidates. CD uses the JET system to extract the candidate arguments. Li et al. (2013) reports the performance with both gold-standard argument candidates and predicted argument candidates. Therefore, we compared our result with methods based on gold argument candidates in Table 4 and methods based on predicted argument candidates in Table 5. The structured preference method and semi-CRF method are proposed by (Lu and Roth, 2012). Since the semi-CRF requires large amount of training data, the author chose to perform their evaluations on the top 4 events ("Attack", "Meet", "Die" and "Transport") which have the most instances. We compared our result with them in Table 6.

7.3 Overall Performance

We conducted experiments to answer the following questions. (1) Can the event type classifier lead to a higher recall in trigger classification, argument identification and classification while retaining the precision value? (2) Can the regularization step improve the performance of argument identification and classification?

Table 4 shows the overall performance on the blind test set. We compared our result with the JET baseline as well as the CEV, CEN and joint methods. When added the event type classifier, in the line named "RBET-Regu", we gain a significant increase in the three measures over the JET baseline in recall. Although our trigger's precision is lower than JET, it gains 6.1% improvement on trigger's F1 measure, 19.5% improvement on argument identification's F1 measure and 11.2% improvement on argument classification's F1 measure. In the predicted argument candidate situation in Table 5, our approach "RBET-Regu" again significantly outperforms the JET baseline. Remarkably, our result is comparable with Joint model although we only use local features. The line with name "RBET-ET" in Table 4 and Table 5 represents the performance when we only use regularization method. In Table 4, Compared to the four baseline systems, the argument identification's F1 measure of "RBET-ET" is significantly higher. In Table 5, the "RBET-ET" again gains a higher F1 measure than the JET, CD, joint model baseline and "RBET-Regu".

The complete approach is denoted as "RBET" in Table 4 and Table 5. Remarkably, our approach performances comparable in trigger classification with the state-of art methods: CD, CEV, CEN, Joint model, and significantly higher than them in argument identification as well as classification although we did not use the cross-document, cross-event information or any global feature. This may prove that our assume that the relationship between argument candidates can help to improve the argument identification performance is right. The event type classifier also contributes a lot in trigger identification & classification. We have done the Wilcoxon Signed Rank Test on trigger classification, argument identification and argument classification, all the three have $p < 0.01$.

The comparison between our approach and Lu and Roth (2012)'s methods is listed in Table 6. The numbers in the table is the micro average F1

Method	Trigger Classification			Argument Identification			Argument Role		
	P	R	F1	P	R	F1	P	R	F1
JET	67.6	53.5	59.7	46.5	37.2	41.3	41.0	32.8	36.5
CEV	68.7	68.9	68.8	50.9	49.7	50.3	45.1	44.1	44.6
CEN	72.9	64.3	68.3	53.4	52.9	53.1	51.6	45.5	48.3
Joint	73.7	62.3	67.5	69.8	47.9	56.8	64.7	44.4	52.7
RBET	67.2	64.7	65.9	63.2	59.4	**61.2**	54.1	53.5	**53.8**
RBET-Regu	65.7	65.9	65.8	60.6	61.1	60.8	47.2	48.3	47.7
RBET-ET	67.2	61.7	64.3	63.6	57.6	60.5	51.6	47.4	49.4

Table 4: Overall Performance with gold-standard entities, timex, and values

Event	SP	semi-CRF	Our Approach		
			RBET-Regu	RBET-ET	RBET
Attack	42.02	63.11	66.84	69.55	70.36
Meet	63.55	76.64	70.89	71.25	73.04
Die	55.38	67.65	65.56	66.63	68.50
Transport	57.29	64.19	63.72	65.44	67.67

Table 6: Performance compared to (Lu and Roth, 2012) with gold-standard entities, timex, and values

measure of argument classification. We did not train separate models for each event type so that the performance of "Meet" event is lower than semi-CRF but we still outperforms the semi-CRF method in "Attack", "Die" and "Transport" events.

However, our approach is just a pipeline approach which suffers from error propagation and the argument performance may not affect the trigger too much. We can see from Table 4, although we use the gold argument candidates, the trigger performance is still lower than CEV, CEN and joint model. Another limitation is that our regularization method did not improve the argument classification too much since it only uses constraints to affect roles. Future work may be done to solve these two limitations.

8 Conclusion

In this paper, we proposed two improvements based on the event extraction baseline JET. We found that JET depends too much on event patterns for event type priori and JET considers each candidate argument separately. However, patterns cannot cover all events and the relationship between candidate arguments may help when identifying arguments. For a trigger, if no pattern can be matched, the event type cannot be assigned and the arguments cannot be correctly identified and classified. Therefore, we developed an event type classifier to assign the event a type, then identify and classify the arguments using the type information as well as other features.

On the other hand, we trained a maximum entropy classifier to predict the relationship between candidate arguments. Then we proposed a regularization method to make full use of the argument relationship. The experiment result shows that the regularization method brings a significant result in the argument identification over previous works.

In summary, by using the event type classifier and the regularization method, we have achieved a good performance in which the trigger classification is comparable to the state-of-the-art methods, and the argument identification & classification performance is significantly better than state-of-the-art methods. However, we only use the sentence-level features and our method is a pipelined approach. Also, the argument classification seems not to be affected too much by the regularization method. Future work may be done to integrate our method into a joint approach, use some global feature and try to improve the argument classification by the regularization, which may make our performance better.

References

Mary Elaine Califf and Raymond J Mooney. 2003. Bottom-up relational learning of pattern matching rules for information extraction. *The Journal of Machine Learning Research*, 4:177–210.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 976–986. Association for Computational Linguistics.

Hai Leong Chieu and Hwee Tou Ng. 2002. A maximum entropy approach to information extraction from semi-structured and free text. *AAAI/IAAI*, 2002:786–791.

Fabio Ciravegna et al. 2001. Adaptive information extraction from text by rule induction and generalisation. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 1251–1256. LAWRENCE ERLBAUM ASSOCIATES LTD.

Aidan Finn and Nicholas Kushmerick. 2004. *Multi-level boundary classification for information extraction*. Springer.

Dayne Freitag. 1998a. Multistrategy learning for information extraction. In *ICML*, pages 161–169.

Dayne Freitag. 1998b. Toward general-purpose learning for information extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 404–408. Association for Computational Linguistics.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyus english ace 2005 system description. *ACE*, 5.

Zhenmei Gu and Nick Cercone. 2006. Segment-based hidden markov models for information extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 481–488. Association for Computational Linguistics.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1127–1136. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: detecting event role fillers in secondary contexts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1137–1147. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295. Association for Computational Linguistics.

Scott B Huffman. 1996. Learning information extraction patterns from examples. In *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260. Springer.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the 46st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 254–262.

Jun-Tae Kim and Dan I Moldovan. 1993. Acquisition of semantic patterns for information extraction from corpora. In *Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on*, pages 171–176. IEEE.

Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. 2005. Using uneven margins svm and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 72–79. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria, August. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797. Association for Computational Linguistics.

Ting Liu and Tomek Strzalkowski. 2012. Bootstrapping events and relations from text. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 296–305. Association for Computational Linguistics.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 835–844. Association for Computational Linguistics.

Mstislav Maslennikov and Tat-Seng Chua. 2007. A multi-resolution framework for information extraction from free text. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 592. Citeseer.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *EMNLP-CoNLL*, volume 7, pages 717–727. Citeseer.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 151–160. Association for Computational Linguistics.

Ellen Riloff et al. 1993. Automatically constructing a dictionary for information extraction tasks. In *AAAI*, pages 811–816.

Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Alan Ritter, Oren Etzioni, Sam Clark, et al. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. Crystal: Inducing a conceptual dictionary. *arXiv preprint cmp-lg/9505020*.

Mark Stevenson and Mark A Greenwood. 2005. A semantic approach to ie pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 379–386. Association for Computational Linguistics.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 224–231. Association for Computational Linguistics.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 940–946. Association for Computational Linguistics.

Kun Yu, Gang Guan, and Ming Zhou. 2005. Resume information extraction with cascaded hybrid model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 499–506. Association for Computational Linguistics.

Deyu Zhou, Liangyu Chen, and Yulan He. 2014. A simple bayesian modelling approach to event extraction from twitter. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 700–705.

Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.