

Language Specific Issue and Feature Exploration in Chinese Event Extraction

Zheng Chen

Department of Computer Science
The Graduate Center
The City University of New York

365 Fifth Avenue, New York, NY 10016, USA
zchen1@gc.cuny.edu

Heng Ji

Queens College and The Graduate Center
The City University of New York

365 Fifth Avenue, New York, NY 10016, USA
hengji@cs.qc.cuny.edu

Abstract

In this paper, we present a Chinese event extraction system. We point out a language specific issue in Chinese trigger labeling, and then commit to discussing the contributions of lexical, syntactic and semantic features applied in trigger labeling and argument labeling. As a result, we achieved competitive performance, specifically, F-measure of 59.9 in trigger labeling and F-measure of 43.8 in argument labeling.

1 Introduction

In this paper we address the event extraction task defined in Automatic Content Extraction (ACE)¹ program. The ACE program defines the following terminology for event extraction task:

- **Trigger:** the word that most clearly expresses an event's occurrence
- **Argument:** an *entity*, or a *temporal expression* or a *value* that plays a certain role in the event instance
- **Event mention:** a phrase or sentence with a distinguished trigger and participant arguments

Some English event extraction systems based on supervised learning have been reported by researchers (Ahn, 2006; Ji and Grishman, 2008). In this paper we developed a modularized Chinese event extraction system. We nicely handled the language specific issue in trigger labeling and explored effective lexical, syntactic and semantic features that were applied in trigger labeling and argument labeling. Tan et al. (2008) addressed the

same task as we did in this paper. However, to our knowledge, the language specific issue and feature contributions for Chinese event extraction have not been reported by earlier researchers.

The remainder of the paper is organized as follows. Section 2 points out a language specific issue in Chinese trigger labeling and discusses two strategies of trigger labeling: word-based and character-based. Section 3 presents argument labeling. Section 4 discusses the experimental results. Section 5 concludes the paper.

2 Trigger Labeling

We split trigger labeling into two steps: 1) *trigger identification*: to recognize the event trigger 2) *trigger classification*: to assign an event type for the trigger. The two strategies we will discuss in trigger labeling (word-based and character-based) only differ in the first step.

2.1 A Language-Specific Issue

Chinese, and some other languages, e.g., Japanese do not have delimiters between words. Thus, segmentation is usually an indispensable step for further processing, e.g., Part-of-Speech tagging, parsing, etc. However, the segmentation may cause a problem in some tasks, e.g., name entity recognition (Jing et al., 2003) and event trigger identification. For a specific example, “**击毙**” (*shoot and kill*) is segmented as a Chinese word. However, there are two triggers in the word, one is “**击**” (*shoot*) with the event type of *Attack*, and the other is “**毙**” (*kill*) with the event type of *Die*. The trigger may also cross two or more words, e.g., the trigger is “**公开信**” (*public letter*) which crosses two words, “**公开**” (*public*) and “**信**” (*letter*).

In the ACE Chinese corpus, 2902 triggers exactly one-to-one match their corresponding words,

¹ <http://www.nist.gov/speech/tests/ace/>

meanwhile, 431 triggers are inconsistent with the words (either within the word, or across words). The inconsistency rate is as high as 13%.

We then discuss two strategies of trigger labeling, one is word-based in which we use a global errata table to alleviate the inconsistency problem, and the other is character-based which solves the inconsistency problem.

2.2 Word-based Trigger Labeling

We apply Maximum-Entropy based classifiers for *trigger identification* and *trigger classification*. The two classifiers share the same set of features:

- **Lexical features:** word, POS of the word, previous word + word, word + next word, previous POS + POS, and POS + next POS.
- **Syntactic features:** 1) *depth*: the depth of the trigger in the parse tree 2) *path to root*: the path from the leaf node of the trigger to the root in the parse tree 3) *sub-categorization*: the phrase structure expanded by the father of the trigger 4) *phrase type*: the phrase type of the trigger
- **Semantic dictionaries:** 1) *predicate existence*: a boolean value indicating the existence of trigger in a predicate list which is produced from Chinese Propbank (Xue and Palmer, 2008) 2) *synonym entry*: the entry number of the trigger in a Chinese synonym dictionary
- **Nearest entity information:** 1) the entity type of the *syntactically* nearest entity to the trigger in the parse tree 2) the entity type of the *physically* nearest entity to the trigger in the sentence

To deal with the language-specific issue in trigger identification, we construct a global errata table to record the inconsistencies existing in the training set. In the test procedure, if the scanned word has an entry in the errata table, we select the possible triggers in the entry as candidate triggers.

2.3 Character-based Trigger Labeling

Although the error table significantly helps to reduce segmentation inconsistencies, it is not a perfect solution since it only recognizes the inconsistencies existing in the training data.

To take a further step we build a separate character-based *trigger identification* classifier for comparison. We use a MEMM (Maximum Entropy Markov Model) to label each character with a tag indicating whether it is out of the trigger (O), or is the beginning of the trigger (B) or is a part of the trigger except the beginning (I). Our MEMM

classifier performs sequential classification by assigning each character one of the three tags. We then apply Viterbi algorithm to decode the tag sequence and identify the triggers in the sequence.

Features used in our MEMM classifier include: the character, previous character, next character, previous tag and word-based features that the character carries. We apply the same set of features for *trigger classification* as used in word-based trigger labeling.

3 Argument Labeling

We also split argument labeling into two steps: 1) *argument identification*: to recognize an entity or a temporal expression or a value as an argument 2) *role classification*: to assign a role to the argument. We apply Maximum-Entropy based classifiers for the two steps and they share the same set of features:

- **Basic features:** trigger, event subtype of the event mention, type of the ACE entity mention, head word of the entity mention, combined value of event subtype and head word, combined value of event subtype and entity subtype.
- **Neighbor words:** 1) left neighbor word of the entity, temporal expression, or value 2) right neighbor word of the entity, temporal expression, or value
- **Syntactic features:** 1) *sub-categorization*: the phrase structure expanding the parent of the trigger 2) *position*: the relative position of the entity regarding to the trigger (before or after) 3) *path*: the minimal path from the entity to the trigger 4) *distance*: the shortest length from the entity to the trigger in the parse tree

4 Experimental Results

4.1 Data and Scoring Metric

We used 2005 ACE training corpus for our experiments. The corpus contains 633 Chinese documents. In this paper we follow the setting of ACE diagnostic tasks and use the ground truth entities, times and values for our training and testing.

We randomly selected 558 documents as training set and 66 documents as test set. For the training set, we reserved 33 documents as development set.

We define the following standards to determine the *correctness* of an event mention:

- A *trigger* is correctly labeled if its event type and offsets exactly match a reference trigger.
- An *argument* is correctly labeled if its event type, offsets, and role match the reference argument mention.

4.2 Overall System Performance

Table 1 shows the overall Precision (P), Recall (R) and F-Measure (F) scores of our baseline system (word-based system with only lexical features in trigger labeling and basic features in argument labeling), word-based system with full integrated features and character-based system with full integrated features.

Comparing to the Chinese event extraction system reported by (Tan et al., 2008), our scores are much lower. However, we argue that we apply much more strict evaluation metrics.

4.3 Comparison between Word-based and Character-based Trigger Labeling

Table 1 lists the comparison results between character-based and word-based trigger labeling. It indicates that the character-based method outperforms the word-based method, mostly due to the better performance in the step of *trigger identification* (3.3% improvement in F-Measure) with precision as high as 82.4% (14.3% improvement), and a little loss in recall (2.1%).

4.4 Feature Contributions for Trigger Labeling

Table 2 presents the feature contributions for word-based trigger labeling, and we observe similar feature contributions for character-based since it only differs from word-based in *trigger identification* and works similarly in *trigger classification* (we omit the results here). Table 2 shows that maintaining an errata table is an effective strategy for word-based *trigger identification* and dictionary resources improve the performance.

It is worth noting that the performance drops when integrating the syntactic features. Our explanation might be that the trigger, unlike the predicate in the semantic role labeling task, can not only be a verb, but also can be a noun or other types. Thus the syntactic position for the trigger in the parse tree is much more flexible than the predicate in Semantic Role Labeling. For this reason, syntactic features are not so discriminative in trigger labeling. Furthermore, the syntactic features cannot

discriminate the word senses of a candidate trigger. In the following example,

S1: 运动员正在 **进入** 球场准备即将到来的球赛
The players are **entering** the stadium to prepare for the coming game.

S2: 很多农产品还没有 **进入** 市场就腐烂。
Many farm products have been rotted before **entering** the market.

The word “**进入**” (*entering*) indicates a “Transport” event in sentence 1 but not in sentence 2. The phrase structures around the word “**进入**” in both sentences are exactly the same (VP→VP-NP). However, if an entity of “PERSON” appears ahead of “**进入**”, the word “**进入**” is much more likely to be a trigger. Hence the features of *nearby entity information* could be effective.

4.5 Feature Contributions for Argument Labeling

Table 3 shows feature contributions for argument labeling after word-based trigger labeling and we also observe the same feature contributions for argument labeling after character-based trigger labeling (results are omitted). It shows that the two *neighbor word* features are fairly effective. We observe that in some patterns of event description, the left word is informative to tell the followed entity mention is an argument. For example, “**被** [Entity]打死”(killed by [Entity]) is a common pattern to describe an attack event, and the left neighbor word of the entity “**被**” (*by*) can strongly imply that the entity is an argument with a role of “Attacker”. Meanwhile, the right word can help reduce the spurious arguments. For example, in the Chinese “**的**” (*of*) structure, the word “**的**” (*of*) strongly suggests that the entity on the left side of “**的**” is not an argument.

The sub-categorization feature contributes little since it is a feature shared by all the arguments in the parse tree. Table 3 also shows that *Path* and *Distance* are two effective features. It is obvious that in the parse tree, each argument attached to the trigger is in a certain syntactic configuration. For example, the path “NP ↑ VP ↓ VV” implies that it might be a Subject-Verb structure and thus the entity in NP is highly likely to be an argument of the trigger (VV). The *Position* feature is helpful to discriminate argument roles in syntactically identical structure, e.g., “Subject Verb Object” structure.

Performance System	Trigger Identification			Trigger Labeling			Argument Identification			Argument Labeling		
	P	R	F	P	R	F	P	R	F	P	R	F
Baseline	61.0	50.0	54.9	58.7	48.2	52.9	49.5	38.2	43.1	44.6	34.4	38.9
Word-based	68.1	52.7	59.4	65.7	50.9	57.4	56.1	38.2	45.4	53.1	36.2	43.1
Character-based	82.4	50.6	62.7	78.8	48.3	59.9	64.4	36.4	46.5	60.6	34.3	43.8

Table 1. Overall system performance (%)

	Trigger Identification			Trigger Labeling		
	P	R	F	P	R	F
Lexical features : (1)	61.0	50.0	54.9	58.7	48.2	52.9
(1) + Errata table: (2)	64.0	52.0	57.4	61.3	49.8	54.9
(2) + Dictionaries: (3)	64.9	53.5	58.6	62.7	51.6	56.6
(3)+ Syntactic features: (4)	64.3	51.8	57.4	60.6	48.9	54.1
(3) + Entity information: (5)	68.1	52.7	59.4	65.7	50.9	57.4

Table 2. Feature contributions for word-based trigger labeling (%)

	Argument Identification			Argument Labeling		
	P	R	F	P	R	F
Basic feature set: (1)	40.5	32.8	36.2	37.7	30.5	33.7
(1)+Left word: (2)	45.2	35.4	39.7	41.6	32.5	36.5
(1)+Right word: (3)	47.7	35.6	40.8	44.1	32.9	37.7
Feature set 2: (2)+(3)	49.0	35.7	41.3	46.1	33.6	38.9
(1)+Sub-categorization: (4)	41.9	33.1	37.0	38.7	30.5	34.1
(1)+Path: (5)	46.6	36.2	40.7	43.4	33.7	38.0
(1)+Distance: (6)	49.5	37.0	42.3	45.0	33.6	38.5
(1)+Position:(7)	43.8	35.3	39.1	41.0	33.1	36.6
Feature set 3 (from 4 to 7)	56.2	36.1	43.9	51.2	32.9	40.0
Total	56.1	38.2	45.4	53.1	36.2	43.1

Table 3. Feature contributions for argument labeling after word-based trigger labeling (%)

5 Conclusions and Future Work

In this paper, we took a close look at language specific issue in Chinese event extraction and explored effective features for Chinese event extraction task. All our work contributes to setting up a high performance Chinese event extraction system.

For future work, we intend to explore an approach to conducting cross-lingual event extraction and investigate whether the cross-lingual inference can bootstrap either side when running two language event extraction systems in parallel.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023 via 27-001022, and the CUNY Research Enhancement Program and GRTI Program.

References

- D. Ahn. 2006. The stages of event extraction. *Proc. COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events*. Sydney, Australia.
- H. Ji and R. Grishman. 2008. Refining Event Extraction Through Cross-document Inference. *Proc. ACL 2008*. Ohio, USA.
- H. Jing, R. Florian, X. Luo, T. Zhang, and A. Ittychiah. 2003. HowtogetaChineseName(Entity): Segmentation and combination issues. *Proc. EMNLP 2003*.
- H. Tan; T. Zhao; J. Zheng. 2008. Identification of Chinese Event and Their Argument Roles. *Proc. of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops*.
- N. Xue and M. Palmer. 2008. Adding Semantic Role to the Chinese Treebank. *Natural Language Engineering*. Cambridge University Press.