Can Document Selection Help Semi-supervised Learning? A Case Study On Event Extraction

Shasha Liao **Ralph Grishman**
Computer Science Department
New York University
liaoss@cs.nyu.edu grishman@cs.nyu.edu

Abstract

Annotating training data for event extraction is tedious and labor-intensive. Most current event extraction tasks rely on hundreds of annotated documents, but this is often not enough. In this paper, we present a novel self-training strategy, which uses Information Retrieval (IR) to collect a cluster of related documents as the resource for bootstrapping. Also, based on the particular characteristics of this corpus, global inference is applied to provide more confident and informative data selection. We compare this approach to self-training on a normal newswire corpus and show that IR can provide a better corpus for bootstrapping and that global inference can further improve instance selection. We obtain gains of 1.7% in trigger labeling and 2.3% in role labeling through IR and an additional 1.1% in trigger labeling and 1.3% in role labeling by applying global inference.

1 Introduction

The goal of event extraction is to identify instances of a class of events in text. In addition to identifying the event itself, it also identifies all of the *participants* and *attributes* of each event; these are the entities that are involved in that event. The same event might be presented in various expressions, and an expression might represent different events in different contexts. Moreover, for each event type, the event participants and attributes may also appear in multiple forms and exemplars of the different forms may be required. Thus, event extraction is a difficult task and requires substantial training data. However, annotating events for training is a tedious task. Annotators need to read the whole sentence, possibly several sentences, to decide whether there is a specific event or not, and then need to identify the event participants (like Agent and Patient), and attributes (like place and time) to complete an event annotation. As a result, for event extraction tasks like MUC4, MUC6 (MUC 1995) and ACE2005, from one to several hundred annotated documents were needed.

In this paper, we apply a novel self-training process on an existing state-of-the-art baseline system. Although traditional self-training on normal newswire does not work well for this specific task, we managed to use information retrieval (IR) to select a better corpus for bootstrapping. Also, taking advantage of properties of this corpus, cross-document inference is applied to obtain more "informative" probabilities. To the best of our knowledge, we are the first to apply information retrieval and global inference to semi-supervised learning for event extraction.

2 Task Description

Automatic Content Extraction (ACE) defines an event as a specific occurrence involving

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 260–265,
Portland, Oregon, June 19-24, 2011.

participants [1] ; it annotates 8 types and 33 subtypes of events.[2] We first present some ACE terminology to understand this task more easily:

- **Event mention**[3]: a phrase or sentence within which an event is described, including one trigger and an arbitrary number of arguments.
- **Event trigger**: the main word that most clearly expresses an event occurrence.
- **Event mention arguments (roles)**: the entity mentions that are involved in an event mention, and their relation to the event.

Here is an example:

(1) ***Bob Cole*** *was* ***killed*** *in* ***France today; he*** *was* ***attacked****…*

Table 1 shows the results of the preprocessing, including name identification, entity mention classification and coreference, and time stamping. Table 2 shows the results for event extraction.

Mention ID	Head	Ent.ID	Type
E1-1	France	E-1	GPE
T1-1	today	T1	Timex
E2-1	Bob Cole	E-2	PER
E2-2	He	E-2	PER

Table 1. An example of entities and entity mentions and their types

Event type	Trigger	Role		
		Place	**Victim**	**Time**
Die	killed	E1-1	E2-1	T1-1
		Place	**Target**	**Time**
Attack	attacked	E1-1	E2-2	T1-1

Table 2. An example of event triggers and roles

[1]http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines_v5.4.3.pdf

[2] In this paper, we treat the event subtypes separately, and no type hierarchy is considered.

[3] Note that we do not deal with event mention coreference in this paper, so each event mention is treated separately.

3 Related Work

Self-training has been applied to several natural language processing tasks. For event extraction, there are several studies on bootstrapping from a seed pattern set. Riloff (1996) initiated the idea of using document relevance for extracting new patterns, and Yangarber et al. (2000, 2003) incorporated this into a bootstrapping approach, extended by Surdeanu et al. (2006) to co-training. Stevenson and Greenwood (2005) suggested an alternative method for ranking the candidate patterns by lexical similarities. Liao and Grishman (2010b) combined these two approaches to build a filtered ranking algorithm. However, these approaches were focused on finding instances of a scenario/event type rather than on argument role labeling. Starting from a set of documents classified for relevance, Patwardhan and Riloff (2007) created a self-trained relevant sentence classifier and automatically learned domain-relevant extraction patterns. Liu (2009) proposed the BEAR system, which tagged both the events and their roles. However, the new patterns were boostrapped based on the frequencies of sub-pattern mutations or on rules from linguistic contexts, and not on statistical models.

The idea of sense consistency was first introduced and extended to operate across related documents by (Yarowsky, 1995). Yangarber et al. (Yangarber and Jokipii, 2005; Yangarber, 2006; Yangarber et al., 2007) applied cross-document inference to correct local extraction results for disease name, location and start/end time. Mann (2007) encoded specific inference rules to improve extraction of information about CEOs (name, start year, end year). Later, Ji and Grishman (2008) employed a rule-based approach to propagate consistent triggers and arguments across topic-related documents. Gupta and Ji (2009) used a similar approach to recover implicit time information for events. Liao and Grishman (2010a) use a statistical model to infer the cross-event information within a document to improve event extraction.

4 Event Extraction Baseline System

We use a state-of-the-art English IE system as our baseline (Grishman et al. 2005). This system extracts events independently for each sentence, because the definition of event mention arguments in ACE constrains them to appear in the same sentence. The system combines pattern matching with statistical models. In the training process, for every event mention in the ACE training corpus, patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. A set of Maximum Entropy based classifiers are also trained:

- **Argument Classifier:** to distinguish arguments of a potential trigger from non-arguments.
- **Role Classifier:** to classify arguments by argument role. We use the same features as the argument classifier.
- **Reportable-Event Classifier (Trigger Classifier):** Given a potential trigger, an event type, and a set of arguments, to determine whether there is a reportable event mention.

In the test procedure, each document is scanned for instances of triggers from the training corpus. When an instance is found, the system tries to match the environment of the trigger against the set of patterns associated with that trigger. If this pattern-matching process succeeds, the argument classifier is applied to the entity mentions in the sentence to assign the possible arguments; for any argument passing that classifier, the role classifier is used to assign a role to it. Finally, once all arguments have been assigned, the reportable-event classifier is applied to the potential event mention; if the result is successful, this event mention is reported.

5 Our Approach

In self-training, a classifier is first trained with a small amount of labeled data. The classifier is then used to classify the unlabeled data. Typically the most confident unlabeled points, together with their predicted labels, are added to the training set. The classifier is re-trained and the procedure repeated. As a result, the criterion for selecting the most confident examples is critical to the effectiveness of self-training.

To acquire confident samples, we need to first decide how to evaluate the confidence for each event. However, as an event contains one trigger and an arbitrary number of roles, a confident event might contain unconfident arguments. Thus, instead of taking the whole event, we select a partial event, containing one confident trigger and its most confident argument, to feed back to the training system.

For each mention m_i, its probability of filling a *role r* in a reportable event whose trigger is *t* is computed by:

$$P_{RoleOfTrigger}(m_i, r, t) = P_{Arg}(m_i) \times P_{Role}(m_i, r) \times P_{Event}(t)$$

where $P_{Arg}(m_i)$ is the probability from the argument classifier, $P_{Role}(m_i,r)$ is that from the role classifier, and $P_{Event}(t)$ is that from the trigger classifier. In each iteration, we added the most confident <role, trigger> pairs to the training data, and re-trained the system.

5.1 Problems of Traditional Self-training (ST)

However, traditional self-training does not perform very well (see our results in Table 3). The newly added samples do not improve the system performance; instead, its performance stays stable, and even gets worse after several iterations.

We analyzed the data, and found that this is caused by two common problems of traditional self-training. First, the classifier uses its own predictions to train itself, and so a classification mistake can reinforce itself. This is particularly true for event extraction, due to its relatively poor performance, compared to other NLP tasks, like Named Entity Recognition, parsing, or part-of-speech tagging, where self-training has been more successful. Figure 1 shows that the precision using the original training data is not very good: while precision improves with increasing classifier threshold, about 1/3 of the roles are still incorrectly tagged at a threshold of 0.90.

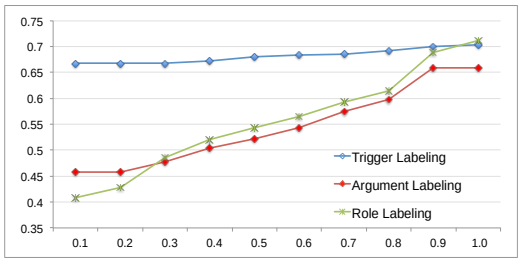

Figure 1. Precision on the original training data with different thresholds (from 0.0 to 0.9)

Another problem of self-training is that nothing "novel" is added because the most confident examples are those frequently seen in the training data and might not provide "new" information. Co-training is a form of self-training which can address this problem to some extent. However, it requires two views of the data, where each example is described using two different feature sets that provide different, complementary information. Ideally, the two views are conditionally independent and each view is sufficient (Zhu, 2008). Co-training has had some success in training (binary) semantic relation extractors for some relations, where the two views correspond to the arguments of the relation and the context of these arguments (Agichtein and Gravano 2000). However, it has had less success for event extraction because event arguments may participate in multiple events in a corpus and individual event instances may omit some arguments.

5.2 Self-training on Information Retrieval Selected Corpus (ST_IR)

To address the first problem (low precision of extracted events), we tried to select a corpus where the baseline system can tag the instances with greater confidence. (Ji and Grishman 2008) have observed that the events in a cluster of documents on the same topics as documents in the training corpus can be tagged more confidently. Thus, we believe that bootstrapping on a corpus of topic-related documents should perform better than a regular newswire corpus.

We followed Ji and Grishman (2008)'s approach and used the INDRI retrieval system[4] (Strohman et al., 2005) to obtain the top *N* related documents for each annotated document in the training corpus. The query is event-based to insure that related documents contain the same events. For each training document, we construct an INDRI query from the triggers and arguments. For example, for sentence (1) in section 2, we use the keywords "*killed*", "*attacked*", "*France*", "*Bob Cole*", and "*today*" to extract related documents. Only names and nominal arguments will be used; pronouns appearing as arguments are not included. For each argument we also add other names coreferential with the argument.

5.3 Self-training using Global Inference (ST_GI)

Although bootstrapping on related documents can solve the problem of "confidence" to some extent, the "novelty" problem still remains: the top-ranked extracted events will be too similar to those in the training corpus. To address this problem, we propose to use a simple form of global inference based on the special characteristics of related-topic documents. Previous studies pointed out that information from wider scope, at the document or cross-document level, could provide non-local information to aid event extraction (Ji and Grishman 2008, Liao and Grishman 2010a). There are two common assumptions within a cluster of related documents (Ji and Grishman 2008):

- **Trigger Consistency Per Cluster:** if one instance of a word triggers an event, other instances of the same word will trigger events of the same type.
- **Role Consistency Per Cluster:** if one entity appears as an argument of multiple events *of the same type* in a cluster of related documents, it should be assigned the same role each time.

Based on these assumptions, if a trigger/role has a low probability from the baseline system, but a high one from global inference, it means that the local context of this trigger/role tag is not frequently seen in the training data, but the tag is still confident. Thus, we can confidently add it to the training data and it can provide novel information which the samples confidently tagged by the baseline system cannot provide.

[4] http://www.lemurproject.org/indri/

To start, the baseline system extracts a set of events and estimates the probability that a particular instance of a word triggers an event of that type, and the probability that it takes a particular argument. The global inference process then begins by collecting all the confident triggers and arguments from a cluster of related documents.[5] For each trigger word and event type, it records the highest probability (over all instances of that word in the cluster) that the word triggers an event of that type. For each argument, within-document and cross-document coreference[6] are used to collect all instances of that entity; we then compute the maximum probability (over all instances) of that argument playing a particular role in a particular event type. These maxima will then be used in place of the locally-computed probabilities in computing the probability of each trigger-argument pair in the formula for $P_{RoleOfTrigger}$ given above.[7] For example, if the entity "Iraq" is tagged confidently (probability > 0.9) as the "Attacker" role somewhere in a cluster, and there is another instance where from local information it is only tagged with 0.1 probability to be an "Attacker" role, we use probability of 0.9 for both instances. In this way, a trigger pair containing this argument is more likely to be added into the training data through bootstrapping, because we have global evidence that this role probability is high, although its local confidence is low. In this way, some novel trigger-argument pairs will be chosen, thus improving the baseline system.

6 Results

We randomly chose 20 newswire texts from the ACE 2005 training corpora (from March to May of 2003) as our evaluation set, and used the remaining newswire texts as the original training data (83 documents). For self-training, we picked 10,000 consecutive newswire texts from the TDT5 corpus from 2003[8] for the ST experiment. For ST_IR and ST_GI, we retrieved the best N (using N = 25, which (Ji and Grishman 2008) found to work best) related texts for each training document from the English TDT5 corpus consisting of 278,108 news texts (from April to September of 2003). In total we retrieved 1650 texts; the IR system returned no texts or fewer than 25 texts for some training documents. In each iteration, we extract 500 trigger and argument pairs to add to the training data.

Results (Table 3) show that bootstrapping on an event-based IR corpus can produce improvements on all three evaluations, while global inference can yield further gains.

	Trigger labeling	Argument labeling	Role labeling
Baseline	54.1	39.2	35.4
ST	54.2	40.0	34.6
ST_IR	55.8	42.1	37.7
ST_GI	**56.9**	**43.8**	**39.0**

Table 3. Performance (F score) with different self-training strategies after 10 iterations

7 Conclusions and Future Work

We proposed a novel self-training process for event extraction that involves information retrieval (IR) and global inference to provide more accurate and informative instances. Experiments show that using an IR-selected corpus improves trigger labeling F score 1.7%, and role labeling 2.3%. Global inference can achieve further improvement of 1.1% for trigger labeling, and 1.3% for role labeling. Also, this bootstrapping involves processing a much

[5] In our experiment, only triggers and roles with probability higher than 0.9 will be extracted.

[6] We use a statistical within-document coreference system (Grishman et al. 2005), and a simple rule-based cross-document coreference system, where entities sharing the same names will be treated as coreferential across documents.

[7] If a word or argument has multiple tags (different event types or roles) in a cluster, and the difference in the probabilities of the two tags is less than some threshold, we treat this as a "conflict" and do not use the conflicting information for global inference.

[8] We selected all bootstrapping data from 2003 newswire, with the same genre and time period as ACE 2005 data to avoid possible influences of variations in the genre or time period on the bootstrapping. Also, we selected 10,000 documents because this size of corpus yielded a set of confidently-extracted events (probability > 0.9) roughly comparable in size to those extracted from the IR-selected corpus; a larger corpus would have slowed the bootstrapping.

smaller but more closely related corpus, which is more efficient. Such pre-selection of documents may benefit bootstrapping for other NLP tasks as well, such as name and relation extraction.

Acknowledgments

We would like to thank Prof. Heng Ji for her kind help in providing IR data and useful suggestions.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. *In Proceedings of 5th ACM International Conference on Digital Libraries.*

Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU's English ACE 2005 System Description. *In Proc. ACE 2005 Evaluation Workshop, Gaithersburg, MD.*

Prashant Gupta and Heng Ji. 2009. Predicting Unknown Time Arguments based on Cross-Event Propagation. *In Proceedings of ACL-IJCNLP 2009.*

Heng Ji and Ralph Grishman. 2008. Refining Event Extraction through Cross-Document Inference. *In Proceedings of ACL-08: HLT, pages 254–262, Columbus, OH, June.*

Shasha Liao and Ralph Grishman. 2010a. Using Document Level Cross-Event Inference to Improve Event Extraction. *In Proceedings of ACL 2010.*

Shasha Liao and Ralph Grishman. 2010b. Filtered Ranking for Bootstrapping in Event Extraction. *In Proceedings of COLING 2010.*

Ting Liu. 2009. Bootstrapping events and relations from text. *Ph.D. thesis, State University of New York at Albany.*

Gideon Mann. 2007. Multi-document Relationship Fusion via Constraints on Probabilistic Databases. *In Proceedings of HLT/NAACL 2007. Rochester, NY, US.*

MUC. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6),* San Mateo, CA. Morgan Kaufmann.

S. Patwardhan and E. Riloff. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. *In Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07).*

Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. *In Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96), pp. 1044-1049.*

M. Stevenson and M. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. *In Proceedings of ACL 2005.*

Trevor Strohman, Donald Metzler, Howard Turtle and W. Bruce Croft. 2005. Indri: A Language-model based Search Engine for Complex Queries (extended version). *Technical Report IR-407, CIIR, UMass Amherst, US.*

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. A Hybrid Approach for the Acquisition of Information Extraction Patterns. *In Proceedings of the 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 2000.*

Roman Yangarber. 2003. Counter-Training in Discovery of Semantic Patterns. *In Proceedings of ACL2003.*

Roman Yangarber and Lauri Jokipii. 2005. Redundancy-based Correction of Automatically Extracted Facts. *In Proceedings of HLT/EMNLP 2005. Vancouver, Canada.*

Roman Yangarber. 2006. Verification of Facts across Document Boundaries. *In Proceedings of International Workshop on Intelligent Information Access. Helsinki, Finland.*

Roman Yangarber, Clive Best, Peter von Etter, Flavio Fuart, David Horby and Ralf Steinberger. 2007. Combining Information about Epidemic Threats from Multiple Sources. *In Proceedings of RANLP 2007 workshop on Multi-source, Multilingual Information Extraction and Summarization. Borovets, Bulgaria.*

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *In Proceedings of ACL 1995. Cambridge, MA.*

Xiaojin Zhu. 2008 Semi-Supervised Learning Literature Survey. *http:// pages.cs.wisc.edu/ ~jerryzhu/research/ssl/semireview.html*