Improving Event Detection with Active Learning

Kai Cao[†] **Xiang Li**[†] **Miao Fan**[‡] **Ralph Grishman**[†]
[†] New York University, [‡] Tsinghua University
{kcao, xiangli, grishman}@cs.nyu.edu
fanmiao.cslt.thu@gmail.com

Abstract

Event Detection (ED), one aspect of Information Extraction, involves identifying instances of specified types of events in text. Much of the research on ED has been based on the specifications of the 2005 ACE [Automatic Content Extraction] event task[1], and the associated annotated corpus. However, as the event instances in the ACE corpus are not evenly distributed, some frequent expressions involving ACE events do not appear in the training data, adversely affecting performance. In this paper, we demonstrate the effectiveness of a *Pattern Expansion* technique to import frequent patterns extracted from external corpora to boost ED performance. The experimental results show that our pattern-based system with the expanded patterns can achieve 70.4% (with 1.6% absolute improvement) F-measure over the baseline, an advance over current state-of-the-art systems.

1 Introduction

Event Extraction involves the extraction of particular types of events along with their arguments. In this paper we shall focus on a subproblem, that of Event Detection (ED) – identifying instances of specified types of events in text. In keeping with the design of the ACE [Automatic Content Extraction] Event task, we will associate each event mention with a *trigger*, which is a word or a sequence of words (most often a single verb or nominalization) that expresses that event. More precisely, our task involves identifying event triggers and classifying them into specific types. For instance, according to the ACE 2005 annotation guidelines[2], in the sentence "*She was **killed** in an automobile accident yesterday*", an event detection system should be able to recognize the word "*killed*" as a trigger for the event DIE. This task is quite challenging, as the same event might appear in the form of various trigger expressions and an expression might represent different events in different contexts. ED is a crucial component in the overall Event Extraction task, which also requires event argument identification and argument role labeling.

Most recent research work on the ACE Event Detection task relies on pattern-based or feature-based approaches, creating classifiers for trigger labeling. Since the distribution of ACE event types in the corpus is skewed, the test data includes some relatively common event expressions that do not occur in the training data. To overcome this problem, we propose to use active learning to help include more patterns for boosting ED performance. These patterns will be extracted from external corpora, such as the *EnglishGigaWord* corpus, labeled, and added to the training data. The experimental results demonstrate that our pattern-based system with the expanded patterns can achieve 70.4% (with 1.6% absolute improvement) F-measure over the baseline, an advance over the state-of-the-art systems.

The paper is organized as follows: In Section 2, we will introduce how to apply pattern expansion inside an active learning framework to improve ED performance. We will describe our ED systems including the baseline and enhanced system utilizing pattern expansion in Section 3, and experimental results as well as detailed discussion and comparison will be presented in Section 4. We will compare our approach with related work in

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

[2] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

Proceedings of Recent Advances in Natural Language Processing, pages 72–77,
Hissar, Bulgaria, Sep 7–9 2015.

Section 5, and Section 6 will conclude this work and list our future research directions.

2 Pattern Expansion

Supervised training can be moderately effective in creating an Event Detection system, but the process of annotating the large corpus required for good performance can be very expensive and time-consuming. The ACE 2005 corpus, with about 300,000 words, is one of the largest such corpora, with detailed event annotations covering 33 event types. Nonetheless, many expressions of these event types are not included, limiting performance of the trained system.

To significantly improve coverage through supervised training would require annotation of a corpus several times larger, which would be prohibitively expensive. Instead we used an active learning approach, in which we identified common constructs which were not represented in the original training corpus, selected examples of these constructs and presented these examples to the user for event annotation. In more detail:

1. Computing the frequency of dependency relations: Since our pattern-based framework is based on syntactic patterns taken from dependency parses, we select examples to be labeled based on their dependency relations. We use a large general news corpus to compute frequencies and select particular types of dependency relations (direct object and prepositional object).

2. Filtering Step: Select dependency relations for which the governor (verb) has appeared as a trigger in the training corpus but the dependency relation as a whole has not appeared in the training corpus.

3. For each high-frequency dependency relation, pick the sentence with at least 5 tokens whose dependency tree contains this dependency relation and maximizes the following ranking score function:

$$score(s) = \begin{cases} 0 & len(s) < 5 \\ \dfrac{\prod\limits_{1 \le i \le n} freq(w_i)}{len(s)} & len(s) \ge 5 \end{cases} \tag{1}$$

   where w_i is the ith word in the sentence s, $freq(w_i)$ is the frequency probability of word w_i in the corpus, and $len(s)$ is the number of tokens of the sentence s[3]. This metric favors short sentences with common words, which should be easy to label.

   With this function, the most representative instance matching a pattern would be extracted. For example, if we try to find an instance containing the pattern "`take office`", the following sentence would be extracted:

   He is to take office today.

   This sentence is an instance of the event *Start-Position*.

4. Add the selected sentences: Annotate the selected instances with respect to the presence of event triggers and incorporate the annotated instances into the training data set.

5. Compare the results: Compare the performance of event detection applying pattern expansion with the AceJet baseline (without pattern expansion)

3 System Description

Jet, the Java Extraction Toolkit[4], provides a set of NLP components which can be combined to create information extraction systems. AceJet[5] is a subsystem of Jet to extract the types of information (entities, relations, and events) annotated on the ACE corpora. The AceJet Event Extraction framework is a combination of a pattern-based system and feature-based system.

Training proceeds in three passes over the annotated training corpus. Pass 1 collects all the event patterns, where a pattern consists of a trigger and a set of arguments along with the path from the trigger to each argument; both the dependency path and the linear sequence path (a series of noun chunks and words) are recorded. Pass 2 records the frequency with which each pattern is associated with an event type – the 'event score'. Pass 3 treats the event score as a feature, combines it with a small number of other features and trains a maximum entropy model.

[3] The stop words are not counted here.

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

[5] `http://cs.nyu.edu/grishman/jet/guide/ACEutilities.html`

At test time, to classify a candidate trigger (any word which has appeared at least once as a trigger in the training corpus) the tagger finds the best match between an event pattern and the input sentence and computes an event score. This score, along with other features, serves as input to the maximum entropy model to make the final ED prediction. (This brief description omits the classifiers for event arguments and argument roles.)

We can see from Table 1 that the resulting system performance is competitive with other recent system results, such as the joint beam search described in (Li et al., 2013).

4 Experiments

In this section, we will introduce the evaluation dataset, compare the performance of applying pattern expansion with other state-of-the-art systems, and discuss the contribution of pattern expansion.

4.1 Data

We used the ACE 2005 corpus as our testbed. For comparison, we used the same test set with 40 newswire articles (672 sentences) as in (Ji and Grishman, 2008; Liao and Grishman, 2010) for the experiments, and randomly selected 30 other documents (863 sentences) from different genres as the development set. The remaining 529 documents (14,840 sentences) are used for training.

Regarding the correctness criteria, following the previous work (Ji and Grishman, 2008; Liao and Grishman, 2010; Ji and Grishman, 2011; Li et al., 2013), a trigger candidate is counted as correct if its event subtype and offsets match those of a reference trigger. The ACE 2005 corpus has 33 event subtypes that, along with one class "*None*" for the non-trigger tokens, constitutes a 34-class classification problem in this work.

Finally we use *Precision* (*P*), *Recall* (*R*), and *F-measure* (*F1*) to evaluate the overall performance.

4.2 Performance Comparison

Table 1 presents the overall performance of the systems with gold-standard entity mention and type information. We can see that our system with active learning can improve the performance over our baseline, and also advances the current state-of-the-art systems. In the test sentence, "*The president is to **take office** tomorrow*", for instance, the system with expanded patterns can correctly identify the *Personnel:Start-Position* event, whereas the AceJet baseline even failed to recognize it as an event instance. Another example is, "*... the anti-communist Gen. Suharto **seized power** in 1965*", where the expanded pattern successfully detects the event trigger with the correct type *Personnel:Start-Position*.

4.3 Discussion

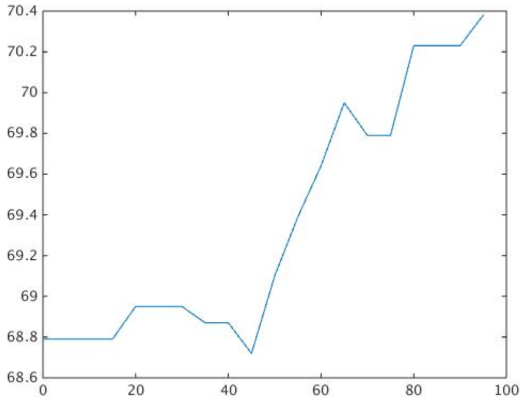

Figure 1: Semi-supervised pattern expansion performance (% in *F-Measure*)

In Figure 1, the x-axis is the number of instances added to the training data, while the y-axis is the corresponding F-measure. We can see from Figure 1 that the pattern expansion helps improve the performance; however the improvement is only modest. This is mainly because the frequent dependency pairs may not be closely related to events and not all dependency pairs align with ACE event patterns very well. Since the pattern-based framework is based on matching dependency relation types and named entity types, noun groups play a central role to identify the events. Therefore, we focus on two types of frequent dependency relations:

- **direct object**
  The object of a verb plays a significant role in understanding the phrase. For example, the phrase "*take office*" means that a duty or title is assumed while other phrases like "*take an apple*" would not trigger an ACE event.

- **preposition and object**
  The noun in the prepositional phrase sometimes conveys as much or more information than the verb. For example, "*fight for independence*" is generally a *Demonstrate* event.

Methods	*P*	*R*	*F1*
Sentence-level in (Ji and Grishman, 2011)	67.6	53.5	59.7
MaxEnt classifier with local features in (Li et al., 2013)	74.5	59.1	65.9
Joint beam search with local features in (Li et al., 2013)	73.7	59.3	65.7
Joint beam search with local and global features in (Li et al., 2013)	73.7	62.3	67.5
Cross-entity in (Ji and Grishman, 2011) †	72.9	64.3	68.3
MaxEnt classifier with local features	70.8	61.4	65.7
AceJet baseline	66.4	71.4	68.8
AceJet system with pattern expansion	68.9	72.0	**70.4**

Table 1: Performance comparison (%) with the state-of-the-art systems. † beyond sentence level.

In contrast, there are three main classes of dependency relations which generally are not helpful in improving ED performance:

1. **Time Patterns**
   Time expressions generally do not help identify the event type. For example, the phrase "*tell Michael on Tuesday*' contains a time-modifying prepositional phrase "*on Tuesday*", but this time modifier plays little role in determining the type of the event. The verb "*tell*" is by itself a strong indicator of a *Contact* event, with the object also playing some role in the classification.

2. **Sports Patterns**
   Since ACE events are mainly about commercial and security-related news, patterns related to sports should be removed. For example, "*win a title*" is one of the top 5 high-frequency dependency pairs in the *EnglishGigaWord* corpus. This pattern appears mostly in a sports-related sentence or article. To remove the sports-related patterns, we plan to build a text classifier and exclude articles classified as sports-related from our frequency counts and as sources of examples.

3. **Redundant Patterns**
   Some verbs strongly favor a single event type. For example, "*die in hospital*" is a high-frequency pattern in *EnglishGigaWord*, however the verb "*die*" is sufficient to identify the *Die* event, whether a man dies in hospital, a room or on the road. Even if this pattern did not appear in the training data, adding it during pattern expansion will do little to improve event classifier accuracy because there are many *Die* events in the training data whose trigger is the verb "*die*". Other information from context will have minimal effect compared to the contribution of the verb "*die*" itself. We believe that such cases can be identified as patterns with triggers a large fraction of whose training examples represent the same event type.

Of the 100 examples tagged, 28 were positive (event triggers); of the 28, we considered 14 to be redundant (not helpful).

5 Related Work

Although there have been quite a few distinct designs for event extraction systems, most are loosely based on using patterns to detect instances of events, where the patterns consist of a predicate, *event trigger*, and constraints on its local syntactic context. The constraints may involve specific lexical items or semantic classes.

Efforts to improve event extraction performance have focused largely on either improving the pattern-matching kernel or adding new reasonable features. Most event extraction frameworks are feature-based systems. Some of the feature-based systems are based on phrase or sentence level extraction. Several recent studies use high-level information to aid local event extraction systems. For example, (Finkel et al., 2005), (Maslennikov and Chua, 2007), (Ji and Grishman, 2008) and (Patwardhan and Riloff, 2007) tried to use discourse, document, or cross-document information to improve information extraction. Other research extends these approaches by introducing cross-event information to enhance the performance of multi-event-type extraction systems. (Liao and Grishman, 2010) use information about other types of events to make predictions or resolve ambiguities regarding a given event. (Li et al., 2013) implements a joint model via structured prediction with cross-event features.

There have been several efforts over the past decade to develop semi-supervised methods for learning such pattern sets. One thread began with Riloff's observation that patterns occurring with substantially higher frequency in relevant documents than in irrelevant documents are likely to be good extraction patterns (Riloff, 1996). (Sudo et al., 2003) sorted relevant from irrelevant documents using a topic description and information retrieval engine. (Yangarber et al., 2000; Yangarber, 2003) developed a bootstrapping approach, starting with some seed patterns, using these patterns to identify some relevant documents, using these documents to identify additional patterns, etc. This approach was further refined in (Surdeanu et al., 2006), which explored alternative pattern ranking strategies. An alternative approach was adopted in (Stevenson and Greenwood, 2005), which used Wordnet-based similarity to expand an initial set of event patterns. (Huang and Riloff, 2012) developed a bootstrapping system to discover new triggers with selected roles. For example, the word "*sniper*" is very likely to be the *agent* of a *Die* event.

There has been growing interest over the last few years in applying active learning methods to reduce the annotation burden involved in developing corpus-trained NLP modules. Active learning has been applied to a variety of Information Extraction tasks, including name tagging, parsing, partial parsing, relation extraction, etc. (Majidi and Crane, 2013) We have previously investigated active learning methods based on co-testing for training relation extractors for ACE relations (Fu and Grishman, 2013). We have also applied such methods for the active learning of ACE event extractors, although with a very different approach (based on the distribution of event triggers across sentences) from that proposed here (Liao and Grishman, 2011).

6 Conclusion and Future Work

To date, the use of supervised methods for creating event extractors has been limited by their poor performance even using large annotated training corpora.

In this paper, we demonstrate the effectiveness of active learning to import more patterns extracted from external corpora to boost Event Detection performance. Since these newly added patterns may never appear in the training data, they can complement the patterns generated from the original training data to enhance ED performance. The experimental results show that our pattern-based system with the expanded patterns can achieve 70.4% (with 1.6% absolute improvement) F-measure over the baseline, an advance over current state-of-the-art systems.

These results were obtained using relatively simple criteria for selecting examples to label: new high-frequency dependency relations involving known triggers. We intend to explore several richer criteria which have have been used for semi-supervised ED, such as similarity measures derived from WordNet, as well as newer methods such as word embeddings using neural network models. This should allow us to improve the efficiency of our active learning by avoiding less promising examples and to improve final ED performance by including triggers not present in the training set.

References

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*.

Lisheng Fu and Ralph Grishman. 2013. An efficient active learning framework for new relation types. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of EACL*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL*.

Heng Ji and Ralph Grishman. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of ACL*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of ACL*.

Shasha Liao and Ralph Grishman. 2011. Using prediction from sentential scope to build a pseudo co-testing learner for event extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*.

Saeed Majidi and Gregory Crane. 2013. Active learning for dependency parsing by a committee of parsers. *IWPT-2013*.

Mstislav Maslennikov and Tat-Seng Chua. 2007. A multi-resolution framework for information extraction from free text. In *Proceedings of ACL*.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of EMNLP*.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of AAAI*.

Mark Stevenson and Mark Greenwood. 2005. A semantic approach to ie pattern induction. In *Proceedings of ACL*.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of ACL*.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. Counter-training in discovery of semantic pattern. In *Proceedings of EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of Coling*.

Roman Yangarber. 2003. Counter-training in discovery of semantic pattern. In *Proceedings of ACL*.