Filtered Ranking for Bootstrapping in Event Extraction

Shasha Liao
Dept. of Computer Science
New York University
liaoss@cs.nyu.edu

Ralph Grishman
Dept. of Computer Science
New York University
grishman@cs.nyu.edu

Abstract

Several researchers have proposed semi-supervised learning methods for adapting event extraction systems to new event types. This paper investigates two kinds of bootstrapping methods used for event extraction: the document-centric and similarity-centric approaches, and proposes a filtered ranking method that combines the advantages of the two. We use a range of extraction tasks to compare the generality of this method to previous work. We analyze the results using two evaluation metrics and observe the effect of different training corpora. Experiments show that our new ranking method not only achieves higher performance on different evaluation metrics, but also is more stable across different bootstrapping corpora.

1 Introduction

The goal of event extraction is to identify instances of a class of events in text, along with the arguments of the event (the participants, place, and time). In this paper we shall focus on the sub-problem of identifying the events themselves.

Event extraction systems from the early and mid 90s relied primarily on hand-coded rules, which must be written anew for every task. Since then, supervised and semi-supervised methods have been developed in order to build systems for new scenarios more easily. Supervised methods can perform quite well with enough training data, but annotating sufficient data may require months of labor. Semi-supervised methods aim to reduce the annotated data required, ideally to a small set of seeds.

Most semi-supervised event extractors seek to learn sets of *patterns* consisting of a predicate and some lexical or semantic constraints on its arguments. The semi-supervised learning was based primarily on one of two assumptions: the document-centric approach, which assumes that relevant patterns should appear more frequently in relevant documents (Riloff 1996; Yangarber et al. 2000; Yangarber 2003; Surdeanu et al 2006); and the similarity-centric approach, which assumes that relevant patterns should have lexically related terms (Stevenson and Greenwood 2005, Greenwood and Stevenson 2006).

An effective semi-supervised extractor will have good performance over a range of extraction tasks and corpora. However, many of the learning procedures just cited have been tested on only one or two extraction tasks, so their generality is uncertain. To remedy this, we have tested learners based on both assumptions, targeting both a MUC (Message Understanding Conference) scenario and several ACE (Automatic Content Extraction) event types. We identify shortcomings of the prior bootstrapping methods, propose a more effective and stable ranking method, and consider the effect of different corpora and evaluation metrics.

2 Related Work

The basic assumption of the document-centric approach is that documents containing a large number of patterns already identified as relevant to a particular IE scenario are likely to contain further relevant patterns. Riloff (1996) initiated

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 680–688,
Beijing, August 2010

this approach and claimed that if a corpus can be divided into documents involving a certain event type and those not involving that type, patterns can be evaluated based on their frequency in relevant and irrelevant documents. Yangarber et al. (2000) incorporated Riloff's metric into a bootstrapping procedure, which started with several seed patterns but required no manual document classification or corpus annotation. The seed patterns were used to identify some relevant documents, and the top-ranked patterns (based on their distribution in relevant and irrelevant documents) were added to the seed set. This process was repeated, assigning a relevance score to each document based on the relevance of the patterns it contains and gradually growing the set of relevant patterns. This approach was further refined by Surdeanu et al. (2006), who used a co-training strategy in which two classifiers seek to classify documents as relevant to a particular scenario. Patwardhan and Riloff (2007) presented an information extraction system that find relevant regions of text and applies extraction patterns within those regions. They created a self-trained relevant sentence classifier to identify relevant regions, and use a semantic affinity measure to automatically learn domain-relevant extraction patterns. They also distinguish primary patterns from secondary patterns and apply the patterns selectively in the relevant regions.

Stevenson and Greenwood (2005) (henceforth 'S&G') suggested an alternative method for ranking the candidate patterns. Their approach relied on the assumption that useful patterns will have similar lexical items to the patterns that have already been accepted. They used WordNet to calculate word similarity. They chose to represent each pattern as a vector consisting of the lexical items and used a version of the cosine metric to determine the similarity between pairs of patterns. Later, Greenwood and Stevenson (2006) introduced a structural similarity measure that could be applied to extraction patterns consisting of linked dependency chains.

3 Ranking Methods in Bootstrapping

Most semi-supervised event extraction systems are based on patterns with variables which have semantic type constraints. A simple example is "*organization* appoints *person* as *position*"; if this pattern matches a passage in a test document, a hiring event will be instantiated with the items matching the variables being the arguments of the event. So training an event extractor becomes primarily a task of acquiring these patterns. In a semi-supervised setting, this involves ranking candidate patterns and accepting the top-ranked patterns at each iteration. Our goal was to create a more robust learner through improved pattern ranking.

3.1 Problems of Document-centric Bootstrapping

Document-centric bootstrapping tries to find patterns with high frequency in relevant documents and low frequency in irrelevant documents. The assumption is that descriptions of the same event or the same type of event may occur multiple times in a document, and so a document containing a relevant pattern is more likely to contain more such patterns. This approach may end up extracting patterns for related events; for example, *start-position* often comes with *end-position* events. This effect may be salutary if the extraction scenario includes these related events (as in MUC-6), but will pose a problem if the goal is to extract individual event types. Also, because an extra corpus for bootstrapping is needed, different corpora might perform quite differently (see Figure 2).

3.2 Problems of Similarity-centric Bootstrapping

Similarity-centric bootstrapping tries to find patterns with high lexical similarities. The most crucial issue is how to evaluate the similarity of two patterns, which is based on the similarity of two words. In this strategy, no extra corpus is needed, which eliminates the effort to find a good bootstrapping corpus, but a semantic dictionary that can provide word similarity is required. S&G used WordNet[1] to provide word similarity information. However, in the similarity-centric approach, lexical polysemy can lead the bootstrapping down false paths. For example, for *start-position (hire)* events, "*name*" and "*charge*" are in the same *Synset* as *appoint*, but including these words is quite dangerous because they contain other common senses

[1] http://wordnet.princeton.edu/

unrelated to *start-position* events. For *die* events, we might have words like "*go*" and "*pass*", which are also used in very specific contexts when they refer to "*die*". If similarity-centric ranking extracts patterns including these words, performance will deteriorate very quickly, because most of the time, these words do not predicate the proper event, and more and more wrong patterns will be extracted.

3.3 Our Approach

We propose a new ranking method, which constrains the document-centric and similarity-centric assumptions, and makes a more restricted assumption: patterns that appear in relevant documents *and* are lexically similar are most likely to be relevant. This method limits the effect of ambiguous patterns by narrowing the search to relevant documents, and limits irrelevant patterns in relevant documents by word similarity restriction. For example, although "charge" has high word similarity to "appoint", its document relevance score is very low, and we will not include this word in bootstrapping starting from "appoint".

Many different combinations are possible; we propose one that uses the word similarity as a filter. The document relevance score is first applied to rank the patterns in relevant documents, then the patterns with lexical similarity scores below a similarity threshold will be removed from the ranking; only patterns above threshold will be added to the seeds. However, if in the current iteration, no pattern meets the threshold, the threshold will be lowered until new patterns can be found. We call this ranking method *filtered ranking*[2]:

$$Filter(p) = \begin{cases} Yangarber(p) & Stevenson(p) >= t \\ 0 & otherwise \end{cases}$$

where t is the threshold, which is initialized to 0.9 in our experiments.

4 System Description

Our approach is similar to that for document-centric bootstrapping, but the ranking function is changed to incorporate lexical similarity information. For our experiments bootstrapping was terminated after a fixed number of iterations; in practice, we would monitor performance on a held-out (dev-test) sample and stop when it declines for k iterations.

4.1 Pre-processing

Instead of limiting ourselves to surface syntactic relations, we want to get more general and meaningful patterns. To this end, we used semantic role labeling (Gildea and Jurafsky, 2002) to generate the logical grammatical and predicate-argument representation automatically from a parse tree (Meyers et al. 2009). The output of the semantic labeling is the dependency representation of the text, where each sentence is a graph consisting of nodes (corresponding to words) and arcs. Each arc captures up to three relations between two words: (1) a SURFACE relation, the relation between a predicate and an argument in the parse tree of a sentence; (2) a LOGIC1 (grammatical logical) relation which regularizes for lexical and syntactic phenomena like passive, relative clauses, and deleted subjects; and (3) a LOGIC2 (predicate-argument) relation corresponding to relations in PropBank (Palmer et al. 2005) and NomBank

In constructing extraction patterns from this graph, we take each dependency link along with its predicate-argument role; if that role is null, we use its logical grammatical role, and finally, its surface role. For example, for the sentence:

`John is hit by Tom's brother.`

we generate the patterns:

<Arg1 hit John>
<Arg0 hit brother>
<T-pos brother Tom>

where the first two represent LOGIC2 relations and the third a SURFACE relation. To reduce data sparseness, all inflected words are changed to their root form (e.g. "attackers"→"attacker"), and all names are replaced by their ACE type (*person, organization, location, etc.*), so the first pattern would become

<Arg1 hit PERSON>

4.2 Document-based Ranking

The document-centric method employs a

[2] We also tried using the product of the document relevance score and word similarity score, and found the results to be quite similar. Due to space limitations, we do not report these results here.

re-implementation of the procedure described in (Yangarber et al. 2000), using the disjunctive voting scheme for document relevance. At each iteration *i* we compute a precision score $Prec^i(p)$ for each pattern *p* and a relevance score $Rel^i(d)$ for each document *d*. Initially the seed patterns have precision 1 and all other patterns precision 0. These are updated by

$$\mathrm{Re}\,l^i(d) = 1 - \prod_{p \in K(d)} (1 - \mathrm{Pr}\,ec^i(p))$$

where *K(d)* is the set of accepted patterns that match document *d*, and

$$\mathrm{Pr}\,ec^{i+1}(p) = \frac{1}{|H(p)|} \bullet \sum_{d \in H(p)} \mathrm{Re}\,l^i(d)$$

where *H(p)* is the set of documents matching pattern *p*. Patterns are then ranked by

$$RankFun_{Yangarber}(p) = \frac{Sup(p)}{|H(p)|} * \log Sup(p)$$

where $$Sup(p) = \sum_{d \in H(p)} \mathrm{Re}\,l(d)$$

(a generalization of Yangarber's metric), and the top-ranked candidates are added to the set of accepted patterns.

4.3 Pattern Similarity

For two words, there are several ways to measure their similarity using WordNet, which can be roughly divided into two categories: distance-based, including Leacock and Chodorow (1998), Wu and Palmer (1994); and information content based, including Resnik (1995), Lin (1998), and Jiang and Conrath (1997). We follow S&G (2005)'s method and use the semantic similarity of concepts based on Information Content (IC).

Every pattern consists of a predicate and a constraint ("argument") on its local syntactic context, and so the similarity of two patterns depends on the similarity of the predicates and the similarity of the arguments. We modified S&G's structural similarity measure to reflect some differences in pattern structure: first, S&G only focus on patterns headed by verbs, while we include verbs, nouns and adjectives; second, they only record the subject and object to a verb, while we record all argument relations; third, our patterns only contain a predicate and a single constraint (argument), while their pattern might contain two arguments, subject and object. With two arguments, many more patterns are possible and the vector similarity calculation over all patterns in a large corpus becomes very time consuming.

We do not limit ourselves to verb patterns because nouns and (occasionally) adjectives can also represent an event. For example, "Stevenson's promotion is a signal …" expresses a *start-position* event. Moreover, in our pattern, we assume that the predicate is more important than constraint, because it is the root (head) of the pattern in the semantic graph structure, and place different weights on predicate and constraint. Finally, the similarity of two patterns p_1 and p_2 is computed as follows:

$$Sim(p_1, p_2) = \alpha * Sim(f_1, f_2) + \beta * Sim(r_1, r_2) * Sim(a_1, a_2)$$

where $\alpha+\beta=1$, *f* represents a predicate, *r* represent a role, and *a* represent an argument. In our experiment, α is set to 0.6 and β is set to 0.4. The role similarity is 1 for identical roles and for roles which generally correspond at the syntactic and predicate-argument level (arg0 ↔ subj; arg1 ↔ obj); selected other role pairs are assigned a small positive similarity (0.1 or 0.2), and others 0.

As with the document-centric method, bootstrapping begins by accepting a set of seed patterns. At each iteration, the procedure computes the similarity between all patterns in the training corpus and the currently accepted patterns and accepts the most similar pattern(s). In S&G's experiments the evaluation corpus also served as the training corpus.

5 Experiments

There have been two types of event extraction tasks. One involved several 'elementary' event types, such as "attack", "die", "injure" etc.; for example, the ACE 2005 evaluation[3] used a set of 33 event types and subtypes. The other type involved a *scenario* – a set of related events, like "attacks and the damage, injury, and death they cause", or "arrest, trial, sentencing etc.". The

[3]See http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines_v5.4.3.pdf for a description of this task.

MUC evaluations included two scenarios that have been the subject of considerable research on learning methods: *terrorist incidents* (MUC-3/4) and *executive succession* (MUC-6).

We conducted experiments on the MUC-6 task to make a comparison to previous work. We also did experiments on ACE 2005 data, because it provides many distinct event types; we conducted experiments on three disparate event types: *attack*, *die*, and *start-position*. Note that MUC-6 identifies a scenario while ACE identifies specific event types, and types which are in the same MUC scenario might represent different ACE events. For example, the *executive succession* scenario (MUC-6) includes the *start-position* and *end-position* events in ACE.

5.1 Data Description

There are four corpora used in the experiments:

MUC-6 corpora

- **Bootstrapping:** pre-selected data from the Reuters corpus (Rose et al. 2002) from 1996 and 1997, including 3000 related documents and 3000 randomly chosen unrelated documents
- **Evaluation:** MUC-6 annotated data, including 200 documents (official training and test). We were guided by the MUC-6 key file in annotating every document and sentence as relevant or irrelevant.

ACE corpora

- **Bootstrapping:** untagged data from the Gigaword corpus from January 2006, including 14,171 English newswire articles from Agence France-Presse (AFP).
- **Evaluation:** ACE 2005 annotated (training) data, including 589 documents

5.2 Parameters used in Experiments

In our bootstrapping process, we only extract patterns appearing more than 2 times in the corpus, and the similarity filter threshold is originally set to 0.9. If no patterns are found, it is reduced by 0.1 until new patterns are found.

In each iteration, the top 3 patterns in the ranking function will be added to the seeds.

For the similarity-centric method, only patterns appearing more than 2 times and in less than 30% of the documents will be extracted, which is the same as S&G's approach.

5.3 MUC-6 Experiments

Our overall goal was to demonstrate that filtered ranking was in all cases competitive with and in at least some cases clearly superior to the earlier methods, over a range of extraction tasks and bootstrapping corpora. We began with the MUC-6 task, where the efficacy of the earlier methods had already been demonstrated.

< Arg0 resign Person >
< Arg1 appoint Person >
< Arg0 appoint Org_commercial>
<Arg1 succeed Person >

Table 1. Seeds for MUC-6 evaluation

For MUC-6 evaluation, we follow S&G's approach and assess extraction patterns by their ability to identify event-relevant sentences.[4] The system treats a sentence as relevant if it matches an extraction pattern. Bootstrapping starts from four seeds which yield 80% precision and 24% recall for sentence filtering.

To compare with previous work, we tested the filtered ranking method on two corpora: the first is the Reuters corpus used in S&G's recreation of Yangarber's experiment (Filter1), to compare with their results for the document-centric method; the second uses the test corpus as S&G did (Filter2), to compare with their results for the similarity-centric method. We compare methods based on peak F score; in practice, this would mean controlling the bootstrapping using a held-out test sample.

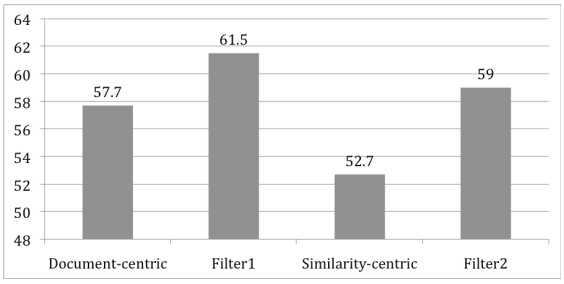

Figure 1. F score for different ranking methods on MUC-6 evaluation

Figure 1 showed that the filtered ranking

[4] We also tried the document filtering evaluation introduced by Yangarber but, as S&G observed, this metric is too insensitive because over 50% of the documents in the MUC-6 test set are relevant.

methods edge out both document and similarity-centric methods. Our scores are comparable to S&G's, although they report somewhat better performance for similarity-centric than for document-centric (55 vs. 51) whereas document-centric did better for us. This difference may reflect differences in pattern generation (discussed above) and possibly differences in the specific corpora used.

However, document-centric bootstrapping needs an extra corpus for bootstrapping; S&G used a pre-selected corpus that contains approximately same number of relevant and irrelevant documents[5]. We wanted to check if such a corpus is essential for the document-centric method, and if the need for pre-selection can be reduced through filtered ranking. Thus, we set up another experiment to see if the document-centric method is stable or sensitive to different corpora. We used two additional corpora for MUC-6 evaluation: one is a subset of the Wall Street Journal (WSJ) 1991 corpus, which contains 18,734 untagged documents; the other is the Gigaword AFP corpus described in section 5.1. Both corpora are much larger than the Reuters corpus, and while we do not have precise information about relevant document density, the WSJ contains quite a few *start-position* events because it is primarily business news; the Gigaword corpus (AFP newswire) has fewer *start-position* events because it contains a wider variety of news.

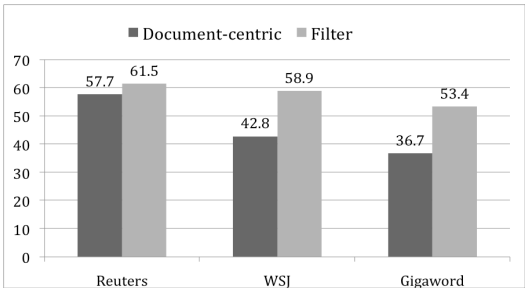

Figure 2. Document-centric and Filtered ranking results on different corpora for MUC-6

Figure 2 showed that the document-centric method performs quite differently on different corpora, which indicates that a pre-selected corpus plays an important role in document-centric ranking. It suggests that the percentage of relevant documents may be more important than the overall corpus size. The figure also shows that filtered ranking is much more stable across different corpora. Richer corpora still have better peak performance, but the difference is not quite as great; also, peak performance on a given corpus is consistently better than the document-centric method.

From the above experiments, we conclude that our filtering method is better in two aspects: first, bootstrapping on the same corpus performs better than either document or similarity-centric methods; second, if we can not get a corpus with an assured high density of relevant documents, it is safer to use filtered ranking because it is more stable across different corpora.

5.4 ACE2005 Experiments

The ACE2005 corpus includes annotations for 33 different event types and subtypes, offering us an opportunity to assess the generality of our methods across disparate event types. We selected 3 event types to report on here:

- **Die:** "occurs whenever the life of a PERSON Entity ends. It can be accidental, intentional or self-inflicted." This event appears 535 times in the corpus.
- **Attack:** "is defined as a violent physical act causing harm or damage." Attack events include a variety of sub-events like "person attack person", "country invade country", and "weapons attack locations". This event type appears 1120 times.
- **Start-Position:** "occurs whenever a PERSON Entity begins working for (or changes offices within) an ORGANIZATION or GPE. This includes government officials starting their terms, whether elected or appointed". It appears 116 times in the corpus.

We choose these three event types because they reflect the diversity of events ACE annotated: *die* events appear frequently in the ACE corpus and its definition is very clear; *attack* events also appear frequently, but its definition is rather complicated and contains several different sub-events; *start-position*'s definition is clear, but it is relatively infrequent in the corpus.

Based on the observations from the MUC-6 corpus, we eschewed corpus pre-selection for

[5] The pre-selection of relevant and irrelevant documents is based on document meta-data provided as part of the Reuters Corpus Volume I (Rose et al., 2002).

two reasons: first, building a different corpus for training each event type is an extra burden in developing a system for handling multiple events; second, we want to demonstrate that filtered ranking would work without pre-selection, while the document-centric method does not. As a result, we used the Gigaword AFP corpus for all event types.

In the ACE 2005 corpus, for every event, the annotators recorded a trigger, which is the main word that most clearly expresses an event occurrence. This added information allowed us to conduct dual evaluations: one based on sentence relevance - following S&G - presented in section 5.4.2, and one based on trigger identification, presented in section 5.4.3.

5.4.1 ACE2005 Supervised Model

To provide a benchmark for our semi-supervised learners, we built a very simple pattern-based supervised learning model. For training, for every pattern, we count how many times it contains an event trigger and how many times it does not. If more than 50% of the time it contains an event trigger, we treat it as a positive pattern.

For sentence level evaluation, if there is a positive pattern in a sentence, we tag this sentence as relevant; otherwise not. For word level evaluation, if the word is the predicator of a positive pattern, we tag it as a trigger; otherwise not[6].

We did a 5-fold cross-validation on the ACE 2005 data, report the average results and compare it to the semi-supervised learning method (see figure 3 & 4).

5.4.2 Sentence level ACE Event Evaluation[7]

Different event types have quite different performance (see figure 3): for the *die* event, the peak performance of all methods is quite good, and quite close to the supervised result; for the *attack* event, filtered ranking performs much better than both document and similarity-centric methods, but still worse than the supervised method; for *start-position* events, the semi-supervised method beats the supervised method. The reason might be as follows:

Die events appear frequently in ACE 2005, and most instances correspond to a small number of forms, so it is easy to find the correct patterns both from WordNet or related documents. As a result, filtered ranking provides no apparent benefit.

Attack is a more complicated event including several sub-events, which also have a lot of related events like *die* and *injure*. As a result, the document-centric method's performance goes down much faster, because patterns for related event types get drawn in; while the similarity-centric method performs worse than filtered ranking because some ambiguous words are introduced. For example, "hit" is an *attack* trigger, but words in the same Synset, such as "reach", "make", "attain", "gain" are quite dangerous because most of the time, these words do not refer to an attack event.

Start-position events do not appear frequently in ACE 2005, and supervised learning cannot achieve good performance because it can't collect enough training samples. The similarity-centric and Filter2 methods, which also depend on the ACE 2005 corpus, do not perform well either. Filter1 performs quite well because the Gigaword AFP corpus is quite large and contains more relevant documents, although the percentage is very small. This confirms our assumption that filtered ranking can achieve reasonable performance on a large unselected corpus, which is especially useful when the event is rare in the evaluation corpus.

<Arg1 kill Person>
<Arg1 slay Person>
<Arg1 death Person>

Table 2. Seeds for Ace 2005 *Die* evaluation

<Arg0 hire ORG>
<Arg1 hire Person>
<Arg1 appoint Person>
<Arg0 appoint ORG>

Table 3. Seeds for Ace 2005 *Start-Position* evaluation

[6]For word-level evaluation, we only consider trigger words with at least one semantic argument such as subject, object or a preposition; for that reason the performance is quite different from sentence level evaluation. We did the same for the word-level evaluation of semi-supervised learning.

[7] We do not list *Attack* seed patterns here as there are 34 patterns used.

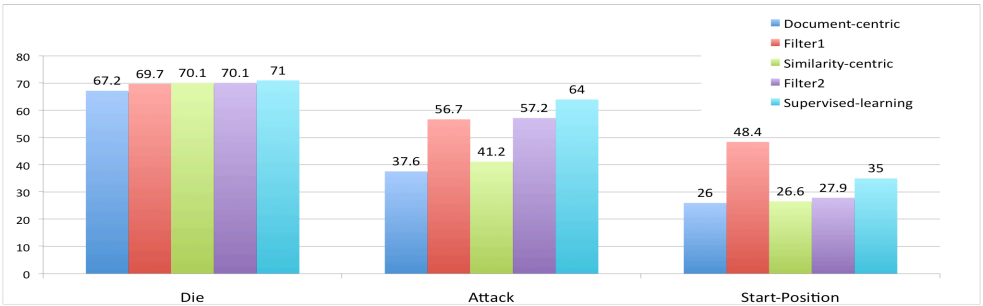

Figure 3. Performance on different ranking methods on ACE2005 sentence level evaluation

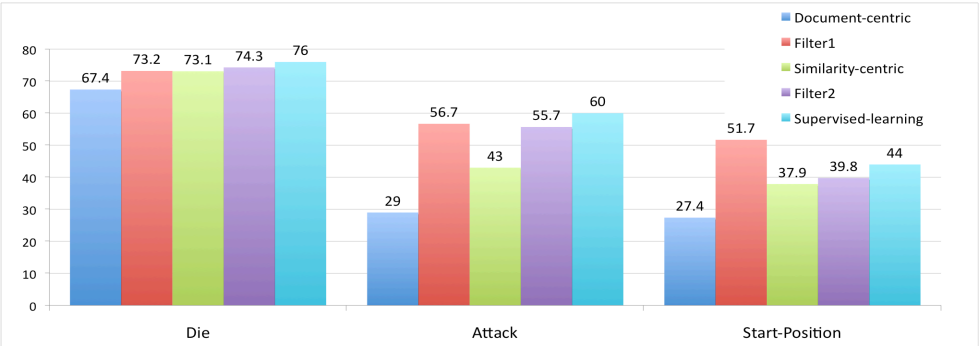

Figure 4. Performance on different ranking methods on ACE2005 word level evaluation

5.4.3 Word-level ACE Event Evaluation

Word-level evaluation is different from sentence-level evaluation because patterns which appear around an event but do not predicate an event are penalized in this evaluation. For example, the pattern *<Sbj chairman PERSON>*, which arises from a phrase like "*PERSON was the chairman of COMPANY*", appears much more in relevant *start-position* sentences than irrelevant sentences, and adding this pattern to the seeds will improve performance using the relevant-sentence metric. We would prefer a metric which discounted such patterns.

As noted above, ACE event annotations contain triggers, which are more specific event locators than a sentence, and we use this as the basis for a more specific evaluation. Extracted patterns are used to identify event triggers instead of identifying relevant sentences. For every word *w* in the ACE corpus, we extract all the patterns whose predicate is *w*. If the event extraction patterns include one of these patterns, we tag *w* as a trigger.

In word level evaluation, document-centric performs worse than the other methods. The reason is that some patterns appear often in the context of an event and are positive patterns for sentence level evaluation, but they do not actually predicate an event and are negative patterns in word level evaluation. In this situation, the document-centric method performs worse than the similarity-centric method, because it extracts many such patterns. For example, of the sentences which contain *die* events, 29% also contain *attack* events.

Thus in word level evaluation, filtered ranking continues to outperform either document- or similarity-centric methods, and its advantage over document-centric methods is accentuated.

6 Conclusions

In this paper, we propose a new ranking method in bootstrapping for event extraction and investigate the performance on different bootstrapping corpora with different ranking methods. This new method can block some irrelevant patterns coming from relevant documents, and, by preferring patterns from relevant documents, can eliminate some lexical ambiguity. Experiments show that this new ranking method performs better than previous ranking methods and is more stable across different corpora.

References

D. Gildea and D. Jurafsky. 2002. *Automatic Labeling of Semantic Roles.* Computational Linguistics, 28:245–288.

MA Greenwood, M. Stevenson. 2006. *Improving semi-supervised acquisition of relation extraction patterns.* Proceedings of the Workshop on Information Extraction Beyond the Document, pages 29–35.

Jay J. Jiang and David W. Conrath. 1997. *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy.* In Proceedings of International Conference Research on Computational Linguistics (ROCLING X), Taiwan

C. Leacock and M. Chodorow. 1998. *Combining local context and WordNet similarity for word sense identification.* In C. Fellbaum, editor, WordNet: An electronic lexical database, pages 265–283. MIT Press.

D. Lin. 1998. *An information-theoretic definition of similarity.* In Proceedings of the International Conference on Machine Learning, Madison, August.

A. Meyers, M. Kosaka, N. Xue, H. Ji, A. Sun, S. Liao and W. Xu. 2009. Automatic Recognition of Logical Relations for English, Chinese and Japanese in the GLARF Framework. *In SEW-2009 (Semantic Evaluations Workshop) at NAACL HLT-2009*

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

Martha Palmer, Dan Gildea, and Paul Kingsbury, *The Proposition Bank: A Corpus Annotated with Semantic Roles,* Computational Linguistics, 31:1, 2005.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. *Using measures of semantic relatedness for word sense disambiguation.* In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico.

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

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. *WordNet::Similarity - Measuring the Relatedness of Concepts.* In Proceedings of the Nineteenth National Conference on Artificial Intelligence (Intelligent Systems Demonstrations), pages 1024-1025, San Jose, CA, July 2004.

P. Resnik. 1995. *Using information content to evaluate semantic similarity in a taxonomy.* In Proceedings of the 14th International Joint Conference on Artificial Intelligence, pages 448–453, Montreal, August.

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

T. Rose, M. Stevenson, and M. Whitehead. 2002. *The Reuters Corpus Volume 1 - from Yesterday's news to tomorrow's language resources.* In LREC-02, pages 827–832, La Palmas, Spain.

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

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2006. *A Hybrid Approach for the Acquisition of Information Extraction Patterns.* Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)

Z. Wu and M. Palmer. 1994. *Verb semantics and lexical selection.* In 32nd Annual Meeting of the Association for Computational Linguistics, pages 133–138, Las Cruces, New Mexico.

Roman Yangarber; Ralph Grishman; Pasi Tapanainen; Silja Huttunen. 2000. *Automatic Acquisition of Domain Knowledge for Information Extraction.* Proc. COLING 2000.

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