

# Unsupervised Induction of IE Domain Knowledge using an Ontology

Mark Stevenson

Department of Computer Science  
University of Sheffield  
Sheffield  
S1 4DP, UK  
marks@dcs.shef.ac.uk

## Abstract

The development of systems which can be easily adapted to new domains is an important goal in current Information Extraction (IE) research. Machine learning algorithms have been applied to the problem but supervised algorithms often require large amounts of examples and unsupervised ones may be hampered by a lack of information. This paper presents an unsupervised algorithm which makes use of the WordNet ontology to compensate for the small number of examples. Comparative evaluation with a previously reported approach shows that the algorithm presented here is in some ways preferable and that benefits can be gained from combining the two approaches.

## Introduction

One of the goals of current research in Information Extraction (IE) is to develop systems which can be easily ported to new domains with the minimum of human intervention. Early IE systems were generally based on knowledge engineering approaches and often proved difficult to adapt to new domains. For example, (Lehnert *et al.* 1992) reported their system required around 1,500 person-hours of expert labour to modify for a new extraction task with much of the effort being in domain knowledge acquisition. A promising approach is to make use of machine learning techniques to automate the knowledge acquisition process. However, supervised learning techniques often require large amounts of annotated text which may also require large amounts of expert effort to produce. For example, (Miller *et al.* 1998) reported an unsupervised algorithm<sup>1</sup> which required an annotated training corpus of 790,000 words. Another approach is to make use of unsupervised machine learning techniques which have the ability to generalise from a few indicative examples. One of the advantages of unsupervised algorithms is that they do not require as much annotated data as supervised

approaches, however this means that the algorithm has access to less information and this may have a detrimental effect on the learning performance. One solution to this problem is to provide the learning algorithm with access to an external knowledge source which compensates for the small number of examples. The work presented here describes an unsupervised algorithm which makes use of the WordNet ontology (Fellbaum 1998) as a knowledge source for the IE domain knowledge acquisition problem.

IE itself can be thought of as, at least, two sub-tasks: named entity recognition and relation extraction. The first of these is the process of identifying every item of a specific semantic type in the text. For example in the sixth MUC the semantic types included PERSON, ORGANIZATION and LOCATION. The second stage, relation extraction, involves the identification of appropriate relations between these entities and their combination into templates. The majority of IE systems carry out the stages of NE recognition and relation extraction as separate processes.

A lot of research has recently been carried out on the application of machine learning to named entity recognition, systems which achieve accurate results have been reported and implementations of named entity identifiers are available freely on the internet (e.g. <http://www.gate.ac.uk>). Unsupervised approaches to the NE recognition problem were presented by (Riloff & Jones 1999; Collins & Singer 1999). However, attempts to automate the relation extraction task have been less successful. Approaches include (Soderland 1999; Chieu, Ng, & Lee 2003). However, each relied on supervised learning techniques and, consequently, depend upon the existence of annotated training data. To our knowledge the only approach which has made use of unsupervised learning techniques for relation extraction was presented by (Yangarber *et al.* 2000). This paper presents an alternative unsupervised algorithm for identifying the relations which can be used to generate domain knowledge for an IE task. This approach is compared with the one presented by (Yangarber *et al.* 2000) and found to complement it.

The remainder of this paper is organised as follows. We begin by describing the system for IE domain knowledge acquisition based on an unsupervised algorithm. Included in this description are details of the way in which the corpus is pre-processed and background information on lexical

Copyright © 2004, American Association for Artificial Intelligence ([www.aaai.org](http://www.aaai.org)). All rights reserved.

<sup>1</sup>The term “unsupervised” is used here to mean an approach which generalises from a small number of labelled seed examples. This usage is common in the field of Natural Language Processing (see (Manning & Schütze 1999)) although the term “semi-supervised” is more usual in other areas.

similarity measures. An existing approach to the problem (Yangarber *et al.* 2000) is then described along with details of how this can be combined with the method presented. Two evaluation regimes are then described; one based on the identification of relevant documents and another which aims to identify sentences in a corpus which are relevant for a particular IE task. Results on each of these evaluation regimes are then presented. It is found that the unsupervised approach introduced here is preferable to the one presented by (Yangarber *et al.* 2000) and that a combination of both approaches achieves the best results.

## System Details

### Document Processing

A number of processing stages have to be applied to the documents before the learning process can take place. Firstly, named entities are marked. The corpus is then parsed to identify Subject-Verb-Object (SVO) patterns in each sentence. Parsing was carried out using a version of MINIPAR (Lin 1999) which was adapted to process the named entities marked in the text. The dependency trees produced by MINIPAR are then analysed to extract the SVO-pattern. Each pattern consists of either two or three elements. Sentences containing intransitive verbs yield patterns containing two elements, the second of which is the verb and the first its logical subject. For example, the sentence “The player scored on his debut” would yield the pattern *player+score*. The first two elements of patterns from sentences containing transitive verbs are the same while the third position represents the verb’s object. Active and passive voice is taken into account in MINIPAR’s output so the sentences “The professor taught the class” and “The class was taught by the professor” would yield the same triple, *professor+teach+class*. The indirect object of ditransitive verbs is not extracted; these verbs are treated like transitive verbs for the purposes of this analysis.

### Semantic Similarity

The aim of our learning mechanism is to learn patterns which are similar to those known to be relevant. To do this we make use of work which has been carried out on measuring semantic similarity between words using the WordNet ontology. We experimented with several semantic similarity measures from the literature and found that the measure proposed by (Lin 1998) was the most suitable for our application. Lin’s approach relies on assigning numerical values to each node in the WordNet hierarchy representing the amount of information they contain (a technique developed by (Resnik 1995)). This value was known as *Information Content (IC)* and was derived from corpus probabilities, so,  $IC(s) = -\log(\Pr(s))$ . For two senses,  $s_1$  and  $s_2$ , the *lowest common subsumer*,  $lcs(s_1, s_2)$ , is defined as the senses with the highest information content which subsumes both senses in the WordNet hierarchy. Lin used these elements to calculate the semantic similarity of two senses according to this formula:  $sim(s_1, s_2) = \frac{2 \times IC(lcs(s_1, s_2))}{IC(s_1) + IC(s_2)}$

It is straightforward to extend the notion of similarity between a pair of word senses to similarity between two words,

$w_1$  and  $w_2$ , by choosing the senses which maximise the similarity score across all possible pairs.

We now extend the notion of word similarity to one of similarity between SVO patterns. The similarity of a pair of patterns can be computed from the similarity between the words in each corresponding pattern slot. Thus, if  $p_1$  and  $p_2$  are patterns consisting of  $m$  and  $n$  elements respectively (where  $1 \leq n, m \leq 3$ ) and that the  $m$ th element of pattern  $p_1$  is denoted by  $p_{1m}$ . Then the similarity can be computed from equation 1 in which  $MIN(m, n)$  and  $MAX(m, n)$  are the smaller and larger of the values  $m$  and  $n$ . Normalising the sum of the word similarity scores by the longer of the two patterns takes into account patterns of differing length by penalising the pattern similarity score.

$$psim(p_1, p_2) = \frac{\sum_{i=1}^{MIN(m,n)} word\_sim(p_{1i}, p_{2i})}{MAX(m, n)} \quad (1)$$

As mentioned above, the document pre-processing includes marking named entities in text and these then appear in the SVO patterns. For example, the sentence “Jones left London” would yield the pattern *NAMPERSON+leave+NAMLOCATION*. The *NAMPERSON* and *NAMLOCATION* identifiers we used to denote the name classes do not appear in the WordNet ontology and so it is not possible to directly compare their similarity with other words. To avoid this problem these tokens are manually mapped onto the most appropriate node in the WordNet hierarchy. For example, *NAMPERSON* is mapped onto the first nominal sense of the word “person” in the WordNet ontology. This mapping process is not particularly time-consuming since the number of named entity types with which a corpus is annotated is usually quite small. For example, in the experiments described later in this paper just seven named entity types were used to annotate the corpus.

### A Semantic-Similarity-based Learning Algorithm

This idea of pattern similarity can be used to create an unsupervised approach to pattern generation. By taking a set of patterns which represent a particular extraction task we can compute the similarity of other patterns. Those which are found to be similar can be added to the set of accepted patterns and the process repeated. Our system starts with an initial set of seed patterns which are indicative of the extraction task. The rest of the patterns in the document set are then compared against the seeds to identify the most similar. Some of the similar patterns are then accepted and added to the seed set and the process repeated with the enlarged set of accepted patterns. The decision to accept a pattern can be either completely automatic or can be passed to a domain expert to include human judgement. Several schemes for deciding which of the scored patterns to accept were implemented and evaluated although a description would be too long for this paper. For the experiments described in this paper we used a scheme where the four highest scoring patterns whose score is within 0.95 of the best pattern are accepted.

We shall now explain the process of deciding which patterns are similar to a given set of currently accepted patterns in more detail. Firstly, our algorithm disregards any patterns which occur just once in the corpus since these may be due to noise. The remainder of the patterns are assigned a similarity score based on equation 2.

$$score(p) = \frac{\sum_{c \in C} psim(c, p) \times conf(c)}{\log(|C|) + 1} \quad (2)$$

The score of a candidate pattern is restricted to the subset of accepted patterns which are “comparable” to it, denoted by  $C$  in this equation. This is useful since a good candidate pattern may be very similar to some of the accepted patterns but not others. For the purposes of this algorithm two patterns are said to be close if they have the same filler in at least one slot, for example `john+phone+mary` and `simon+phone` would qualify as close.

Equation 2 includes the term  $conf(c)$ , a value in the range 0 to 1 representing the system’s confidence in pattern  $c$ . Such a confidence score is necessary since it is inevitable that some patterns accepted during the learning process will be less reliable than the seed patterns. These patterns may in turn contribute to the acceptance of other less suitable patterns and, if this process continues, the learning process may be misled into accepting many unsuitable patterns. The approach used here to avoid this problem is to introduce a score for pattern confidence which is taken into account during the scoring of candidate patterns.

We can be reasonably sure that seed patterns will be suitable for a domain and therefore these are given a confidence score of 1. After each iteration the newly accepted patterns are assigned confidence score based on the confidence of patterns already accepted. More formally, the confidence of the patterns accepted during iteration  $i + 1$  is based on the confidence of the patterns which contributed towards its acceptance (that is those which are in the set  $C$  in equation 2) and their own confidence scores in the previous iteration. The formula for calculating the score is shown in equation 3.

$$conf^{i+1}(p) = \frac{\sum_{c \in C} conf^i(c)}{|C|} \cdot \left( \underset{c \in C}{MAX} \sqrt{psim(p, c)} \right) \quad (3)$$

Equation 3 guarantees that the confidence of the newly accepted pattern will be no greater than the highest confidence score of the patterns which contributed to its acceptance. However, the confidence score of patterns which have already been accepted can also be improved if they contribute to a new pattern whose score is higher than their own. So, if  $conf^{i+1}(p) > conf^i(c)$  for some  $c \in C$  in equation 3 then  $conf^{i+1}(p)$  is increased to  $conf^i(c)$ .

## Alternative Approaches

### Distributional Similarity

The approach just described was inspired by another unsupervised algorithm for learning relations (Yangarber *et al.* 2000). Their approach can be thought of as *document*

*centric* and is motivated by the assumption that a document containing a large number of patterns which have already been identified as relevant is likely to contain further patterns which are relevant to the domain. We implemented an algorithm similar to the one described by (Yangarber *et al.* 2000) which evaluated candidate patterns according to their distribution in a corpus. For the remainder of the paper this is referred to as the “distributional” approach. The pattern evaluation method is identical to the one described by (Yangarber *et al.* 2000) but it is important to mention that the algorithm used here is not identical. The system described by (Yangarber *et al.* 2000) makes some generalisations across pattern elements by grouping certain elements together. However, there is no difference between the expressiveness of the patterns learned by either approach.

### Combining Approaches

The semantic- and distributional-based classifiers use different approaches to identify suitable patterns. The first chooses patterns which are semantically similar to those which have already been accepted while the second selects those which tend to occur in the same documents.

Previous work has shown that techniques such as cotraining (Blum & Mitchell 1998) which combine learning algorithms can improve results. In the context of the two algorithms presented here, cotraining would operate by combining the set of patterns returned by the two approaches after each iteration to create a larger set of accepted patterns. A cotraining approach was implemented but found to perform poorly. The reason for this seemed to be that the two learning algorithms were sometimes misled into accepting patterns which were not relevant. When the set of proposed patterns were combined each learning algorithm was misled by the larger set of accepted irrelevant patterns.

For this particular application it seemed results could be improved if accepting irrelevant patterns could be avoided as much as possible. One way to do this is to accept the patterns which are identified by both classifiers, essentially a collaborative voting approach. Cotraining can be thought of as accepting the set union of a set of classifiers as their combined output. The voting approach we adopt is equivalent to choosing the intersection of each approaches’ accepted patterns. If the intersection is empty the best pattern identified by each learning algorithm during the previous iteration is chosen.

## Evaluation

(Yangarber *et al.* 2000) noted that quantitative evaluation of pattern induction systems is difficult to achieve. The discovery process does not easily fit into MUC-style evaluations since the learned patterns do not directly fit into an IE system. However, in addition to learning a set of patterns, the system also notes the relevance of documents relative to a particular set of seed patterns. (Yangarber *et al.* 2000) quantitatively evaluated the documents relevance scores. This evaluation is similar to the “text-filtering” sub-task used in MUC-6 in which systems were evaluated according to their ability to discriminate between relevant and non-relevant

documents for the extraction task. A similar evaluation was implemented for this study which allows comparison between the results reported by (Yangarber *et al.* 2000) and those reported here. After each iteration the induced patterns can be assigned a score based on the documents they match and these scores can be used to update the document relevance scores based on the best set of patterns which match them. This process can be used to assign relevance scores to documents regardless of which learning algorithm is being used.

Identifying the document containing relevant information can be considered as a preliminary stage of an IE task. A further step is to identify the sentences within those documents which are relevant. This “sentence filtering” task is a more fine-grained evaluation and is likely to provide more information about how well a given set of patterns is likely to perform as part of an IE system. (Soderland 1999) developed a version of the MUC6 corpus in which events are marked at the sentence level. If a sentence contains an event description this is marked in the text. The set of patterns learned by the algorithm after each iteration can be compared against this corpus to determine how accurately they identify the relevant sentences for this extraction task.

NAMCOMPANY+appoint+NAMPERSON
NAMCOMPANY+elect+NAMPERSON
NAMCOMPANY+promote+NAMPERSON
NAMCOMPANY+name+NAMPERSON
NAMPERSON+resign
NAMPERSON+depart
NAMPERSON+quit
NAMPERSON+step-down

Table 1: Seed patterns for the management succession domain extraction task

The evaluation compared the three approaches described above. The similarity-based algorithm is referred to as `smx_sim`, the document centric approach presented by (Yangarber *et al.* 2000) as `dist` and their combination as `comb`. Each of these approaches were compared against a simple baseline, called `random`, which chose four random patterns at each iteration. Each of the approaches used the set of seed patterns listed in Table 1 which are indicative of the management succession extraction task.

### Evaluation Corpus

The corpus used for the experiments was compiled from two sources: the training and testing corpus used in the sixth Message Understanding conference (MUC-6) (MUC 1995) and a subset of the Reuters Corpus (Rose, Stevenson, & Whitehead 2002). The MUC-6 task was to extract information about the movements of executives from newswire texts. A document is relevant if it has a filled template associated with it. 590 documents from a version of the MUC-6 evaluation corpus described by (Soderland 1999) were used.

The documents which make up the Reuters corpus are also newswire texts. However, unlike the MUC-6 corpus they have not been marked for relevance to the MUC-6

extraction task. Each document in the Reuters corpus is marked with a set of codes which indicate the general topic of the story. One of the topic codes (C411) refers to management succession events and this can be used to identify relevant documents. A corpus of 6,000 documents was extracted from the Reuters corpus. One half of this corpus contained the first (chronologically) 3000 documents marked with the C411 topic code and the remainder contained the first 3000 documents which were not marked with that code.

Each document in this corpus was preprocessed to extract the patterns they contain following the process outlined earlier. Relevant named entities are already marked in the MUC corpus and, since these have been manually verified, were used for the preprocessing. These simply had to be transformed into a format suitable for the adapted version of MINIPAR. Named entities are not marked in the Reuters corpus and so the 6,000 documents were run through the named entity identifier in GATE (Cunningham *et al.* 2002) before parsing.

## Results

### Document Filtering

Results for both the document and sentence filtering experiments are reported, starting with document filtering. Table 2 shows the precision, recall and F-measure scores for each of the three approaches and the random baseline. Continuous F-measure scores are also presented in graphical format in Figure 1.

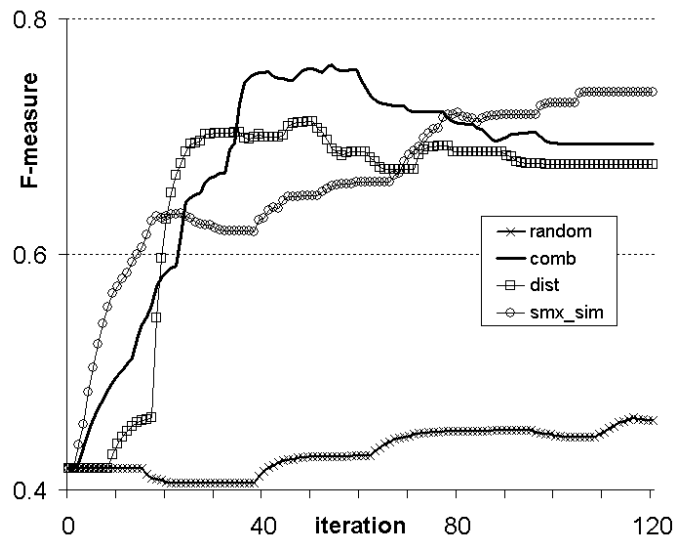


Figure 1: F-measure scores for alternative approaches applied to the document filtering task over 120 iterations

It can be seen that each of the three methods outperforms the random baseline. The baseline method records a slight improvement in F-measure score during the learning process. This is because the set of seed patterns matches few documents in the corpus resulting in a low recall score. The

#	random			smx_sim			dist			comb		
	P	R	F	P	R	F	P	R	F	P	R	F
0	1.00	0.26	0.42	1.00	0.26	0.42	1.00	0.26	0.42	1.00	0.26	0.42
20	0.89	0.26	0.41	0.73	0.53	0.61	0.73	0.55	0.63	0.83	0.45	0.58
40	0.88	0.27	0.42	0.62	0.72	0.67	0.67	0.74	0.70	0.76	0.75	0.75
60	0.88	0.28	0.43	0.60	0.74	0.66	0.59	0.83	0.69	0.69	0.81	0.75
80	0.89	0.30	0.45	0.63	0.83	0.71	0.57	0.87	0.69	0.58	0.93	0.71
100	0.85	0.30	0.45	0.63	0.91	0.74	0.55	0.87	0.68	0.55	0.94	0.69
120	0.82	0.32	0.46	0.62	0.91	0.74	0.55	0.87	0.68	0.55	0.94	0.69

Table 2: Comparison of different approaches applied to the document filtering task over 120 iterations

corpus consists of an equal amount of relevant and irrelevant documents so there are many patterns which improve the recall without too much detriment to precision and this leads to an overall increase in the F-measure.

The two learning algorithms, *smx\_sim* and *dist*, behave quite differently. The improvement of the *smx\_sim* algorithm is slower than the distributional algorithm although the performance after 120 iterations is higher. The approach which records the highest score is the combination of these approaches.

### Sentence Filtering

For the sentence filtering experiments only the results of the three implemented approaches are described. As with the document filtering experiments, the random baseline performed badly. The results from the sentence filtering experiment are shown in Figure 2. It can be seen that there is a noticeable benefit from the use of a combination of algorithms and that the improvement is more pronounced for this task compared to document filtering. The semantic similarity algorithm also seems to perform better than the distributional approach. This is perhaps not surprising since the distributional approach will learn patterns which tend to occur in documents with relevant ones although these patterns themselves may not identify sentences containing relevant information. The semantic similarity algorithm chooses sentences with a similar meaning to those already accepted. However, there appears to be a clear benefit to be gained from using a combination of these algorithms.

Despite the improvement gained from using a combination of learning algorithms the overall results are not what might be hoped for. The best F-measure occurs after 49 iterations of the combined approach with both precision and recall scores of 55. However, it should be borne in mind that this approach uses only a simple representation scheme for sentences which means that the majority of patterns learned identify both relevant and irrelevant sentences. For example, the set of seed patterns (see Table 1) returns an F-measure of 18.1 with a precision of 81.2 and recall of 10.2. The precision is not 1 since the pattern *NAMPERSON+resign* happened to match sentences describing historical events which were not marked as relevant in this corpus.

These results should be placed in the context of an unsupervised learning algorithm which makes use of a straightforward sentence representation scheme. While it is unlikely that this approach could be used to directly produce an IE

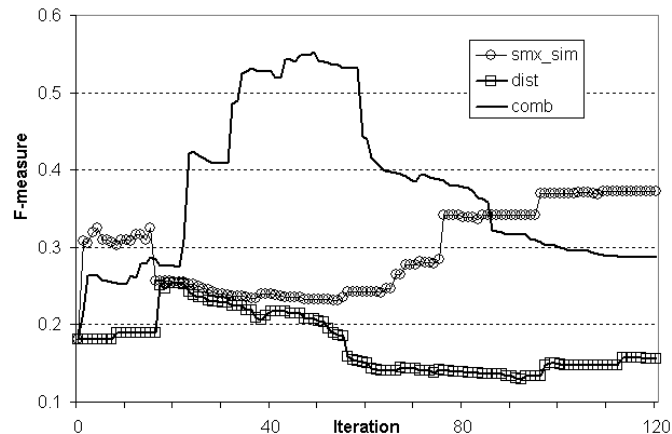


Figure 2: F-measure scores for alternative approaches applied to the sentence filtering task over 120 iterations

system there is a clear benefit to be gained from an approach which shows the improvements in sentence filtering accuracy presented in Figure 2 with no manual intervention.

### Conclusion

The approach presented here is inspired by the approach of (Yangarber *et al.* 2000) but makes use of a different assumption regarding which patterns are likely to be relevant to a particular extraction task. Evaluation showed that the proposed approach performs well when compared against the existing method. In addition, the approaches are complementary and the best results are obtained when the results of the learning algorithms are combined.

This work represents a step towards truly domain-independent IE systems. Employing an unsupervised learning algorithm removes almost all of the requirement for a human annotator to provide example patterns to the system. However, unsupervised algorithms are often hampered by a lack of information so linking it to a resource such as WordNet has benefits. WordNet is also a generic resource which is not associated with a particular domain and this means the learning algorithm make use of that knowledge to acquire knowledge for a diverse range of IE tasks.

iteration	dist			smx_sim			comb		
	P	R	F	P	R	F	P	R	F
0	0.81	0.10	0.18	0.81	0.10	0.18	0.81	0.10	0.18
20	0.32	0.21	0.25	0.26	0.25	0.26	0.52	0.19	0.28
40	0.20	0.23	0.22	0.22	0.26	0.24	0.57	0.49	0.53
60	0.11	0.23	0.15	0.22	0.27	0.24	0.33	0.66	0.44
80	0.10	0.25	0.14	0.28	0.42	0.34	0.26	0.68	0.38
100	0.10	0.30	0.15	0.30	0.48	0.37	0.19	0.68	0.30
120	0.10	0.32	0.16	0.30	0.49	0.37	0.18	0.68	0.29

Table 3: Comparison of different approaches to sentence filtering over 120 iterations

## Acknowledgements

I am grateful to Roman Yangarber for advice on the implementation of the distributional similarity algorithm and to Neil Ireson, Mirella Lapatta and Angus Roberts from Sheffield University for advice on early drafts of this paper.

## References

- Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98)*, 92–100.
- Chieu, H.; Ng, H.; and Lee, Y. 2003. Closing the Gap: Learning-based Information Extraction Rivaling Knowledge-engineering Methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, 216–223.
- Collins, M., and Singer, Y. 1999. Unsupervised models for Named Entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 100–110.
- Cunningham, H.; Maynard, D.; Bontcheva, K.; and Tablan, V. 2002. GATE: an Architecture for Development of Robust HLT. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 168–175.
- Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database and some of its Applications*. Cambridge, MA: MIT Press.
- Lehnert, W.; Cardie, C.; Fisher, D.; McCarthy, J.; Riloff, E.; and Soderland, S. 1992. University of Massachusetts: Description of the CIRCUS System used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, 282–288.
- Lin, D. 1998. An Information-Theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine learning (ICML-98)*, 296–304.
- Lin, D. 1999. MINIPAR: a minimalist parser. In *Maryland Linguistics Colloquium*.
- Manning, H., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
- Miller, S.; Crystal, M.; Fox, H.; Ramshaw, L.; Schwartz, R.; Stone, R.; and Weischedel, R. 1998. Algorithms that learn to extract information—BBN: Description of the SIFT system as used for MUC. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.
- MUC. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, San Mateo, CA: Morgan Kaufmann.
- Resnik, P. 1995. Using Information Content to evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, 448–453.
- Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*.
- Rose, T.; Stevenson, M.; and Whitehead, M. 2002. The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-02)*, 827–833.
- Soderland, S. 1999. Learning Information Extraction Rules for Semi-structured and free text. *Machine Learning* 31(1-3):233–272.
- Yangarber, R.; Grishman, R.; Tapanainen, P.; and Hutunnen, S. 2000. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of the Applied Natural Language Processing Conference (ANLP 2000)*, 282–289.