# A Bootstrapping Approach to Information Extraction Domain Porting

**Cheng Niu, Wei Li and Rohini K. Srihari**

Cymfony Inc.
600 Essjay Road Williamsville NY14221 USA
{cniu,wei,rohini}@Cymfony.com

## Abstract

This paper presents a seed-driven, bootstrapping approach to domain porting that could be used to customize a generic information extraction (IE) capability for a specific domain. The approach taken is based on the existence of a robust, domain-independent IE engine that can continue to be enhanced, independent of any particular domain. This approach combines the strengths of parsing-based symbolic rule learning and the high performance linear string-based Hidden Markov Model (HMM) to automatically derive a customized IE system with balanced precision and recall. The key idea is to apply precision-oriented symbolic rules learned in the first stage to a large corpus in order to construct an automatically tagged training corpus. This training corpus is then used to train an HMM to boost the recall. The experiments conducted in named entity (NE) tagging and relationship extraction show a performance close to the performance of supervised learning systems.

## Introduction

There are two learning approaches to information extraction, supervised learning and unsupervised or weakly supervised learning. It is generally recognized that there is a knowledge bottleneck for supervised machine learning, since it requires a sizable manually-annotated training corpus. Without a sizable training corpus, the 'sparse data' problem can be so serious as to affect the usability of the trained model. An added difficulty arises from the inconsistency between annotators and quality control of the annotated corpus.

State-of-the-art rule-based systems and supervised learning systems achieve the best performance, especially for NE tagging near human performance can be reached [Krupka and Hausman 1998] [Miller et al., 1998]. However, such systems are difficult for rapid domain porting. They cannot effectively support user-defined IE. The two basic issues facing IE domain porting are:(i) overcoming the performance degradation problem, and (ii) extending IE to capture domain-specific, user-defined information.

As an alternative, more and more researchers are now exploring unsupervised or weakly supervised learning algorithms, e.g. [Yarowsky 1995] [Lin 1998] [Thelen & Riloff 2002]. The advantage of this approach is its ability to make use of the almost unlimited raw corpus in learning a model. If designed properly, the availability of a huge corpus can lead to a high performance model with outstanding resistance to statistical random noise.

[Cucchiarelli & Velardi 2001] discussed boosting the performance of an existing named entity (NE) tagger by unsupervised learning based on parsing structures. [Cucerzan & Yarowsky 1999], [Collins and Singer 1999], and [Kim 2002] presented various techniques using co-training schemes for NE extraction seeded by a small list of proper names or handcrafted NE rules.

As for the bootstrapping system for relationship extraction, [Riloff 1996] described a system which automatically generates parsing-based relationship and event extraction patterns from an untagged corpus. This system requires document classification. [Agichtein & Gravano 2000] proposed a bootstrapping approach for relationship extraction which only requires a few relationship instances (facts)[1] as initial seeds. [Ravichandran & Hovy 2002] use a bootstrapping method that extracts relationships from the web in order to enhance their Question-Answering system.

This paper addresses these issues in the same boot-strapping fashion. Our approach differentiates itself by using a successive learning method that combines precision-oriented symbolic rule learning with recall-oriented HMM. It leverages the parsing capabilities of our domain-independent natural language processing (NLP)/IE system. IE extension is supported by the guidance (i.e. weak supervision) given to this learning process.

Bootstrapped learning involves "seeding" with exemplars from lexical resources, or a small set of easily formulated extraction rules. Such 'seeds' or exemplars represent human guidance in the porting process. Except for the need for a few 'seeds', the learning process is fully automatic. In particular, structure-based unsupervised learning has leveraged InfoXtract parsing capabilities and learned new rules from a parsed corpus, guided by IE 'seeds'. 'Structure' refers to basic phrases constructed by the shallow parser or logical dependency relationships such as logical subject-verb-object (SVO) relationships decoded by the deep parser. A large domain-representative un-annotated corpus is fed to the InfoXtract parser to prepare structural conditions for learning and applying domain-dependent IE rules. In a sense, the corpus is "marked-up" by the full-fledged, domain-independent InfoXtract engine. This provides richer data for learning and enables

[1] Relationship *instance* refers to a unique entity pair that holds a targeted relationship. Note that there may be multiple *mentions* of the same relationship instance in a corpus.

comprehensive domain porting. Unlike conventional linear string-based methods, structure-based learning can filter random noise and handle 'sparse data' more effectively by normalizing the text to disregard superfluous details in word order, phrasing, etc. It results in highly precise rules. Once applied to a large parsed corpus, these rules are used to construct an automatically tagged training corpus which approaches the quality of a human annotated training corpus. In the last stage, the traditional supervised learning algorithm such as HMM training can be applied to generate a high performance IE capability with greatly enhanced recall.

This paper is based on our previous bootstrapping efforts in NE and relationship extraction [Niu et al. 2003a, 2003b]. We generalize and emphasize the commonality of the two efforts and argue that our IE bootstrapping approach is a generally applicable approach to IE domain porting.

## System Design

The foundation for this effort is the Cymfony NLP/IE engine, *InfoXtract* [Srihari et al. 2003]. InfoXtract is a domain-independent, intermediate level IE engine that is equipped with tools for both machine learning and rule writing. The design philosophy for InfoXtract is that the core engine should remain as domain independent as possible; domain specialization or tuning should happen with minimum change to the core engine. More specifically, the IE system remains domain independent at the algorithmic level while permitting domain adaptation to mainly happen at the resource level.

IE domain porting relies heavily on the process of knowledge mining, including IE rule learning and lexical knowledge acquisition. A repository module links knowledge mining modules with the core InfoXtract engine. This defines a process for discovering domain dependent knowledge by applying domain-independent NLP and IE to a raw training corpus in the target domain. The mined knowledge will be fed back into the core engine to support the adaptation of the engine to the new domain, thus creating a domain dependent version of the engine. Our vision is to eventually implement a largely self-learning mechanism whereby an IE engine automatically or semi-automatically adapts to a new domain by 'training' on large volumes of domain dependent raw data.

Figure 1 shows the overall design of the domain porting framework. The enhanced system contains four key components: (i) InfoXtract core engine, (ii) IE Repository, (iii) machine learning module, and (iv) knowledge mining module.

The key idea of unsupervised machine learning for IE is to utilize context redundancy for rule learning. Co-training is the most common way to perform unsupervised learning utilizing context redundancy. The key of co-training is the separation of features into several orthogonal views. In the case of NE classification, usually one view uses context evidence and the other relies on lexicon evidence. Learners corresponding to different views learn from each other iteratively.

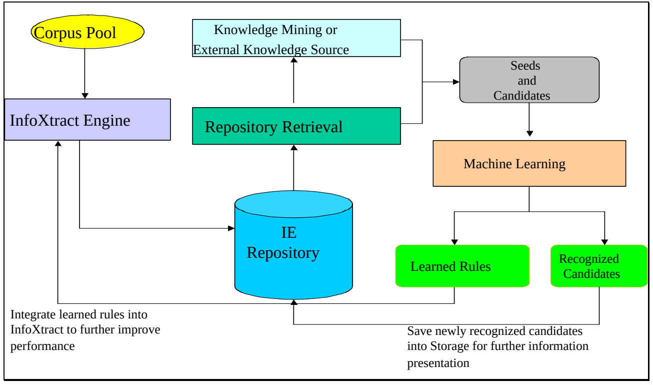

Corpus Pool
Knowledge Mining or External Knowledge Source
Seeds and Candidates
InfoXtract Engine
Repository Retrieval
Machine Learning
IE Repository
Learned Rules
Recognized Candidates
Integrate learned rules into InfoXtract to further improve performance
Save newly recognized candidates into Storage for further information presentation

**Figure 1. Domain Porting Architecture**

One issue of co-training is error propagation in the process of iterative learning. Rule precision drops iteration by iteration. In the early stages, only a few instances are available for learning. This makes some powerful statistical models such as HMM difficult to apply due to the extremely sparse data.

This paper presents a new bootstrapping approach using successive learners to overcome this problem. The first learner makes use of the structural evidences to learn high precision patterns. Then, using a corpus that is automatically tagged by the first learner, an HMM is trained, which greatly improves recall with little precision cost by either generalizing patterns learned by the first learner or exploiting rich surface patterns.

Although the first learner typically suffers from the recall problem, we can apply the learned rules to a huge parsed corpus. In other words, the availability of an almost unlimited raw corpus compensates for the modest recall. As a result, large quantities of IE instances can still be tagged to form a sizable automatically constructed training corpus. This automatically annotated IE corpus can then be used to train the second HMM-based learner to boost the recall.

## Experimental Results and Benchmarks

This section describes our experiments in applying our IE bootstrapping approach to NE classification and relationship detection (experiments on event extraction are in progress). In both cases, we have reached performance which approaches that of supervised learning systems. This verifies the validity and effectiveness of this generic IE porting approach.

### Bootstrapping for NE Domain Porting

This sub-section presents the application of the bootstrapping strategy in NE classification [Niu et al 2003a]. This approach requires only a few common noun/pronoun seeds that correspond to the concept for the target NE type, e.g. he/she/man/woman for PERSON NE. The entire bootstrapping procedure is implemented as training two successive learners: (i) a decision list is used to learn the parsing-based high precision NE rules, and (ii) a Hidden Markov Model is then trained to learn the string sequence-based NE patterns. The second learner uses the training corpus automatically tagged by the first learner. The resulting NE system approaches supervised NE performance and also demonstrates intuitive support for tagging user-defined NE types. The NE bootstrapping is performed as follows:

1. Provide concept-based seeds (by the user)
2. Retrieve parsing structures involving concept-based seeds from the repository to train a decision list for NE classification
3. Apply the learned rules to the NE candidates stored in the repository
4. The proper names tagged in Step 3 and their neighboring words are assembled as an NE annotated corpus
5. Train an HMM based on the annotated corpus.

Five types of structural relationships decoded by our parser are used for parsing-based NE rule learning. These are all directional, binary dependency links between linguistic units: (i) S-V (from logical Subject to Verb); (ii) O-V (from logical Object to Verb); (iii) N-M (from Noun to its adjective modifier); (iv) P-N (from the Possessive noun modifier to head Noun); and (v) IsA: equivalence relation from one NP to another NP.

The concept-based seeds used in the experiments are:

1. PERSON (PER): he, she, his, her, him, man, woman
2. LOCATION (LOC): city, province, town, village
3. ORGANIZATION (ORG): company, firm, organization, bank, airline, army, committee, government, school, university
4. PRODUCT (PRO): car, truck, vehicle, product, plane, aircraft, computer, software, operating system, data-base, book, platform, network

From the parsed corpus in the repository, all instances of concept-based seeds associated with one or more of the five dependency relations were retrieved: a total of 821,267 instances in our experiment. Each seed instance was assigned a concept tag corresponding to an NE. For example, each instance of *he* is marked as PER. The marked instances plus their associated parsing relationships form an annotated NE corpus are shown below:

| | |
|---|---|
| he/PER: | S-V(say) |
| she/PER: | S-V(get) |
| company/ORG: | O-V(compel) |
| city/LOC: | P-N(mayor) |
| car/PRO: | O-V(manufacture) |
| …………………………… | |

Based on this training corpus, the Decision List Learning algorithm [Segal & Etzioni 1994] was used. The accuracy of each rule was evaluated using Laplace smoothing as follows,

Equation 1:

$$accuracy = \frac{positive + 1}{positive + negative + NE\ categoryNo} .$$

A total of 1,290 parsing-based NE rules were learned, with accuracy higher than 0.9. The following are sample rules of the learned decision list:

P-N(wife) → PERSON
S-V(divorce) → PERSON
O-V(deport) → PERSON
N-M(northern) → LOCATION
N-M(non-profit) → ORGANIZATION
P-N(ceo) → ORGANIZATION
N-M(handheld) → PRODUCT
O-V(crash) → PRODUCT
…………………………………..

Due to the unique equivalence nature of the IsA relation, we only need to add the following IsA-based rules to the top of the decision list: IsA(seed) → tag of the seed, e.g. IsA(man) → PERSON.

These parsing-based rules are used to tag a raw corpus in order to train the second NE learner. From the repository, we retrieve all the NE candidates, i.e. noun chunks with proper name POS tags (NNP and NNPS), which are associated with at least one of the five parsing relationships. A total of 1,607,709 NE candidates were retrieved. After applying the decision list to the above, the NE candidates, i.e. 33,104 PER names, 16,426 LOC names, 11,908 ORG names and 6,280 PRO names, were extracted.

In constructing the training corpus, we used the heuristic *one tag per domain for multi-word NE*, in addition to the *one sense per discourse* principle [Gale et al. 1992a] [Gale et al. 1992b]. These heuristics are found to improve the performance of the bootstrapping algorithm for the purpose of both increasing positive instances (i.e. tag propagation) and decreasing the spurious instances (i.e. tag elimination). The tag propagation/elimination scheme is adopted from [Yarowsky 1995]. After this step, a total of 386,614 proper names were recognized, including 134,722 PER names, 186,488 LOC names, 46,231 ORG names and 19,173 PRO names. The overall precision is ~90%. Unlike manually annotated running text corpus, this training corpus consists only of sample string sequences containing the automatically tagged NE instances and their left and right neighboring words within the same sentence. A sample of the automatically constructed corpus is shown below:

in <LOC> Argentina </LOC> .
and <PER> Troy Glaus </PER> walk
call <ORG> Prudential Associates </ORG>
, <PRO> Photoshop </PRO> has
……………………………

This corpus was used for training the second NE learner based on evidence from the string sequence. String sequence-based HMM learning is set as our final goal for NE bootstrapping because of the demonstrated high performance of this type of NE taggers. In this research, a bi-gram HMM is trained, following [Bikel 1997].

We used the same blind testing corpus of 300,000 words containing 20,000 PER, LOC and ORG instances that were truthed in-house originally for benchmarking the existing supervised NE tagger: this has the benefit of precisely measuring performance degradation from supervised learning to unsupervised learning. Our annotators also added the PRO category which is a new tag beyond our original tag set. The benchmarking results are shown below.

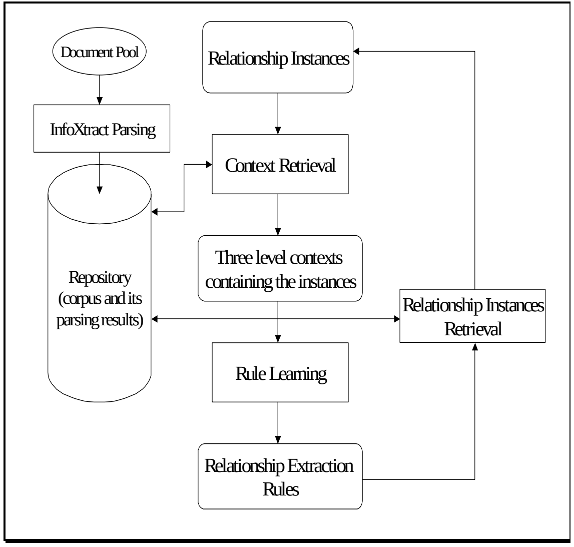
| NE Type | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| PERSON | 86.6% | 88.9% | 87.7% |
| LOCATION | 82.9% | 81.7% | 82.3% |
| ORGANIZATION | 57.1% | 48.9% | 52.7% |
| PRODUCT | 67.3% | 72.5% | 69.8% |

The degradation from supervised learning to weakly supervised learning using the presented bootstrapping method for PER, LOC, and ORG are 5%, 6%, and 34% respectively. This shows that this system approaches the performance of a supervised NE tagger for two of the three proper name NE types in the Message Understanding Conference (MUC), namely, PER NE and LOC NE.

The reason for the poor performance of ORG (~50%) is mainly due to the fact that there are hundreds of sub-types of ORGANIZATION that cannot be covered by less than a dozen concept-based seeds used in our experiment.

## Bootstrapping for Relationship Domain Porting

This sub-section presents our experiment on the successive bootstrapping approach as used in relationship extraction from raw text [Niu et al 2003b]. The bootstrapping procedure consists of two learning phases as shown below.


Document Pool
InfoXtract Parsing
Repository (corpus and its parsing results)
Relationship Instances
Context Retrieval
Three level contexts containing the instances
Relationship Instances Retrieval
Rule Learning
Relationship Extraction Rules

First, symbolic extraction rules are learned after three levels of parsing, namely, (i) post-Named-Entity-tagging (post-NE), (ii) post-shallow-parsing (post-SP), and (iii) post-deep-parsing (post-DP). Then, the HMM is trained to classify whether the post-SP context expresses the targeted relationship. HMM training uses the training corpus automatically tagged by the symbolic rules leaned in the first phase. The resulting HMM is, in effect, a generalization of post-SP symbolic rules, and hence achieves higher recall. Benchmarking shows that the performance of the resulting system approaches supervised learning methods.

This approach has three characteristics: (i) exploration of multi-level parsing contexts; (ii) separation of the pattern generalization process from the pattern extraction process: targeted patterns are extracted during bootstrapped iterative learning, and the learned patterns are subsequently generalized by statistical modeling; and (iii) formulating the generalization task as a language modeling task.

Bootstrapping starts with the input of a few seeds in the form of entity pairs that hold the targeted relationship (e.g. {Microsoft, Redmond} for LOCATION_OF relationship, and {[Abraham Lincoln], [February 12, 1809]} for BIRTHDAY_OF relationship). The learning system then retrieves all sentences containing the entity pairs as the contexts. Three levels of contexts are retrieved: (i) post-NE token sequence; (ii) post-SP unit sequence; and (iii) post-DP dependency trees. In order to capture the linguistic phenomena representing relationships, multi-level rules are desirable since correlated-entity relationships are often expressed in English at different levels of linguistic structure.

In the stage of symbolic rule learning, the patterns of multi-level contexts are evaluated, and patterns with high accuracy are learned as relationship extraction rules. These learned extraction rules are applied to the training corpus in the repository to extract new relationship instances. Then the contexts of new instances are used to learn more extraction rules iteratively. The learning algorithm stops when no new rules can be learned. The algorithm for three-level symbolic rule leaning is as follows:

1. Provide relationship seeds for bootstrapping and put them into a relationship instance set;
2. Retrieve three-level contexts from the repository for each new relationship instance;
3. Construct three-level rule candidates based on the retrieved contexts; evaluate the accuracy of each rule candidate; the rule candidates with accuracy >=0.9 and positive mentions >= 2 are sent into symbolic rule set;
4. Stop if no new symbolic rules are learned; otherwise, apply the new symbolic rules to the corpus in the repository, and put extracted relationship instances into relationship instance set. Then proceed to Step 2.

The symbolic relationship extraction rules have the following format:

Rule =: RuleLevel RulePattern
RuleLevel =: @postNE | @postSP | @postDP

For @postNE rules or @postSP rules, RulePattern is a linear sequence containing the targeted entity pair. For each context in the form of $a_0\, a_1$ @NeX $a_2 \cdots a_3$ @NeY $a_4\, a_5$, nine rule patterns are extracted in rule learning (@NeX and @NeY are the targeted entity pair):

$$\mathrm{@NeX}\ a_1 \cdots a_2\ \mathrm{@NeY}$$
$$a_0\ \mathrm{@NeX}\ a_1 \cdots a_2\ \mathrm{@NeY}$$
$$\mathrm{@NeX}\ a_1 \cdots a_2\ \mathrm{@NeY}\ a_3$$
$$a_0\ \mathrm{@NeX}\ a_1 \cdots a_2\ \mathrm{@NeY}\ a_3$$
$$\ldots\ldots\ldots\ldots$$
$$a_0\ a_1\ \mathrm{@NeX}\ a_2 \cdots a_3\ \mathrm{@NeY}\ a_4\ a_5$$

To evaluate the accuracy of each rule candidate, we define the positive mentions and negative mentions for each as follows. Positive mentions are existing relationship mentions recognized by the rule candidates, and negative mentions are defined as the mentions in conflict with the existing recognized relationships. A negative mention for LOCATION_OF relationship is illustrated as follows: assume that LOCATION_OF *Microsoft* is *Seattle*, then, if a candidate extraction rule identifies a piece of conflicting information that LOCATION_OF *Microsoft* is *Chicago*, the system counts this relationship mention as a negative mention. The accuracy of a rule candidate is computed as follows:

**Equation 2:**

$$accuracy = \frac{positive\ mentions}{positive\ mentions + negative\ mentions}$$

The post-SP contexts associated with the extracted relationship instances are used to train an HMM-based context classifier. The resulting HMM is a generalization of the post-SP context rules learned in the previous stage. Therefore, the recall is significantly enhanced.

The rationale behind using HMM for relationship extraction lies in the equivalence of the relationship extraction task to the binary context classification task, i.e. given a context containing a pair of appropriate entities, determine whether the context expresses the targeted relationship between the two entities.

Post-NE context and post-SP context are linear token sequences. So the context classification for relationship detection is equivalent to the binary token sequence classification. This is a task of language modeling [Jelinek 1997]. HMM is one of the most powerful devices for language modeling. One important feature of the probabilistic finite automata, such as HMM, is the feasibility of learning from positive instances only [Murphy 1996]. This feature makes this new bootstrapping procedure possible since the symbolic extraction rules learned from the first phase can only provide positive instances of contexts.

HMM training for relationship extraction is based on the tagged corpus automatically constructed by applying the learned symbolic extraction rules. All recognized post-SP contexts are retrieved from the repository to form this training corpus for HMM. Each post-SP context is a token sequence. The token sequences start from two tokens before the targeted entity pair and end at two tokens following the entity pair, i.e. $a_0\ a_1\ \mathrm{@NeX}\ a_2 \cdots a_3\ \mathrm{@NeY}\ a_4\ a_5$. The following are some training samples for a seeded entity pair {HP, [Palo Alto, Calif.]}.

@ORG , based in @LOC , revised
Based in @LOC , @ORG won award
medium , @ORG - base @ORG save $1,000,000
…………

Note that @LOC and @ORG are treated as two special token symbols. Given the above post-SP context corpus, a bi-gram HMM [Bikel 1997] is used to estimate the generation probability of any token sequence as a post-SP context expressing the targeted relationship. In the tagging stage, for each candidate entity pair, the $\Pr(W)$ of the corresponding post-SP context W is computed. Only when the associated perplexity of the context is higher than a predefined threshold will the context be recognized as expressing the relationship.

The following four entity-pair seeds for the LOCATION_OF relationship are used in our experiment:

{[Microsoft], [Seattle]}
{[Microsoft], [Redmond, Wash.]}
{[IBM], [Armonk, N.Y.]}
{[Office Depot], [Delray Beach, Fla.]}

In 88,000,000 words of corpus, 172,575 candidate sentences are found containing at least one mention of ORG and one mention of LOC. Using the above four seeds, the system has learned 3,818 symbolic rules which extract 7,645 LOCATION_OF relationships from the candidate sentences before it stops learning new rules after 14 iterations. Among the 3,818 extraction rules, there are 1,845 post-NE rules, 1,848 post-SP rules and 125 post-DP rules. Some sample rules are shown below:

@postNE @ORG , in @LOC
@postSP @LOC – base @ORG

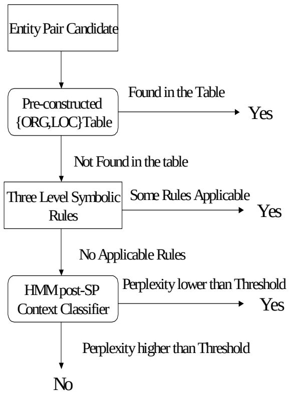
To evaluate the system performance, two types of precision and recall are defined: Retrieval Precision/Recall and IE Precision/Recall. Following [Agichtein & Gravano 2000], the Retrieval Precision/Recall measures are based on counting the relationship instances. The IE Precision/Recall measures are based on counting mentions of relationships. To evaluate the Retrieval Precision of the learned relationship extraction rules, 1,000 recognized relationship instances are randomly selected for manual checking. The results are shown below.

| | |
|---|---|
| Extracted Relationship Instances | 1,000 |
| Correct | 884 |
| Error due to Named Entity Tagging | 73 |
| Error due to Relationship Extraction | 43 |
| Retrieval Precision | 88% |

To evaluate Retrieval Recall of the learned relationship extraction rules, we selected 1,000 entity pairs that hold the LOCATION_OF relationship. Among the 1,000 testing pairs, 378 are extracted by the symbolic rules. Therefore, the Retrieval-Recall of the symbolic rules is 38%.

To evaluate the IE Precision/Recall, we build a relationship extraction module containing three components as shown below.

To benchmark IE Precision/Recall, our parser processed a testing corpus and extracted 50,000 sentences containing LOC and ORG. The extracted 50,000 sentences were processed by the above procedure. IE Precision benchmarking was performed by manually checking 1,000 of the recognized relationship mentions. For IE Recall, we selected the first 1,000 sentences that contain the LOCATION_OF relationship. The benchmarking results are shown below.


Entity Pair Candidate
Pre-constructed {ORG,LOC}Table
Found in the Table
Yes
Not Found in the table
Three Level Symbolic Rules
Some Rules Applicable
Yes
No Applicable Rules
HMM post-SP Context Classifier
Perplexity lower than Threshold
Yes
Perplexity higher than Threshold
No

| IE Precision | IE Recall | F-measure |
|---|---|---|
| 82% | 69% | 75% |

Using kernel-based supervised machine learning, [Zelenko et al. 2002] report an F-measure of 83% for LOCATION_OF relationships. Our new bootstrapping method achieves an F-measure of 75%, which is approaching the performance of supervised machine learning.

## Conclusion

We have presented an effective IE bootstrapping approach based on successive learning. This approach combines symbolic rule learning and HMM training for enhanced precision and recall. The resulting HMMs in the experiments of NE and relationship extraction reach a performance close to supervised systems.

**Acknowledgement**: This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-03-C-0044. Thanks to Carrie Pine of AFRL for supporting and reviewing this work.

## References

Agichtein, E. & Gravano, L. 2000. Snowball: Extracting Relations from Large Plain-Text Collections. Proceedings of the 5th ACM International Conference on Digital Libraries. San Antonio.

Aone, A. & M. Ramos-Santacruz 2000. REES: A Large-Scale Relation and Event Extraction System. Proceedings of ANLP-NAACL 2000, Seattle.

Bikel, D. M. 1997. Nymble: a high-performance learning name-finder. Proceedings of the Fifth Conference on ANLP: 194-201, Morgan Kaufmann Publishers.

Borthwick, A. et al. 1998. Description of the MENE named Entity System. Proceedings of MUC-7.

Collins, M. and Y. Singer. 1999. Unsupervised Models for Named Entity Classification. Proceedings of the Joint SIGDAT Conference on EMNLP and VLC.

Cucchiarelli, A. and P. Velardi. 2001. Unsupervised Named Entity Recognition Using Syntactic and Se-mantic Contextual Evidence. Computational Linguistics, Volume 27, Number 1, 123-131.

Cucerzan, S. and D. Yarowsky. 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC, 90-99.

Gale, W., K. Church, and D. Yarowsky. 1992. One Sense Per Discourse. Proceedings of the 4th DARPA Speech and Natural Language Workshop. 233-237.

Jelinek, F. 1997. Statistical Methods for Speech Recognition. The MIT Press.

Kim, J., I. Kang, and K. Choi. 2002. Unsupervised Named Entity Classification Models and their Ensembles. COLING 2002.

Krupka, G. R. and K. Hausman. 1998. IsoQuest Inc: Description of the NetOwl Text Extraction System as used for MUC-7. Proceedings of MUC-7.

Lin, D.K. 1998. Automatic Retrieval and Clustering of Similar Words. COLING-ACL 1998.

MUC-7, 1998. Proceedings of the Seventh Message Understanding Conference (MUC-7).

Segal, R. and O. Etzioni. 1994. Learning decision lists using homogeneous rules. Proceedings of the 12th National Conference on Artificial Intelligence.

Srihari, R., C. Niu, & W. Li. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. Proceedings of ANLP 2000, Seattle.

Murphy, K. P. 1996. Passively Learning Finite Automata. Technical Report, Santa Fe Institute.

Niu, C., W. Li, J. Ding, and R. Srihari 2003a.. A Bootstrapping Approach to Named Entity Classification using Successive Learners. In *Proceedings of ACL 2003*. Sapporo, Japan. pp. 335-342

Niu, C., W. Li, R. Srihari, and L. Crist 2003b. Bootstrapping a Hidden Markov Model for Relationship Extraction Using Multi-level Contexts. In *Proceedings of Pacific Association for Computational Linguistics 2003* (PACLING03). Halifax, Nova Scotia, Canada

Ravichandran, D. & E. Hovy, 2002. Learning surface text patterns for a Question Answering System. ACL-2002.

Riloff, E. 1996. Automatically Generating Extraction Patterns from Untagged Text. AAAI-96: 1044-1049.

Srihari, R., W. Li, C. Niu and T. Cornell. 2003. InfoXtract: A Customizable Intermediate Level Information Extraction Engine. In *Proceedings of HLT/NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems* (SEALTS). pp. 52-59, Edmonton, Canada.

Thelen, M. and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. Proceedings of EMNLP 2002.

Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. ACL-1995.

Zelenko, D., C. Aone & A. Richardella. 2002. Kernel Methods for Relation Extraction. EMNLP-2002.