Joint Modeling for Chinese Event Extraction with Rich Linguistic Features

Chen CHEN   Vincent NG
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688, USA
{yzcchen,vince}@hlt.utdallas.edu

Abstract

Compared to the amount of research that has been done on English event extraction, there exists relatively little work on Chinese event extraction. We seek to push the frontiers of supervised Chinese event extraction research by proposing two extension to Li et al.'s (2012) state-of-the-art event extraction system. First, we employ a joint modeling approach to event extraction, aiming to address the error propagation problem inherent in Li et al.'s pipeline system architecture. Second, we investigate a variety of rich knowledge sources for Chinese event extraction that encode knowledge ranging from the character level to the discourse level. Experimental results on the ACE 2005 dataset show that our joint-modeling, knowledge-rich approach significantly outperforms Li et al.'s approach.

Title and Abstract in Chinese

运用丰富语言学特征的中文事件抽取联合模型

与英文的事件抽取研究相比，对于中文的事件抽取研究工作相对较少。在Li et al.(2012)的基于监督学习的事件抽取系统基础上，我们提出了两个扩展以进一步推动中文事件抽取的研究。首先，我们使用了一个联合模型，以解决Li et al.管道式系统中的错误传播问题。其次，针对中文信息抽取，我们研究了一系列从字符层面到文章层面的特征。在ACE2005数据上的实验结果表明，我们运用丰富语言学特征的联合模型显著地优于Li et al.的方法。

Keywords: event extraction, Chinese language processing.

Keywords in Chinese: 信息抽取, 中文自然语言处理.

1 Introduction

Recent years have seen a surge of interest in automatically extracting events from textual documents. While diverse types of event extraction have been examined in the literature, the one we will focus on in this paper is ACE event extraction, which involves extracting instances of a predefined event type from documents. For example, consider the following Chinese text segment:

Resneft 收购尤甘斯克付出了仅 93.5 亿美元
(Resneft acquired Yugansk, paying only 9.35 billion US dollars)

When applied to this example, an ACE event extraction system for Chinese should identify one event instance, which (1) is *triggered* by the verb 收购 [acquired] whose type is TRANSFER-MONEY, and (2) has three *arguments*, Resneft, 尤甘斯克 [Yugansk] and 93.5 亿美元 [9.35 billion], which fulfill the roles of BUYER, ARTIFACT and PRICE, respectively.

This example illustrates the four primary subtasks of an ACE event extraction system, namely, (1) *trigger identification* (e.g., 收购 [acquired] should be identified as the trigger of an event); (2) *trigger type determination* (e.g., 收购 [acquired] should be assigned the type TRANSFER-MONEY; (3) *argument identification* (e.g., Resneft, 尤甘斯克 [Yugansk] and 93.5 亿美元 [9.35 billion] are the arguments of this TRANSFER-MONEY event); and (4) *argument role determination* (e.g., Resneft, 尤甘斯克 [Yugansk] and 93.5 亿美元 [9.35 billion] play the roles of BUYER, ARTIFACT and PRICE respectively in this event).

Compared to the amount of research on English event extraction (e.g., Finkel et al. (2005), Grishman et al. (2005), Ahn (2006), Hardy et al. (2006), Maslennikov and Chua (2007), Ji and Grishman (2008), Patwardhan and Riloff (2009), Liao and Grishman (2010), Hong et al. (2011)), there is considerably less work on Chinese event extraction. Work on end-to-end Chinese event extraction was pioneered by Chen and Ji (2009b), who adopt a pipeline system architecture composed of four components that correspond to the four major subtasks mentioned above. More specifically, in *training*, they learn a classifier to perform each of the four subtasks independently using primarily lexico-syntactic features but also a couple of semantic features (see Section 3 for an overview of these features); and in *testing*, they feed a raw document through the pipeline of components where the output of one component is the input of the subsequent one. Li et al.'s Chinese event extraction system also employs a pipeline architecture, but aims to improve the first component, trigger identification, via two techniques, namely compositional semantics and discourse consistency. They show that with these two techniques, their system substantially outperforms Chen and Ji's system, achieving state-of-the-art results.

Our goal in this paper is to improve the state of the art in Chinese event extraction. Specifically, we take Li et al.'s event extraction system as a baseline, and investigate two extensions to their system. Our first extension is a machine learning extension where we employ *joint learning* for event extraction. As is commonly known in the natural language processing (NLP) community, a pipeline architecture, such as the one adopted by Chen and Ji and Li et al. , suffers from the *error propagation* problem, where the errors made by an upstream component will propagate to and could adversely affect the performance of a downstream component. We address this problem by recasting event extraction as two *joint learning* tasks where we (1) jointly learn trigger identification and trigger type determination and (2) jointly learn argument identification and argument role determination.[1]

[1] A natural alternative would be to jointly learn the four subtasks. Though possible, this would substantially increase the complexity of the learning task. In fact, our preliminary experiments indicate that this alternative yields inferior results to our way of applying joint learning to event extraction and therefore will not be pursued in this paper.

Our second extension is linguistic extension where we employ a *knowledge-rich* approach, investigating a variety of knowledge sources for Chinese event extraction. In this extension, not only do we propose more effective use of existing features such as character-based features, but we also investigate novel features that exploit results of zero pronoun resolution and noun phrase (NP) coreference resolution, as well as features that exploit trigger probability and trigger type consistency (see Section 3 for details). The strength of our linguistic extension stems in part from the richness in the variety of features it considers: these features capture linguistic information ranging from the character level to the discourse level, and exploit Chinese-specific phenomena such as the presence of zero pronouns.

We evaluate our approach on the ACE 2005 Chinese event extraction task, which involves identifying event instances that belong to one of 33 predefined event types. Unlike previous work (Chen and Ji, 2009; Li et al., 2012), which reserves only 10% of the annotated data for testing and uses the rest for training, we provide a more robust evaluation of our system via performing 10-fold cross-validation experiments. We discover that the F-scores achieved on different folds can vary by as many as 10 percentage points for all four subtasks. The sensitivity of the system performance can be attributed in part to the small size of the ACE 2005 dataset, which is composed of only 633 documents, suggesting that cross validation is needed to more accurately reveal the performance of an event extraction system when evaluated on this dataset. Overall, our experimental results demonstrate that our joint-learning, knowledge-rich approach substantially improves Li et al.'s system, suggesting that (1) joint learning offers benefits over the pipeline approach; (2) all but the coreference features improve performance; and (3) while each of our features provide small gains, their cumulative benefits are substantial.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, we provide an overview of our baseline Chinese event extraction system. Sections 4 and 5 describe our machine learning extension and our linguistic extension to the baseline system, respectively. We present evaluation results in Section 6 and conclude in Section 7.

2 Related Work

Much attention has been devoted to the event extraction task in the NLP community. In the early years, researchers focused on sentence-level extraction, employing local information from just one sentence (e.g., Grishman et al. (2005), Hardy et al. (2006), Ahn (2006)).

However, in many cases local information alone is insufficient to make the right decisions, so later work incorporates more context around a sentence and seeks high level information. For example, Gu and Cercone (2006) and Patwardhan and Riloff (2009) consider broader sentential context. Ji and Grishman (2008) extend the scope to a cluster of topic-related documents and utilize global information from related documents. Gupta and Ji (2009) employ cross-event information to extract implicit time information. Liao and Grishman (2010) leverage document-level cross-event inference; Liao and Grishman (2011a) extract topic features to improve event extraction; and Liao and Grishman (2011b) present a self-training strategy and combine it with global inference. McClosky et al. (2011) use the tree of event-argument relations in a reranking dependency parser to capture global event structure properties. Hong et al. (2011) explore entity type consistency to predict event mentions. Huang and Riloff (2012b) initially identify arguments and then include discourse properties to model textual cohesion. More recently, some researchers have tried to improve other aspects of event extractions. For example, Lu and Roth (2012) introduce a novel sequence labeling framework called structured preference modeling, and Huang and Riloff (2012a) propose a bootstrapping solution for argument extraction with little annotated data.

As far as work on Chinese event extraction is concerned, Chen and Ji (2009b) point out the Chinese-specific issue of word segmentation errors and create an errata table to alleviate this problem, analyzing the impactof different types of features. Chen and Ji (2009a) bootstrap Chinese event extraction with extra information from an English event extraction system using cross-lingual information projection. Ji (2009) extracts cross-lingual predicate clusters and uses a cross-lingual information extraction system to improve Chinese event extraction. Li et al. (2012) explore compositional semantics and discourse consistency to address the unknown trigger problem and word segmentation errors.

3 Baseline System

In order to establish a strong baseline Chinese event extraction system adopting the pipeline architecture mentioned in the introduction, we train a classifier for each of the four components by using a feature set that is the *union* of the features employed by Chen and Ji (2009b) and Li et al. (2012). We augment it with compositional semantics and discourse consistency, the two extensions proposed by Li et al. that aim to improve the trigger identification component, as described in detail in this section. Below we will discuss our implementation of each of the four components of the baseline Chinese event extraction pipeline. All classifiers are trained using the implementation of SVM available from the SVM^{multiclass} package[2]. Word segmentation, syntactic parsing, and dependency parsing are performed using Stanford's Chinese NLP and Speech Processing tool.[3].

3.1 Trigger Identification Component

Following Li et al. (2012), we employ a two-step approach to identify triggers. First, in the *extraction* step, we use heuristics to extract candidate triggers. Then, in the *pruning* step, we aim to improve precision by employing two types of pruning, namely *heuristic-based* pruning and *learning-based* pruning. Both steps are detailed below.

Extraction. To extract candidate triggers, we first follow Chen and Ji (2009b), positing a word in a test document as a candidate trigger if it appears in a training document as a (true) event trigger. Li et al. (2012) observe that this simple candidate extraction method has a low recall: it fails to extract many true triggers in a test document since many of them do not appear in the training set. To improve recall, they propose a technique to extract additional candidate antecedents based on *compositional semantics*.

The use of compositional semantics is motivated by the observation that the meaning of a Chinese word is largely determined by the meaning of its component characters. For instance, the meaning of 刺伤 [injure by stabbing] can be determined from the meaning of its component characters, 刺 [stab] and 伤 [injure]; similarly, the meaning of 撞伤 [injure by hitting] can be determined from the meaning of 撞 [hit] and 伤 [injure]. Now, assume that 撞伤 appears in a test document. If 撞伤 does not appear in the training data but 刺伤 appears as a trigger in the training data, we want to be able to infer that 撞伤 is a trigger from the fact that 刺伤 is a trigger since the two verbs both describe an "injure" event.

To be able to do this kind of inference for extracting additional candidate triggers, we employ a simple method proposed by Li et al.: (1) add all single-character verb triggers to a set (call it BV^4); (2) split all other verb triggers in the training set into characters and add each character to BV; and

[2] http://svmlight.joachims.org/svm_multiclass.html

[3] http://nlp.stanford.edu/projects/chinese-nlp.shtml

[4] Li et al. (2012) refer to these verbs as "basic verbs" (BV).

(3) posit a word in a test document as a candidate trigger if it contains an element in BV. It should be easy to see that this method can easily handle cases such as the 撞伤 example discussed above.

Heuristic-based pruning. To prune spurious triggers from the list of candidate triggers, we employ the three heuristics proposed by Li et al.: non-trigger filtering, POS filtering, and verb structure filtering. We refer the reader to their paper for details of these heuristics.

Learning-based pruning. After heuristic-based pruning, we follow Li et al. and apply learning-based pruning to further prune the candidate triggers. Specifically, we train a classifier on the training data to determine whether a candidate is a trigger or not. Since Li et al. did not specify the training instance creation method, we experiment with several methods and found that creating one training instance from each word worked best. We train the classifier using 19 linguistic features, most of which were proposed by Chen and Ji, as shown below:

- Lexical features (6): trigger word; POS of trigger word; previous word + trigger word; previous POS + trigger POS; trigger word + next word; trigger POS + next POS
- Syntactic features (5): depth of trigger word in syntax parse tree; the path from leaf node of trigger to the root in syntax parse tree; the phrase structure expanded by the father of the trigger; phrase type of the trigger; the path from the leaf node of the trigger to the governing clause
- Semantic dictionaries (2): whether trigger word exists in a predicate list from Chinese PropBank (Xue and Palmer, 2008); the entry number of the trigger in a Chinese synonym dictionary[5]
- Nearest entity information (6): entity type of the syntactically/physically nearest entity to the trigger in syntax parse tree; entity type of the syntactically/physically left/right nearest entity to the trigger in syntax parse tree + entity

Reclassifying unconfidently labeled instances. Not all trigger candidates were classified with the same confidence by the trigger identifier. For our SVM-based trigger identifier, those instances that are closer to the hyperplane are classified with less confidence than those that are farther away. Li et al. propose to improve the accuracy of trigger identification by reclassifying those instances that were not confidently classified[6], specifically by training a *discourse consistency* (DC) classifier.

Before describing the DC classifier, let us motivate DC. Consider the sentence *The talks are serious*, where *talks* is a trigger of a MEET event whose arguments are not in the same sentence as the trigger itself. Because of the absence of nearest entity information (and hence the inability to compute the nearest entity information features), Li et al. observe that the corresponding test instances were typically classified with low confidence. To address this problem, they make an observation. Given a candidate trigger word t, if many other occurrences of t in the same discourse are being classified as a trigger, then t is likely to be a trigger due to DC. Similarly, if many other occurrences of t in the same discourse are being classified as non-triggers, then t is not likely to be a trigger due to DC.

Li et al. create five linguistic features that encode this observation (see their paper for details), train a DC classifier on a feature set composed of these five features as well as the 19 features used by the trigger identifier, and use it to reclassify those instances not confidently classified by the trigger identifier. Following Li et al., we train this DC classifier on the development set, which comprises 5% of the available training data reserved solely for the purpose of training this classifier.

[5] This dictionary is created by Harbin Institute of Technology's NLP Group and is available from http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=162. The entry number can be thought of as the equivalent of the synset id in English WordNet.

[6] Following Li et al., we posit that an instance is not confidently classified if the probability associated with its classification is between 0.05 and 0.95. We obtain these probability values by converting the signed distance values returned by the SVM using a sigmoid function.

3.2 Trigger Type Determination Component

Given a word identified as a trigger by the preceding component, the trigger type determination component employs a classifier to classify a word as belonging to one of the 33 predefined subtypes. Following Li et al., we train this classifier using the same 19 linguistic features that were used to train the trigger identifier.

3.3 Argument Identification Component

Given a typed trigger produced by the trigger type determination component, the argument identification component employs a classifier to determine whether a candidate argument is its actual argument or not. The list of candidate arguments for a trigger includes all and only those entity mentions, values and time expressions that appear in the same sentence as the trigger under consideration. Hence, training instances are created by pairing each trigger with each of its candidate arguments. The class value of a training instance is either YES (if the candidate is a true argument) or NO (if it is not). Following Li et al., we train this classifier using 19 linguistic features, as shown below.

- Basic features (6): trigger subtype; type of entity mention; head word of entity mention; event subtype + head word; event subtype + entity subtype; POS of trigger word
- Neighboring words (6): left/right neighbor word of entity; left/right neighbor word of the entity + word's POS; left/right neighbor word of the trigger + word's POS
- Syntactic features (4): the phrase structure expanding the parent of the trigger in the syntactic parse tree; whether the trigger is before or after the trigger; the minimal path from the entity to the trigger in the syntactic parse tree; the shortest length of the above minimal path
- Dependency feature (1): the dependency path from the entity to the trigger

3.4 Argument Type Determination Component

Given an entity mention identified as an argument of a trigger, the argument type determination component employs a classifier to determine its argument role. Following Li et al., we train this classifier using the same 19 linguistic features that were used to train the argument identifier.

We conclude this section with a note on feature computation. Following the setting of the ACE diagnostic tasks, we use ground truth entities, times and values in argument identification and argument type determination, but the rest of the features are all computed entirely automatically. This is also the setup adopted by Chen and Ji (2009b) and Li et al. (2012).

4 Machine learning Extension

In this section, we present our machine learning extension to the baseline system, which involves joint modeling of the event extraction subtasks. Unlike in pipeline modeling where we train four classifiers (i.e., one classifier per subtask), in joint modeling we train only two classifiers: the joint trigger classifier jointly performs trigger identification and trigger type determination, and the joint argument classifier jointly performs argument identification and argument role determination. Below we describe how these two joint models are trained.[7]

[7] Like us, Tan et al. (2008) also train two classifiers for event extraction, one related to labeling trigger type and the other argument type. So, at first glance, it seems that they are also performing some sort of joint learning for event extraction. However, there is an important difference between our goal and theirs: while we are performing end-to-end event extraction, they are *not*. Their first classifier, the trigger type labeler, is a multi-label *sentence* classifier that assigns to a sentence the

4.1 The Joint Trigger Classifier

To train the joint trigger classifier, we create one training instance for each word in the training set. If the word is not a trigger, the class label of the corresponding training instance is NONE. Otherwise, the class label is the type of the trigger. Each instance is represented by the same 19 features that were used to train the trigger identifier in the baseline system. As in the baseline, we employ SVM^{multiclass} to train this multiclass classifier.

After training, we apply the resulting SVM classifier to classify the test instances. Test instances are created via the same trigger candidate extraction process (including the use of compositional semantics) as described in the baseline system and are represented using the same 19 features as the training instances. If a test instance is assigned the class NONE by the classifier, the corresponding trigger candidate will be posited as a non-trigger. On the other hand, if the instance is classified as belonging to one of the 33 trigger types, the corresponding trigger is posited as a true trigger, and its type is the class value assigned by the classifier.

4.2 The Joint Argument Classifier

To train the joint argument classifier, we create a training instance by pairing each predicted trigger with each of its candidate arguments, where a candidate argument can be an entity mention, a value, or a time expression that appears in the same sentence as the predicted trigger. If the candidate argument is indeed a true argument of the trigger, the class label of the training instance is the argument's role. Otherwise, its class label is NONE. Each instance is represented by the same 19 features that were used to train the argument identifier in the baseline system. As in the baseline, we employ SVM^{multiclass} to train this multiclass classifier.

After training, we apply the resulting SVM classifier to classify the test instances. Test instances are created in the same way as the training instances. If a test instance is assigned the class NONE by the classifier, the corresponding argument candidate is classified as not an argument of the trigger under consideration. Otherwise, the argument candidate is indeed a true argument of the trigger, and its role is the class value assigned by the classifier.

5 Linguistic Extension

In this section, we describe our linguistic extension to the baseline system, where we introduce six groups of features for Chinese event extraction. Before we describe these features, there are two points that deserve mention. First, while this is a linguistic extension to the baseline system, there is nothing that prevents it from being applied to an event extraction system that employs joint learning. Second, solely for the sake of convenience, we follow the convention adopted in the baseline system, ensuring that (1) the trigger identifier and the trigger type labeler employ the same set of features, and (2) the argument identifier and the argument role labeler employ the same set of features. This implies that any group of features that we introduce below has to be used by either (1) both trigger-related classifiers; or (2) both argument-related classifiers, or (3) all four classifiers.

set of trigger types of the triggers it contains, but unlike ours, it does not explicitly identify triggers, even though it has some joint learning flavor in the sense that it allows the NULL class to be assigned to a sentence to indicate that it does not contain any triggers. Their second classifier, the argument type labeler, assumes that the input arguments of a trigger are correctly identified and simply performs argument labeling.

5.1 Character-Based Features

Recall that Li et al. (2012) have employed compositional semantics to improve the extraction of candidate triggers, enabling us to extract 撞伤 [injure by hitting], which appears in the test set but not the training set, as a candidate trigger if 刺伤 [injure by stabbing] is present in the training set as a true trigger, for instance.

Now, recall that each trigger candidate will be classified by the learned trigger identifier as either a true trigger or a non-trigger. Consider, for example, how the instance corresponding to 撞伤 will be classified. One of the linguistic features representing this instance is the word itself. But since 撞伤 never appears in the training set, the word feature is useless as far as classification is concerned. In other words, although 撞伤 and 刺伤 are similar verbs, the word feature does not capture such similarity, and therefore the classifier cannot exploit this similarity when classifying 撞伤.

To address the problem, we decompose a word into two units, putting the two units into separate bins. There are a four cases to consider: (1) if the word has two characters, we put the first character into the first bin and the second character into the second bin; (2) if the word has only one character, we put this character into both bins; (3) if the word has three characters and it can be segmented by a word segmenter, we put the resulting units into the two bins (for example, given the word 公开信 [open letter], 公开 [open] is placed in the first bin and 信 [letter] in the second); and (4) if the word has four characters or the word has three characters that cannot be segmented, we simply put the first two characters into the first bin and the remaining characters into the second bin.

Given these bins, we create four character-based features for the two trigger-related classifiers: The first two features come from the characters in the first and second bins respectively. The third and fourth features consist of the Harbin Institute of Technology NLP Group's synonym dictionary entry numbers for characters in the first and second bins respectively.

5.2 Semantic Role Labeling

It should be easy to see why semantic role labeling is useful for event extraction. First, a large portion of the triggers defined in event extraction task are predicates. Second, if a predicate happens to be a trigger, the predicate's arguments are essentially its event arguments. Furthermore, even though a semantic role labeler typically assigns PropBank-style roles (e.g., Arg0, Arg1) and the event argument roles are FrameNet-style roles, there is a close correspondence between the roles in these two styles.

Given the above observation, we hypothesize that semantic roles are useful for argument identification and argument role labeling. Consequently, we introduce four binary features for the argument classifiers: whether the trigger under consideration is a predicate according to the semantic role labeler, and whether the argument is the predicate's Arg0, Arg1, and time argument. We obtain semantic roles automatically using a publicly available semantic role labeling tool (Björkelund et al., 2009).

In addition, we hypothesize that semantic roles are also useful for trigger identification and trigger type classification. Our hypothesis stems in part from two observations: (1) many predicates identified by a semantic role labeler are triggers; and (2) knowing the entity types of the arguments identified by the semantic role labeler can help predict the type of the trigger. As a result, we propose to employ features that encode semantic role information for trigger identification and trigger type classification. Specifically, we introduce five features: whether the word under consideration is a predicate according to the semantic role labeler; if yes, the entity type and subtype of its Arg0;

the entity type and subtype of its Arg1.

5.3 Trigger Probability Feature

We define *trigger probability* of a word *w* as the probability that *w* appears as a true trigger in the training set. This probability is potentially useful for trigger identification: a word with a higher probability is more likely to be a true trigger.

We create a new feature for the two trigger-related classifiers whose value is the trigger probability of the word under consideration. If the word does not appear in the training set, we determine whether one of its "similar" verbs appears as a trigger in the training set.[8] If so, its trigger probability is that of its similar verb.[9] Otherwise, its trigger probability is set to zero.

5.4 Zero Pronoun Features

To motivate zero pronoun features, consider the following sentence:

国家主席江泽民今天晚上乘专机离开深圳前往文莱。
President Jiang Zemin took the plane tonight, left Shenzhen and went to Brunei.

The verb 乘 [took] has an overt subject 江泽民 [Jiang Zemin]. However, neither of two verbs 离开 [left] or 前往 [went] has an overt subject. Here, the two gaps before 离开 [left] and 前往 [went] are called zero pronouns. A zero pronoun has as its antecedent an entity mention that can fill the gap. In this example, 江泽民 [Jiang Zemin] is the entity mention that should be used to fill the gap: as we can see, 江泽民 [Jiang Zemin] is coreferent with those two zero pronouns.

Kim (2000) studied the difference of the usage of overt subject between English and Chinese. He found that the usage percent of overt subject in Chinese is only 64%, while for English the percent is more than 96%. Thus, zero pronoun is a prominent phenomenon in Chinese, and also appears frequently in the ACE 2005 dataset.

For event extraction, if there is a zero pronoun before a trigger, the entity mention to which this zero pronoun refers is likely to be an argument of this trigger. Thus, zero pronoun resolution, which involves (1) detecting zero pronouns and (2) finding their antecedents, is helpful for argument extraction. To our knowledge, results of zero pronoun resolution have not been exploited to improve event extraction.

In order to exploit zero pronoun resolution for use in event extraction, we need to build a zero pronoun resolver. In our experiments, we apply a rule-based method for zero pronoun resolution. Specifically, to detect zero pronouns, we employ a simple heuristic, which posits that a zero pronoun exists before a word if it is a verb and it has no overt subject in the corresponding syntactic parse tree. After detecting a zero pronoun, we resolve it in one of two ways:

Case 1: the verb following the zero pronoun is in a CP node and modifies a NP to its right (i.e., the

[8] We adopt Li et al.'s (2012) method for determining whether two Chinese verbs are similar. Specifically, we first analyze the structure of each of the verbs under consideration. According to Li et al.'s empirical observation, Chinese verbs possess one of six main structures, where BV is one of the elements of the set *BV* (as defined in Section 3): (1) BV (e.g., " 逮" [arrest]); (2) BV + verb (e.g., " 追杀" [chase to kill]); (3) verb + BV (e.g., " 躲进" [hide]); (4) BV + complementation (e.g., " 进了" [enter]); (5) BV + noun/adjective (e.g., " 开枪" [shoot]); (6) noun/adjective + BV (e.g., " 内战"[civil war]). If the two verbs have the same BV (basic verb) and the same structure, they are considered similar to each other.

[9] It is, of course, possible for a verb to have more than one similar verb. In this case, we compute its trigger probability based on one of its randomly selected similar verbs.

character " 的" appears between the verb and the NP), as shown in the following example, where 歼敌 [kills] modifies 胡修道 [Hu XiuDao]

歼敌 280 人的胡修道.
Hu XiuDao, who kills 280 enemies.

In this case, we resolve the zero pronoun to the NP modified by the verb, 胡修道 [Hu XiuDao].

Case 2 (default case): we resolve the zero pronoun to the nearest preceding NP that occupies the subject position and appears in the same sentence as the zero pronoun.

Next we create features that encode the output of the zero pronoun resolver for the two argument-related classifiers. Recall that each instance in the argument-related classifiers corresponds to a trigger and one of its candidate arguments. Keeping this in mind, the first feature we create encodes whether or not there is a zero pronoun before this trigger. If so, the second feature tells whether or not this argument is coreferent with the zero pronoun.

5.5 Trigger Type Consistency Features

All the features we have described thus far have focused on sentence-level extraction. However, document-level information also plays an important role in event extraction task. A good example is DC (Li et al., 2012). In this subsection, we propose another kind of document-level information, which we call trigger type consistency, to improve event extraction.

Trigger type consistency is motivated by one observation: documents in the ACE 2005 Chinese corpus are mostly news articles, each of which describes one theme, and most of the true triggers are compatible with this document theme. For example, if a document is about a fire accident, most of the annotated triggers in the gold standard are of type DIE. Therefore, knowing the document theme may help to identify triggers. We represent the theme of a document by the trigger type that occurs most frequently among the triggers in the document.[10] For example, if a document has 10 triggers and six of them have type DIE, then we use DIE to represent its theme. If a candidate trigger's type is the same as that of the majority of the triggers in the document, it is being *trigger-type-consistent* with the other triggers in the document and is more likely be a true trigger.

We create 33 features for the two trigger-related classifiers based on trigger type consistency. Each feature corresponds to one of the 33 predefined trigger types in the ACE 2005 event extraction task. We compute the feature values as follows. If, for example, one trigger has type DIE, then (1) the value of the feature corresponding to DIE is the probability that a trigger in this document has type DIE; and (2) the values of the remaining 32 trigger type consistency features are all zero.

A natural question is: since the type consistency features are to be used by the trigger-related classifiers, how is it possible that they are computed based on knowing which words are triggers and what their types are? The answer is that before computing the type consistency features, we run the baseline trigger identifier and the trigger type classifier to identify triggers and predict their types on each document.[11]

[10] For another way of computing the document theme, see Liao and Grishman (2011a).

[11] To identify triggers and predict their types on a test document, we train the two trigger-related baseline classifiers on the training set. On the other hand, to identify triggers and predict their types on a training document, we employ cross validation on the training set: we partition the training set into 9 folds, train the baseline classifiers on 8 folds, apply them to the documents in the remaining fold, and repeat this process 9 times so that we can obtain triggers and their types for each document in the training set.

5.6 Argument Consistency Feature

Another piece of document-level information we employ is argument consistency. It is based on the following observation: the true triggers typically correspond to events that are related to the main person or some major entities mentioned in the documents. Hence, if a candidate trigger has arguments that are coreferent with the arguments of true triggers, the candidate trigger will likely be a true trigger. Consider, for example, the following sentences:

[一家三口] 在昨天深夜集体喝下农药 [自杀].

[A family of three] drank pesticide to [suicide] last night.

[三个人] 总算是稳住了 [病情].

[Three people] finally stabilize the [patient's condition].

Since 自杀 [suicide] is a predicate that is frequently annotated as a trigger in the training data, it should be fairly easy for the learned trigger identifier to predict 自杀 as a trigger. On the other hand, it is difficult to predict 病情 [patient's condition] in the second sentence as a true trigger because many useful features, such as being a predicate or having many entities nearby, cannot be computed due to the lack of useful local information. However, if we know that (1) 病情 takes the argument 三个人 [Three people]; (2) 三个人 is coreferent with 一家三口 [A family of three]; and (3) 一家三口 is an argument of a true trigger 自杀, then we may be able to provide useful document-level information to make the classifier correctly classify the trigger candidate 病情 as a true trigger.

Based on the above observation, we create one feature that encodes this kind of document-level information, which we call argument consistency, for the trigger-related classifiers. The feature is the *role* of the argument that is coreferent with a predicted true trigger's argument.[12] Here is the reason behind using roles as feature values: some roles are more important than the others. Specifically, roles for arguments that serve as subjects or objects, such as VICTIM, are intuitively more important than roles of adjunct arguments, such as PLACE. Using the above two sentences as an example, the role of 三个人 [Three people] is VICTIM, so we will set the value of the feature corresponding to VICTIM as 1.

A natural question is: since the argument consistency feature is to be used by the argument-related classifiers, how is it possible that it is computed based on knowing which words are triggers and what their arguments and argument roles are? The answer is that before computing this feature, we run the baseline classifiers to identify triggers, predict their types, their arguments, and the argument roles on each document (see Footnote 11 for details on how to train these baseline classifiers).

6 Evaluation

Next, we evaluate our joint-learning, knowledge-rich approach to Chinese event extraction.

6.1 Experimental Setup

Dataset and evaluation methodology. All 633 Chinese documents in the ACE 2005 training corpus[13] are used in our evaluation. Unlike previous work (Chen and Ji, 2009b; Li et al., 2012) which designate 10% of the 633 documents as the test set, we perform 10-fold cross-validation experiments in order to obtain more accurate estimation of system performance. While we report

[12] As mentioned at the end of Section 3, since our evaluation setting follows that of the ACE diagnostic tasks, we compute our argument consistency feature based on gold coreference information.

[13] Note that the ACE 2005 test documents are not made publicly available.

results that are averaged over 10 folds, it is worth noting that the results achieved for each of the four subtasks on different folds vary considerably, sometimes by as many as 10 percentage points in F-score, due to the small size of the training and test sets. This suggests the importance of reporting cross-validation results when conducting experiments on the ACE 2005 corpus.

Evaluation measures. For each subtask, we report performance in terms of recall (R), precision (P), and F-score (F). These performance measures are computed based on the following definitions of correctness for the subtasks. For trigger identification, a trigger is correctly identified if its offsets exactly match a reference trigger. For trigger type determination, a trigger type is correctly determined if its trigger type and offsets exactly match a reference trigger. For argument identification, an argument is correctly identified if its offset, related trigger type and trigger's offsets exactly match a reference argument. Finally, for argument role determination, an argument role is correctly determined if its offsets, role, related trigger type and trigger's offsets exactly match a reference argument. Note that these definitions are also adopted by Chen and Ji (2009b) and Li et al. (2012).

6.2 Feature Selection

To determine which of the feature groups described in Section 5 are useful when used in combination with the baseline features in Section 3, we conduct feature selection experiments to identify the best feature subset. There are seven feature groups to be considered in our feature selection experiments: (G1) discourse consistency features (Li et al., 2012); (G2) semantic role labeling features; (G3) trigger probability features; (G4) character-based features; (G5) the argument consistency feature; (G6) trigger type consistency features; and (G7) zero pronoun features. Two points deserve mention. First, among these seven feature groups, G1, G2, G3, G4, G5 and G6 are used for training the trigger-related classifiers whereas G2 and G7 are used for training the argument-related classifiers. Second, while G1 is not a feature group proposed by us, we consider it in our feature selection experiments. The reason is that some of our feature groups (e.g., G5 and G6) also capture document-level information like G1, and because they overlap in terms of the information they capture, we may be better off not retaining all of them.

Feature selection is done using cross validation on the training documents. Specifically, we partition the training documents into 9 folds, train the classifier whose features are to be selected on 8 folds, apply the classifier to the remaining fold, and repeat this process 9 times in order to select the feature groups that have the best average performance over the 9 folds when used in combination with the baseline features.

As far as the feature selection algorithm is concerned, we employ backward elimination. It starts with the full feature set (containing the 7 feature groups to be selected plus the baseline features), and removes in each iteration the feature group whose removal yields the best system performance. We run the algorithm until all but the baseline features are removed, and identify the feature subset that achieves the best performance during the feature selection process.

Note that feature selection is performed separately for each of the four classifiers used in the pipeline approach and each of the two classifiers used in the joint learning approach.

Table 1 shows the feature groups selected for each classifier. Let us first consider the classifiers for the pipeline approach. For trigger identification, all feature groups are retained. For trigger type identification, only G4 (character-based features) are retained. This makes sense because other feature groups are designed only to help discriminate true triggers from wrong triggers. Finally, G2 (semantic role labeling) and G7 (zero pronoun features) prove to be effective for argument identification but not argument role labeling. This means that the best result for argument role labeling is achieved by training the classifier on only the baseline features.

Approach	Classifier	Selected Features
Pipeline	Trigger Identification	G1, G2, G3, G4, G5, G6
	Trigger Type Determination	G4
	Argument Identification	G2, G7
	Argument Role Determination	---
Joint	Trigger Component	G2, G3, G4, G6
	Argument Component	G2, G7

Table 1: Feature selection results

Let us turn to the two classifiers in the joint approach. Interestingly, for the trigger classifier, G1 (discourse consistency) and G5 (argument consistency) are removed. This result provides suggestive evidence that G1 and G5 serve overlapping purposes with the rest of the feature groups we proposed and therefore not all of them need to be retained to achieve the best performance. For the argument classifier, both G2 (semantic role labeling) and G7 (zero pronoun features) are retained. This is consistent with the pipeline results, where both feature groups are shown to be useful for argument identification.

6.3 Test Set Results

Using the features selected for each classifier, we obtain test set results. The average 10-fold cross-validation results for the pipeline approach and the joint approach are shown in Tables 2 and 3, respectively. In each case, we compare our approach against two baselines, one where the classifiers are trained using the baseline feature set without the discourse consistency features, and one where the classifiers are trained using the baseline feature set including the DC features. This setup enables us to better evaluate the usefulness of the DC features: since our feature selection experiments indicate that the DC features are not always useful when used in combination with our proposed features, we want to examine whether they are always useful when used in combination with the baseline features.

	Trigger Identification			Trigger Type Determination			Argument Identification			Argument Role Determination		
Feature Set	R	P	F	R	P	F	R	P	F	R	P	F
Baseline features without DC	50.6	75.5	60.6	47.5	70.8	56.8	35.1	52.3	42.0	31.2	46.5	37.4
Baseline features with DC	55.6	72.7	63.0	52.0	67.9	58.9	38.9	50.2	43.8	34.8	45.0	39.2
Our selected features	60.5	70.1	**64.9**	56.6	65.6	**60.8**	43.8	50.2	**46.8**	39.3	45.1	**42.0**

Table 2: Pipeline modeling results on the test set.

	Trigger Identification			Trigger Type Determination			Argument Identification			Argument Role Determination		
Feature Set	R	P	F	R	P	F	R	P	F	R	P	F
Baseline features without DC	50.0	77.0	60.7	47.5	73.1	57.6	34.1	58.7	43.2	30.4	52.3	38.5
Baseline features with DC	55.3	75.6	63.9	52.6	71.8	60.7	38.2	57.4	45.9	34.3	51.5	41.1
Our selected features	62.2	71.9	**66.7**	58.9	68.1	**63.2**	43.6	57.3	**49.5**	39.2	51.6	**44.6**

Table 3: Joint modeling results on the test set.

A few points about these results deserve mention. First, the DC features offer benefits when used in combination with the baseline features in both the pipeline and joint approaches. Second, we can see that joint modeling always offers benefits over pipeline modeling when we consider comparable rows in the two tables. In fact, using the selected features, the joint approach performs significantly better than the pipeline approach on all four subtasks (paired t-test; $p < 0.05$). Finally, our best-performing system (row 3 of Table 3) significantly outperforms the approach adopted by state-of-the-art event extraction systems (row 2 of Table 2) on all four subtasks (paired t-test, $p < 0.05$): F-score increases by 3.7% for trigger identification, by 4.3% for trigger type determination, by 5.7% for argument identification, and by 5.4% for argument role labeling.

6.4 Feature Analysis

To gain better insight into the contribution of each feature group to our best-performing system (row 3 of Table 3), we add each feature group incrementally to the baseline feature set. Results are shown in Table 4. Recall that the DC features were not selected by the feature selection algorithm for our best-performing system, so none of the results in this table involves DC features. In particular, we start with the baseline features without any DC features (row 1), and add the feature groups incrementally to this baseline feature set.

	Trigger Identification			Trigger Type Determination			Argument Identification			Argument Role Determination		
Features	R	P	F	R	P	F	R	P	F	R	P	F
Baseline features without DC	50.0	77.0	60.7	47.5	73.1	57.6	34.1	58.7	43.2	30.4	52.3	38.5
+G2 (Semantic role labeling)	52.1	77.7	62.4	49.8	74.4	59.7	36.9	61.7	46.2	33.2	55.4	41.5
+G3 (Trigger probability)	56.0	75.3	64.3	53.3	71.5	61.1	39.2	59.7	47.3	35.2	53.7	42.5
+G4 (Character features)	59.8	73.8	66.1	56.6	69.6	62.6	41.2	57.9	48.2	37.2	52.3	43.5
+G6 (Trigger type consistency)	62.2	71.9	66.7	58.9	68.1	63.2	42.7	56.5	48.6	38.5	50.9	43.8
+G7 (Zero pronouns)	62.2	71.9	66.7	58.9	68.1	63.2	43.6	57.3	49.5	39.2	51.6	44.6

Table 4: Results of incremental addition of features to the joint model on the test set

As we can see, except for the argument consistency feature, all feature groups provide gains when added to the feature set in an incremental fashion. For example, in each of the four subtasks, adding semantic role labeling improves F-score by 1.7%, 2.1%, 3.0%, and 3.0%, respectively. Adding trigger probability next improves F-score by 1.9%, 1.4%, 1.1% and 1.0%. After that, adding character features improves F-score by 1.8%, 1.5%, 0.9% and 1.0%. Note that the zero pronoun features are only designed to improve the two argument-related subtasks. So, with the addition of these zero pronoun features, the two trigger-related subtasks are unaffected, while argument identification and argument role determination are improved by 0.9% and 0.8% in F-score, respectively.

Conclusion and Perspectives

We proposed a joint-learning, knowledge-rich approach to a relatively under-studied yet important task, Chinese event extraction, aiming to extend Li et al.'s (2012) state-of-the-art Chinese event extraction system. Linguistically, not only did we propose more effective use of existing features such as character-based features, but we also investigated novel features that exploit results of zero pronoun resolution and noun phrase coreference resolution, as well as those that exploit trigger probability and trigger type consistency. In 10-fold cross-validation experiments on the ACE 2005 dataset, we showed that our system outperformed Li et al.'s system by 3.7−5.7% on the four event extraction subtasks. Our results also indicated that all but the argument consistency feature contributed positively to overall performance. In particular, while each of these feature groups provided small gains, their cumulative benefits were substantial.

Acknowledgments

We thank the three anonymous reviewers for their invaluable comments on an earlier draft of the paper. This work was supported in part by NSF Grant IIS-1147644.

References

Ahn, D. (2006). The stages of event extraction. In *Proceedings of the COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events*, pages 1−8, Sydney, Australia.

Björkelund, A., Hafdell, L., and Nugues, P. (2009). Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 43−48.

Chen, Z. and Ji, H. (2009a). Can one language bootstrap the other: A case study on event extraction. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, pages 66−74.

Chen, Z. and Ji, H. (2009b). Language specific issue and feature exploration in chinese event extraction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 209−212.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363−370.

Grishman, R., Westbrook, D., and Meyers, A. (2005). NYU's English ACE 2005 system description. Technical report, Department of Computer Science, New York University.

Gu, Z. and Cercone, N. (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.

Gupta, P. and Ji, H. (2009). Predicting unknown time arguments based on cross-event propagation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 369−372.

Hardy, H., Kanchakouskaya, V., and Strzalkowski, T. (2006). Automatic event classification using surface text features. In *Proceedings of the AAAI 2006 Workshop on Event Extraction and Synthesis*, pages 36−41.

Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., and Zhu, Q. (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*, pages 1127−1136.

Huang, R. and Riloff, E. (2012a). 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.

Huang, R. and Riloff, E. (2012b). Modeling textual cohesion for event extraction. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*.

Ji, H. (2009). Cross-lingual predicate cluster acquisition to improve bilingual event extraction by inductive learning. In *Proceedings of the NAACL HLT 2009 Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, pages 27−35.

Ji, H. and Grishman, R. (2008). Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254−262.

Kim, Y.-J. (2000). Subject/object drop in the acquisition of korean: A cross-linguistic comparison. *Journal of East Asian Linguistics*, 9:325−351.

Li, P., Zhou, G., Zhu, Q., and Hou, L. (2012). Employing compositional semantics and discourse consistency in chinese event extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1006−1016.

Liao, S. and Grishman, R. (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.

Liao, S. and Grishman, R. (2011a). Acquiring topic features to improve event extraction: in preselected and balanced collections. In *Recent Advances in Natural Language Processing*, pages 9−16.

Liao, S. and Grishman, R. (2011b). Can document selection help semi-supervised learning? a case study on event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 260−265.

Lu, W. and Roth, D. (2012). Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 835−844.

Maslennikov, M. and Chua, T.-S. (2007). A multi-resolution framework for information extraction from free text. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 592−599.

McClosky, D., Surdeanu, M., and Manning, C. (2011). Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1626−1635.

Patwardhan, S. and Riloff, E. (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*, pages 151−160.

Tan, H., Zhao, T., and Zheng, J. (2008). Identification of chinese event and their argument roles. In *Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, CITWORKSHOPS '08, pages 14−19, Washington, DC, USA. IEEE Computer Society.