

Unsupervised Information Extraction Approach Using Graph Mutual Reinforcement

Hany Hassan

Ahmed Hassan

Ossama Emam

IBM Cairo Technology Development Center
Giza, Egypt
P.O. Box 166 Al-Ahram

hanyh@eg.ibm.com

hasanah@eg.ibm.com

emam@eg.ibm.com

Abstract

Information Extraction (IE) is the task of extracting knowledge from unstructured text. We present a novel unsupervised approach for information extraction based on graph mutual reinforcement. The proposed approach does not require any seed patterns or examples. Instead, it depends on redundancy in large data sets and graph based mutual reinforcement to induce generalized “extraction patterns”. The proposed approach has been used to acquire extraction patterns for the ACE (Automatic Content Extraction) Relation Detection and Characterization (RDC) task. ACE RDC is considered a hard task in information extraction due to the absence of large amounts of training data and inconsistencies in the available data. The proposed approach achieves superior performance which could be compared to supervised techniques with reasonable training data.

1 Introduction

In this paper we propose a novel, and completely unsupervised approach for information extraction. We present a general technique; however we focus on relation extraction as an important task of Information Extraction. The approach depends on constructing generalized extraction patterns, which could match many instances, and deploys graph based mutual reinforcement to weight the importance of these patterns. The mutual reinforcement is used to automatically iden-

tify the most informative patterns, where patterns that match many instances tend to be correct. Similarly, instances matched by many patterns tend to be correct. The intuition is that large unsupervised data is redundant, i.e. different instances of information could be found many times in different contexts and by different representation. The problem can therefore be seen as hubs (instances) and authorities (patterns) problem which can be solved using the Hypertext Induced Topic Selection (HITS) algorithm (Kleinberg, 1998).

HITS is an algorithmic formulation of the notion of authority in web pages link analysis, based on a relationship between a set of relevant “authoritative pages” and a set of “hub pages”. The HITS algorithm benefits from the following observation: when a page (hub) links to another page (authority), the former confers authority over the latter.

By analogy to the authoritative web pages problem, we could represent the patterns as authorities and instances as hubs, and use mutual reinforcement between patterns and instances to weight the most authoritative patterns. Highly weighted patterns are then used in extracting information.

The proposed approach does not need any seeds or examples. Human involvement is only needed in determining the entities of interest; the entities among which we are seeking relations.

The paper proceeds as follows: in Section 2 we discuss previous work followed by a brief definition of our general notation in Section 3. A detailed description of the proposed approach then follows in Section 4. Section 5 discusses the application of the proposed approach to the prob-

lem of detecting semantic relations from text. Section 6 discusses experimental results while the conclusion is presented in Section 7.

2 Previous Work

Most of the previous work on Information Extraction (IE) focused on supervised learning. Relation Detection and Characterization (RDC) was introduced in the Automatic Content Extraction Program (ACE) (ACE, 2004). The approaches proposed to the ACE RDC task such as kernel methods (Zelenko et al., 2002) and Maximum Entropy methods (Kambhatla, 2004) required the availability of large set of human annotated corpora which are tagged with relation instances. However human annotated instances are limited, expensive, and time consuming to obtain, due to the lack of experienced human annotators and the low inter-annotator agreements.

Some previous work adopted weakly supervised or unsupervised learning approaches. These approaches have the advantage of not needing large tagged corpora but need seed examples or seed extraction patterns. The major drawback of these approaches is their dependency on seed examples or seed patterns which may lead to limited generalization due to dependency on handcrafted examples. Some of these approaches are briefed here:

(Brin,98) presented an approach for extracting the authorship information as found in books description on the World Wide Web. This technique is based on dual iterative pattern relation extraction wherein a relation and pattern set is iteratively constructed. This approach has two major drawbacks: the use of handcrafted seed examples to extract more examples similar to these handcrafted seed examples and the use of a lexicon as the main source for extracting information.

(Blum and Mitchell, 1998) proposed an approach based on co-training that uses unlabeled data in a particular setting. They exploit the fact that, for some problems, each example can be described by multiple representations.

(Riloff & Jones, 1999) presented the Meta-Bootstrapping algorithm that uses an unannotated training data set and a set of seeds to learn a dictionary of extraction patterns and a domain specific semantic lexicon. Other works tried to exploit the duality of patterns and their extractions for the purpose of inferring the semantic class of words like (Thelen & Riloff, 2002) and (Lin et al, 2003).

(Muslea et al., 1999) introduced an inductive algorithm to generate extraction rules based on user labeled training examples. This approach suffers from the labeled data bottleneck.

(Agichtein et. al, 2000) presented an approach using seed examples to generate initial patterns and to iteratively obtain further patterns. Then ad-hoc measures were deployed to estimate the relevancy of the patterns that have been newly obtained. The major drawbacks of this approach are: its dependency on seed examples leads to limited capability of generalization, and the estimation of patterns relevancy requires the deployment of ad-hoc measures.

(Hasegawa et. al. 2004) introduced unsupervised approach for relation extraction depending on clustering context words between named entities; this approach depends on ad-hoc context similarity between phrases in the context and focused on certain types of relations.

(Etzioni et al, 2005) proposed a system for building lists of named entities found on the web. Their system uses a set of eight domain-independent extraction patterns to generate candidate facts.

All approaches, proposed so far, suffer from either requiring large amount of labeled data or the dependency on seed patterns (or examples) that result in limited generalization.

3 General Notation

In graph theory, a graph is a set of objects called vertices joined by links called edges. A bipartite graph, also called a bigraph, is a special graph where the set of vertices can be divided into two disjoint sets with no two vertices of the same set sharing an edge.

The Hypertext Induced Topic Selection (HITS) algorithm is an algorithm for rating, and therefore ranking, web pages. The HITS algorithm makes use of the following observation: when a page (hub) links to another page (authority), the former confers authority over the latter. HITS uses two values for each page, the "authority value" and the "hub value". "Authority value" and "hub value" are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that authority. A hub value is the sum of the scaled authority values of the authorities it points to.

A template, as we define for this work, is a sequence of generic forms that could generalize

over the given instances. An example template is:

```
GPE POS (PERSON)+

GPE: Geographical Political Entity
POS: possessive ending
PERSON: PERSON Entity
```

This template could match the sentence: “France’s President Jacque Chirac...”. This template is derived from the representation of the Named Entity tags, Part-of-Speech (POS) tags and semantic tags. The choice of the template representation here is for illustration purpose only; any combination of tags, representations and tagging styles might be used.

A pattern is more specific than a template. A pattern specifies the role played by the tags (first entity, second entity, or relation). An example of a pattern is:

```
GPE (E2) POS (PERSON)+(E1)
```

This pattern indicates that the word(s) with the tag GPE in the sentence represents the second entity (Entity 2) in the relation, while the word(s) tagged PERSON represents the first entity (Entity 1) in this relation, the “+” symbol means that the (PERSON) entity is repetitive (i.e. may consist of several tokens).

A tuple, in our notation during this paper, is the result of the application of a pattern to unstructured text. In the above example, one result of applying the pattern to some raw text is the following tuple:

```
Entity 1: Jacque Chirac
Entity 2: France
Relation: EMP-Executive
```

4 The Approach

The unsupervised graph-based mutual reinforcement approach, we propose, depends on the construction of generalized “extraction patterns” that could match many instances. The patterns are then weighted according to their importance by deploying graph based mutual reinforcement techniques. This duality in patterns and extracted information (tuples) could be stated that patterns could match different tuples, and tuples in turn could be matched by different patterns. The proposed approach is composed of two main steps namely, initial patterns construction and pattern

weighting or induction. Both steps are detailed in the next sub-sections.

4.1 Initial Patterns Construction

As shown in Figure 1, several syntactic, lexical, and semantic analyzers could be applied to the unstructured text. The resulting analyses could be employed in the construction of extraction patterns. It is worth mentioning that the proposed approach is general enough to accommodate any pattern design; the introduced pattern design is for illustration purposes only.

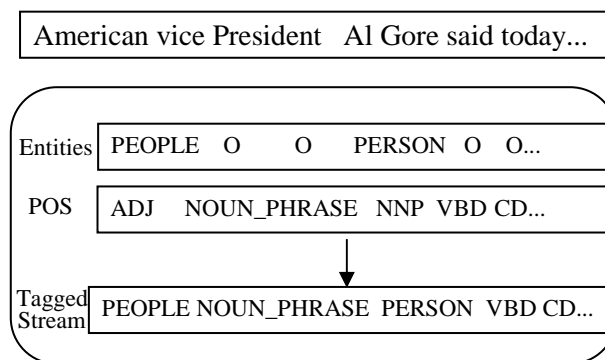


Figure 1: An example of the output of analyzers applied to the unstructured text

Initially, we need to start with some templates and patterns to proceed with the induction process. Relatively large amount of text data is tagged with different taggers to produce the previously mentioned patterns styles. An n-gram language model is built on this data and used to construct weighted finite state machines.

Paths with low cost (high language model probabilities) are chosen to construct the initial set of templates; the intuition is that paths with low cost (high probability) are frequent and could represent potential candidate patterns.

The resulting initial set of templates is applied to a very large text data to produce all possible patterns. The number of candidate initial patterns could be reduced significantly by specifying the candidate types of entities; for example we might specify that the first entity could be PEROSN or PEOPLE while the second entity could be ORGANIZATION, LOCATION, COUNTRY and etc...

The candidate patterns are then applied to the tagged stream and the unstructured text to collect a set of patterns and matched tuples pairs.

The following procedure briefs the Initial Pattern Construction Step:

- Select a random set of text data.

- Apply various taggers on text data and construct templates style.
- Build n-gram language model on template style data.
- Construct weighted finite state machines from the n-gram language model.
- Choose n-best paths in the finite state machines.
- Use best paths as initial templates.
- Apply initial templates on large text data.
- Construct initial patterns and associated tuples sets.

4.2 Pattern Induction

The inherent duality in the patterns and tuples relation suggests that the problem could be interpreted as a hub authority problem. This problem could be solved by applying the HITS algorithm to iteratively assign authority and hub scores to patterns and tuples respectively.

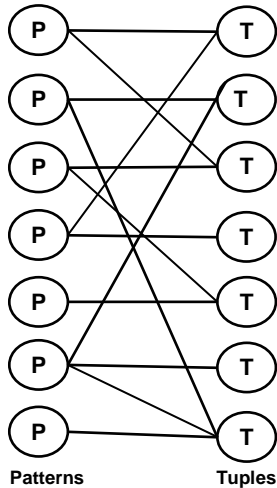


Figure 2: A bipartite graph representing patterns and tuples

Patterns and tuples are represented by a bipartite graph as illustrated in figure 2. Each pattern or tuple is represented by a node in the graph. Edges represent matching between patterns and tuples. The pattern induction problem can be formulated as follows: Given a very large set of data D containing a large set of patterns P which match a large set of tuples T , the problem is to identify \tilde{P} , the set of patterns that match the set of the most correct tuples \tilde{T} . The intuition is

that the tuples matched by many different patterns tend to be correct and the patterns matching many different tuples tend to be good patterns. In other words; we want to choose, among the large space of patterns in the data, the most informative, highest confidence patterns that could identify correct tuples; i.e. choosing the most “authoritative” patterns in analogy with the hub authority problem. However, both \tilde{P} and \tilde{T} are unknown. The induction process proceeds as follows: each pattern p in P is associated with a numerical authority weight a_v which expresses how many tuples match that pattern. Similarly, each tuple t in T has a numerical hub weight h_t which expresses how many patterns were matched by this tuple. The weights are calculated iteratively as follows:

$$a^{(i+1)}(p) = \frac{\sum_{u=1}^{T(p)} h^{(i)}(u)}{H^{(i)}} \quad (1)$$

$$h^{(i+1)}(t) = \frac{\sum_{u=1}^{P(t)} a^{(i)}(u)}{A^{(i)}} \quad (2)$$

where $T(p)$ is the set of tuples matched by p , $P(t)$ is the set of patterns matching t , $a^{(i+1)}(p)$ is the authoritative weight of pattern p at iteration $(i+1)$, and $h^{(i+1)}(t)$ is the hub weight of tuple t at iteration $(i+1)$. $H^{(i)}$ and $A^{(i)}$ are normalization factors defined as:

$$H^{(i)} = \sum_{p=1}^{|P|} \sum_{u=1}^{T(p)} h^{(i)}(u) \quad (3)$$

$$A^{(i)} = \sum_{t=1}^{|T|} \sum_{u=1}^{P(t)} a^{(i)}(u) \quad (4)$$

Highly weighted patterns are identified and used for extracting relations.

4.3 Tuple Clustering

The tuple space should be reduced to allow more matching between pattern-tuple pairs. This space reduction could be accomplished by seeking a tuple similarity measure, and constructing a weighted undirected graph of tuples. Two tuples are linked with an edge if their similarity measure exceeds a certain threshold. Graph clustering algorithms could be deployed to partition the graph into a set of homogeneous communities or clusters. To reduce the space of tuples, we seek a matching criterion that group similar tuples together. Using WordNet, we can measure the semantic similarity or relatedness between a pair of concepts (or word senses), and by extension, between a pair of sentences. We use the similarity

measure described in (Wu and Palmer, 1994) which finds the path length to the root node from the least common subsumer (LCS) of the two word senses which is the most specific word sense they share as an ancestor. The similarity score of two tuples, S_T , is calculated as follows:

$$S_T = \sqrt{S_{E1}^2 + S_{E2}^2} \quad (5)$$

where S_{E1} , and S_{E2} are the similarity scores of the first entities in the two tuples, and their second entities respectively.

The tuple matching procedure assigns a similarity measure to each pair of tuples in the dataset. Using this measure we can construct an undirected graph G . The vertices of G are the tuples. Two vertices are connected with an edge if the similarity measure between their underlying tuples exceeds a certain threshold. It was noticed that the constructed graph consists of a set of semi isolated groups as shown in figure 3. Those groups have a very large number of inter-group edges and meanwhile a rather small number of intra-group edges. This implies that using a graph clustering algorithm would eliminate those weak intra-group edges and produce separate groups or clusters representing similar tuples. We used Markov Cluster Algorithm (MCL) for graph clustering (Dongen, 2000). MCL is a fast and scalable unsupervised clustering algorithm for graphs based on simulation of stochastic flow.

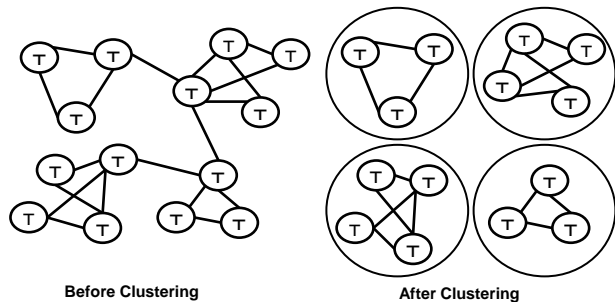


Figure 3: Applying Clustering Algorithms to Tuple graph

An example of a couple of tuples that could be matched by this technique is:

```
United States(E2) presi-
dent(E1)
US(E2) leader(E1)
```

A bipartite graph of patterns and tuple clusters is constructed. Weights are assigned to patterns and tuple clusters by iteratively applying the

HITS algorithm and the highly ranked patterns are then used for relation extraction.

5 Experimental Setup

5.1 ACE Relation Detection and Characterization

In this section, we describe Automatic Content Extraction (ACE). ACE is an evaluation conducted by NIST to measure Entity Detection and Tracking (EDT) and Relation Detection and Characterization (RDC). The EDT task is concerned with the detection of mentions of entities, and grouping them together by identifying their coreference. The RDC task detects relations between entities identified by the EDT task. We choose the RDC task to show the performance of the graph based unsupervised approach we propose. To this end we need to introduce the notion of mentions and entities. Mentions are any instances of textual references to objects like people, organizations, geopolitical entities (countries, cities ...etc), locations, or facilities. On the other hand, entities are objects containing all mentions to the same object. Here, we present some examples of ACE entities and relations:

Spain's Interior Minister announced this evening the arrest of separatist organization Eta's presumed leader Ignacio Garcia Arregui. Arregui, who is considered to be the Eta organization's top man, was arrested at 17h45 Greenwich. The Spanish judiciary suspects Arregui of ordering a failed attack on King Juan Carlos in 1995.

In this fragment, all the underlined phrases are mentions to "Eta" organization, or to "Garcia Arregui". There is a management relation between "leader" which references to "Garcia Arregui" and "Eta".

5.2 Patterns Construction and Induction

We used the LDC English Gigaword Corpus, AFE source from January to August 1996 as a source for unstructured text. This provides a total of 99475 documents containing 36 M words. In the performed experiments, we focus on two types of relations EMP-ORG relations and GPE-AFF relations which represent almost 50% of all relations in RDC – ACE task.

POS (part of speech) tagger and mention tagger were applied to the data, the used pattern design consists of a mix between the part of speech (POS) tags and the mention tags for the words in the unsupervised data. We use the mention tag, if it exists; otherwise we use the part of speech tag. An example of the analyzed text and the presumed associated pattern is shown:

```
Text: Eta's presumed leader
Arregui ...
Pos: NNP POS JJ NN NNP
Mention: ORG 0 0 0 PERSON
Pattern: ORG(E2) POS JJ
NN(R) PERSON(E1)
```

An n-gram language model, 5-gram model and back off to lower order n-grams, was built on the data tagged with the described patterns' style. Weighted finite states machines were constructed with the language model probabilities. The n-best paths, 20 k paths, were identified and deployed as the initial template set. Sequences that do not contain the entities of interest, and hence cannot represent relations, were automatically filtered out. This resulted in an initial templates set of around 3000 element. This initial templates set was applied on the text data to establish initial patterns and tuples pairs. Graph based mutual reinforcement technique was deployed with 10 iterations on the patterns and tuples pairs to weight the patterns.

We conducted two groups of experiments, the first with simple syntactic tuple matching, and the second with semantic tuple clustering as described in section 4.3

6 Results and Discussion

We compare our results to a state-of-the-art supervised system similar to the system described in (Kambhatla, 2004). Although it is unfair to make a comparison between a supervised system and a completely unsupervised system, we chose to make this comparison to test the performance of the proposed unsupervised approach on a real task with defined test set and state-of-the-art performance. The supervised system was trained on 145 K words which contain 2368 instances of the two relation types we are considering.

The system performance is measured using precision, recall and F-Measure with various amounts of induced patterns. Table 1 presents the precision, recall and F-measure for the two relations using the presented approach with the utili-

zation of different amount of highly weighted patterns. Table 2 presents the same results using semantic tuple matching and clustering, as described in section 4.3.

No. of Patterns	Precision	Recall	F-Measure
1500	35.9	66.3	46.58
1000	41.2	59.7	48.75
700	43.1	58.1	49.49
500	46	56.5	50.71
400	46.9	52.9	49.72
200	50.1	44.9	47.36

Table 1: The effect of varying the number of induced patterns on the system performance (syntactic tuple matching)

No. of Patterns	Precision	Recall	F-Measure
1500	36.1	67.2	46.97
1000	43.7	59.6	50.43
700	44.1	59.3	50.58
500	46.3	57.2	51.18
400	47.3	57.6	51.94
200	48.1	45.9	46.97

Table 2: The effect of varying the number of induced patterns on the system performance (semantic tuple matching)

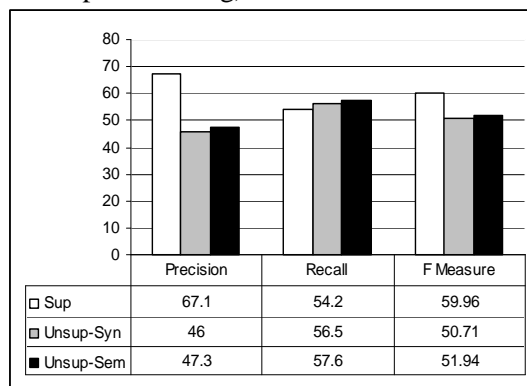


Figure 4: A comparison between the supervised system (Sup), the unsupervised system with syntactic tuple matching (Unsup-Syn), and with semantic tuple matching (Unsup-Sem)

Best F-Measure is achieved using relatively small number of induced patterns (400 and 500 patterns) while using more patterns increases the recall but degrades the precision.

Table 2 indicates that the semantic clustering of tuples did not provide significant improve-

ment; although better performance was achieved with less number of patterns (400 patterns). We think that the deployed similarity measure and it needs further investigation to figure out the reason for that.

Figure 4 presents the comparison between the proposed unsupervised systems and the reference supervised system. The unsupervised systems achieves good results even in comparison to a state-of-the-art supervised system.

Sample patterns and corresponding matching text are introduced in Table 3 and Table 4. Table 3 shows some highly ranked patterns while Table 4 shows examples of low ranked patterns.

Pattern	Matches
GPE (PERSON)+	Peruvian President Alberto Fujimori
GPE (PERSON)+	Zimbabwean President Robert Mugabe
GPE (PERSON)+	PLO leader Yasser Arafat
GPE POS (PERSON)+	Zimbabwe 's President Robert Mugabe
GPE JJ PERSON	American clinical neuropsychologist
GPE JJ PERSON	American diplomatic personnel
PERSON IN JJ GPE	candidates for local government
ORGANIZATION PERSON	Airways spokesman
ORGANIZATION PERSON	Ajax players
PERSON IN DT (ORGANIZATION)+	chairman of the opposition parties
(ORGANIZATION)+ PERSON	opposition parties chairmans

Table3: Examples of patterns with high weights

Pattern	Matches
GPE CC (PERSON)+	Barcelona and Johan Cruyff
GPE , CC PERSON	Paris , but Riccardi
GPE VBZ VBN PERSON	Pyongyang has accepted Gallucci
GPE VBZ VBN PERSON	Russia has abandoned us
GPE VBZ VBN P PERSON	Rwanda 's defeated Hutu
GPE VBZ VBN PERSON	state has pressed Arafat
GPE VBZ VBN TO VB PERSON	Taiwan has tried to keep Lee
(PERSON)+ VBD GPE ORGANIZATION	Alfred Streim told German radio
(PERSON)+ VBD GPE ORGANIZATION	Dennis Ross met Syrian army
(PERSON)+ VBD GPE ORGANIZATION	Van Miert told EU industry

Table4: Examples of patterns with low weights

7 Conclusion and Future Work

In this work, a general framework for unsupervised information extraction based on mutual reinforcement in graphs has been introduced. We construct generalized extraction patterns and deploy graph based mutual reinforcement to automatically identify the most informative patterns. We provide motivation for our approach from a graph theory and graph link analysis perspective. Experimental results have been presented supporting the applicability of the proposed approach to ACE Relation Detection and Characterization (RDC) task, demonstrating its applicability to hard information extraction problems. The proposed approach achieves remarkable results comparable to a state-of-the-art supervised system, achieving 51.94 F-measure compared to 59.96 F-measure of the state-of-the-art supervised system which requires huge amount of human annotated data. The proposed approach represents a powerful unsupervised technique for information extraction in general and particularly for relations extraction that requires no seed patterns or examples and achieves significant performance.

In our future work, we plan to focus on generalizing the approach for targeting more NLP problems.

8 Acknowledgements

We would like to thank Salim Roukos for his invaluable suggestions and support. We would also like to thank Hala Mostafa for helping with the early investigation of this work. Finally we would like to thank the anonymous reviewers for their constructive criticism and helpful comments.

References

- ACE. 2004. The NIST ACE evaluation website. <http://www.nist.gov/speech/tests/ace/>
- Eugene Agichtein and Luis Gravano. 2000. *Snowball: Extracting Relations from Large Plain-Text Collections*. Proceedings of the 5th ACM Conference on Digital Libraries (DL 2000).
- Sergy Brin. 1998. *Extracting Patterns and Relations from the World Wide Web*. Proceedings of the 1998 International Workshop on the Web and Databases"
- Stijn van Dongen. 2000. *A Cluster Algorithm for Graphs*. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands.

- Stijn van Dongen. 2000. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht
- Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. *Web-scale information extraction in KnowItAll (preliminary results)*. In Proceedings of the 13th World Wide Web Conference, pages 100-109.
- Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. *Unsupervised Named-Entity Extraction from the Web: An Experimental Study*. Artificial Intelligence, 2005.
- Radu Florian, Hany Hassan, Hongyan Jing, Nanda Kambhatla, Xiaqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. *A Statistical Model for multilingual entity detection and tracking*. Proceedings of the Human Language Technologies Conference (HLT-NAACL 2004).
- Dayne Freitag, and Nicholas Kushmerick. 2000. *Boosted wrapper induction*. The 14th European Conference on Artificial Intelligence Workshop on Machine Learning for Information Extraction
- Rayid Ghani and Rosie Jones. 2002. *A Comparison of Efficacy and Assumptions of Bootstrapping Algorithms for Training Information Extraction Systems*. Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Data at the Linguistic Resources and Evaluation Conference (LREC 2002).
- Takaaki Hasegawa, Satoshi Sekine, Ralph Grishman. 2004. *Discovering Relations among Named Entities from Large Corpora*. Proceedings of The 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004).
- Taher Haveliwala. 2002. *Topic-sensitive PageRank*. Proceedings of the 11th International World Wide Web Conference
- Thorsten Joachims. 2003. *Transductive Learning via Spectral Graph Partitioning*. Proceedings of the International Conference on Machine Learning (ICML 2003).
- Nanda Kambhatla. 2004. *Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction*. Proceedings of The 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004).
- John Kleinberg. 1998. *Authoritative Sources in a Hyperlinked Environment*. Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms.
- N. Kushmerick, D.S. Weld, R.B. Doorenbos. 1997. *Wrapper Induction for Information Extraction*. Proceedings of the International Joint Conference on Artificial Intelligence.
- Winston Lin, Roman Yangarber, Ralph Grishman. 2003. *Bootstrapped Learning of Semantic Classes from Positive and Negative Examples*. Proceedings of the 20th International Conference on Machine Learning (ICML 2003) Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining.
- Ion Muslea, Steven Minton, and Craig Knoblock. 1999. *A hierarchical approach to wrapper induction*. Proceedings of the Third International Conference on Autonomous Agents.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. *WordNet::Similarity - Measuring the Relatedness of Concepts*. Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2004)
- Ellen Riloff and Rosie Jones. 2003. *Learning dictionaries for information extraction by multilevel bootstrapping*. Proceedings of the Sixteenth national Conference on Artificial Intelligence (AAAI 1999).
- Michael Thelen and Ellen Riloff. 2002. *A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts*. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002).
- Scott White, and Padhraic Smyth. 2003. *Algorithms for Discovering Relative Importance in Graphs*. Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
- Zhibiao Wu, and Martha Palmer. 1994. *Verb semantics and lexical selection*. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994).
- Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. *Semi-supervised Learning using Gaussian Fields and Harmonic Functions*. Proceedings of the 20th International Conference on Machine Learning (ICML 2003).