Learning Information Extraction Patterns Using WordNet

Mark Stevenson and **Mark A. Greenwood**
Department of Computer Science
University of Sheffield
Sheffield, S1 4DP, UK
marks@dcs.shef.ac.uk
m.greenwood@dcs.shef.ac.uk

Abstract

Information Extraction (IE) systems often use patterns to identify relevant information in text but these are difficult and time-consuming to generate manually. This paper presents a new approach to the automatic learning of IE patterns which uses WordNet to judge the similarity between patterns. The algorithm starts with a small set of sample extraction patterns and uses a similarity metric, based on a version of the vector space model augmented with information from WordNet, to learn similar patterns. This approach is found to perform better than a previously reported method which relied on information about the distribution of patterns in a corpus and did not make use of WordNet.

1 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. One approach to this problem is to use machine learning to automatically learn the domain-specific information required to port a system. Soderland [1999] developed an approach that learned rules from text which had been annotated with the information to be extracted. However, the annotated text required for training is often difficult and time-consuming to obtain. An alternative approach is to use weakly supervised learning algorithms, these do not require large amounts of annotated training data and rely on a small set of examples instead. These approaches greatly reduced the burden on the application developer by alleviating the knowledge acquisition bottleneck.

Weakly supervised algorithms have the benefit of requiring only small amounts of annotated training data. But the learning task is more challenging since there are fewer examples of the patterns to be learned. Providing the learning algorithm with access to additional knowledge can compensate for the limited number of annotated examples. The approach we have chosen is to augment an IE pattern learning algorithm with information from WordNet which allows our system to decide when patterns have similar meanings.

The remainder of this paper is organised as follows. We begin by describing the general process of weakly supervised pattern induction and an existing approach, based on the distribution of patterns in a corpus (Section 2). Section 3 introduces a new algorithm that uses WordNet to generalise extraction patterns and Section 4 an implementation of this approach. Section 5 describes an evaluation regime based on the MUC-6 management succession task [MUC, 1995]. The results of an experiment in which several methods for calculating the similarity between extraction patterns are compared is presented in Section 6. Section 7 compares the proposed approach with an existing method.

2 Weakly Supervised Extraction Pattern Learning

We begin by outlining the general process of learning extraction patterns, similar to the approach presented by Yangarber [2003].

1. For a given IE scenario we assume the existence of a set of documents against which the system can be trained. The documents are either relevant (contain the description of an event relevant to the scenario) or irrelevant. However, the documents are not annotated

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 95–102.

and the algorithm does not have access to this information.

2. This corpus is pre-processed to generate a set of all patterns which could be used to represent sentences contained in the corpus, call this set P. The aim of the learning process is to identify the subset of P representing patterns which are relevant to the IE scenario.
3. The user provides a small set of seed patterns, P_{seed}, which are relevant to the scenario. These patterns are used to form the set of currently accepted patterns, P_{acc}, so $P_{acc} \leftarrow P_{seed}$. The remaining patterns are treated as candidates for inclusion in the accepted set, forming the set $P_{cand} (= P - P_{acc})$.
4. A function, f, is used to assign a score to each pattern in P_{cand} based on those which are currently in P_{acc}. This function assigns a real number to candidate patterns so $\forall\ c \in P_{cand},\ f(c, P_{acc}) \mapsto \mathbb{R}$. A set of high scoring patterns (based on absolute scores or ranks after the set of patterns has been ordered by scores) are chosen as being suitable for inclusion in the set of accepted patterns. These form the set P_{learn}.
5. The patterns in P_{learn} are added to P_{acc} and removed from P_{cand}, so $P_{acc} \leftarrow P_{acc} \cup P_{learn}$ and $P_{cand} \leftarrow P_{acc} - P_{learn}$
6. If a suitable set of patterns has been learned then stop, otherwise go to step 4

An important choice in the development of such an algorithm is step 4, the process of ranking the candidate patterns, this effectively determines which of the candidate patterns will be learned. Yangarber *et. al.* [2000] chose an approach motivated by the assumption that documents containing a large number of patterns which have already been identified as relevant to a particular IE scenario are likely to contain more relevant patterns. Patterns which occur in these documents far more than others will then receive high scores. This approach can be viewed as being document-centric.

This approach has been shown to successfully acquire useful extraction patterns which, when added to an IE system, improved its performance [Yangarber et al., 2000]. However, it relies on an assumption about the way in which relevant patterns are distributed in a document collection and may learn patterns which tend to occur in the same documents as relevant ones whether or not they are actually relevant. For example, we could imagine an IE scenario in which relevant documents contain a piece of information which is related to, but distinct from, the information we aim to extract. If patterns expressing this information were more likely to occur in relevant documents than irrelevant ones the document-centric approach would also learn these irrelevant patterns.

Rather than focusing on the documents matched by a pattern, an alternative approach is to rank patterns according to how similar their meanings are to those which are known to be relevant. This approach is motivated by the fact that the same event can be described in different ways in natural language. Once a pattern has been identified as being relevant it is highly likely that its paraphrases and patterns with similar meanings will also be relevant to the same extraction task. This approach also avoids the problem which may be present in the document-centric approach since patterns which happen to co-occur in the same documents as relevant ones but have different meanings will not be ranked highly.

The approach presented here uses WordNet [Fellbaum, 1998] to determine pattern similarity. Other systems which use WordNet to help with the learning of IE patterns include [Chai and Biermann, 1999; Català et al., 2003]. Although they used WordNet's hierarchical structure to generalise patterns rather than identify those with similar meanings.

3 Semantic IE Pattern Learning

For these experiments extraction patterns consist of predicate-argument structures, as proposed by Yangarber [2003]. Under this scheme patterns consist of triples representing the subject, verb, and object (SVO) of a clause. The first element is the "semantic" subject (or agent), for example "John" is a clausal subject in each of the sentences "John hit Bill", "Bill was hit by John", "Mary saw John hit Bill", and "John is a bully". The second element is the verb in the clause and the third the object (patient) or predicate. "Bill" is a clausal object in the first three example sentences and "bully" in the final sentence. When a verb is being used intransitively, the pattern for that clause is restricted to only the first pair of elements.

The filler of each pattern element can be either a lexical item or semantic category such as person name, country, currency values, numerical expressions etc. In this paper lexical items are represented in lower case and semantic categories are capitalised. For example, in the pattern `COMPANY+fired+ceo`, `fired` and `ceo` are lexical items and `COMPANY` a semantic category which could match any lexical item belonging to that type.

A vector space model, similar to the ones used in Information Retrieval [Salton and McGill, 1983], is used to represent patterns and a similarity metric defined to identify those with similar meanings. Each pattern can be represented as a set of pattern element-filler pairs. For example, the pattern `COMPANY+fired+ceo` consists of three pairs: `subject_COMPANY`, `verb_fire` and `object_ceo`. Each pair consists of either a lexical item or semantic category and pattern element. The set of all possible element-filler pairs for a group of patterns can be used to generate the basis of a vector space in which each pattern can be represented. In this space patterns are represented as binary vectors in which an element with value 1 denotes that the pattern contains a particular pair and 0 that it does not.

3.1 Pattern Similarity

The similarity of two pattern vectors can be compared using the measure shown in Equation 1. Here $\vec{a}$ and $\vec{b}$ are pattern vectors, $\vec{b^T}$ the transpose of $\vec{b}$ and W a matrix that lists the similarity between each of the possible pattern element-filler pairs.

$$similarity(\vec{a}, \vec{b}) = \frac{\vec{a} W \vec{b^T}}{|\vec{a}||\vec{b}|} \qquad (1)$$

The semantic similarity matrix, W, contains non-negative real numbers and is crucial for this measure. Assume that the set of patterns, P, consists of n element-filler pairs denoted by $p_1, p_2, \ldots p_n$. Each row and column of W represents one of these pairs and they are consistently labelled. So, for any i such that $1 \leq i \leq n$, row i and column i are both labelled with pair p_i. If w_{ij} is the element of W in row i and column j then the value of w_{ij} represents the similarity between the pairs p_i and p_j. Note that we assume the similarity of two element-filler pairs is symmetric, so $w_{ij} = w_{ji}$ (W is symmetric). Pairs with different pattern elements (i.e. grammatical roles) are automatically given a similarity score of 0. Diagonal elements of W represent the self-similarity between pairs and have the greatest values. The actual values denoting the similarity between pattern elements are acquired using existing lexical similarity metrics (see Section 4).

Figure 1 gives an example using three patterns which shows how they could be represented as vectors given the set of element filler pairs forming a basis for the vector space. A similarity matrix with example values is also shown.

Table 1: Similarity values for example patterns using Equation 1 and cosine metric

patterns	similarity	cosine
a, b	0.275	0.25
a, c	0.31	0
b, c	0.177	0

Table 1 shows the similarity values for each pair of example patterns obtained using equation 1 under the column labelled "similarity". The patterns `president+resign` and `executive+leave+job` are identified as having the most similar meanings despite the fact that they have no element filler pairs in common. This table also shows the values obtained for each vector pair using the cosine metric which is a standard measure for determining the similarity of documents when using the vector space model for Information Retrieval. It can be seen that this metric chooses patterns `president+resign` and `president+comment` as the most similar caused by the fact that they share one element filler pair. The cosine metric does not take the similarity between elements of a vector into account and would not perform well for our application.[1]

WordNet is an appropriate resource for this application since it is constructed to reflects paradigmatic semantics, providing information about words which may be substituted for one another. It is therefore ideal for this application since it

[1] The cosine metric for a pair of vectors is given by the calculation $\frac{a.b}{|a||b|}$. Substituting the matrix multiplication in the numerator of Equation 1 for the dot product of vectors $\vec{a}$ and $\vec{b}$ would give the cosine metric. Note that taking the dot product of a pair of vectors is equivalent to multiplying by the identity matrix, i.e. $\vec{a}.\vec{b} = \vec{a} I \vec{b^T}$. Under our interpretation of the similarity matrix, W, this equates to saying that all pattern element-filler pairs are identical to each other and not similar to anything else.

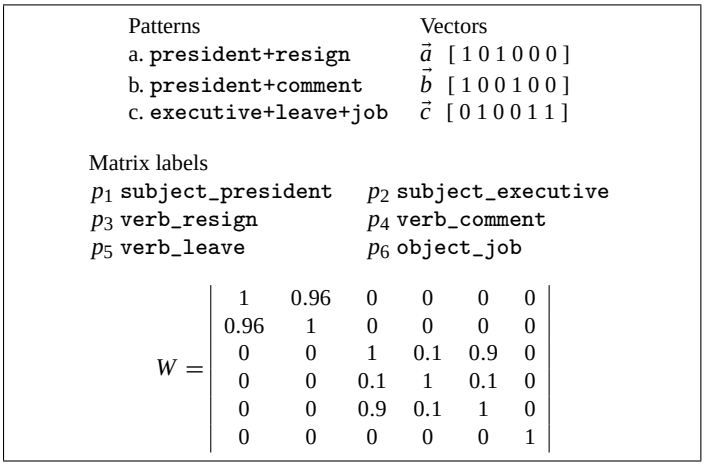

Figure 1: Similarity scores and matrix for an example vector space formed from three patterns

provides a way of determining when patterns have similar meanings.

3.2 Learning Algorithm

This pattern similarity measure can be used to create a weakly supervised approach to pattern acquisition following the general outline provided in Section 2. Each candidate pattern is compared against the set of currently accepted patterns using the measure described in Section 3.1. We experimented with several techniques for ranking candidate patterns based on these scores, including using the best and average score, and found that the best results were obtained when each candidate pattern was ranked according to its score when compared against the centroid vector of the set of currently accepted patterns. We also experimented with several schemes for deciding which of the scored patterns to accept and chose one where the four highest scoring patterns are accepted provided their score is within 0.95 of the highest scoring pattern.

Our algorithm disregards any patterns whose corpus occurrences are below a set threshold, α, since these may be due to noise. In addition, a second threshold, β, is used to discard very frequent patterns which are often too general to be useful for IE. Patterns which occur in more than $\beta \times C$, where C is the number of documents in the collection, are discarded. For the experiments in this paper we set α to 2 and β to 0.3.

4 Implementation

A number of pre-processing stages have to be applied to documents in order for the set of patterns to be extracted before learning can take place. Firstly, items belonging to semantic categories are identified using a named entity identifier. The corpus is then parsed using a version of MINIPAR [Lin, 1999] adapted to process text marked with named entities. The dependency trees produced by MINIPAR are then analysed to extract the SVO-patterns. Active and passive voice is taken into account in MINIPAR's output so the sentences "COMPANY fired their C.E.O." and "The C.E.O. was fired by COMPANY" would yield the same triple, `COMPANY+fire+ceo`. The indirect object of ditransitive verbs is not extracted; these verbs are treated as transitive verbs for the purposes of this analysis.

We experimented with a number of different methods for populating the semantic similarity matrix described in Section 3.1. Several methods for calculating the similarity of a pair of words using information from the WordNet lexicon [Fellbaum, 1998] have been described in the literature. These methods can be grouped into two main approaches according to the information they use (1) path length and (2) node informativeness.

The first of these are based on the intuitive notion that concepts closer to each other in the WordNet hierarchy are more similar than those which are distant. Leacock and Chodrow [1998] use the formula $sim_{LCh} = -\log(length/2 \times D)$ to define the similarity between two synsets, s_1 and s_2, where $length$ is the length of the

shortest path between s_1 and s_2 in WordNet and D is the maximum depth. An alternative approach was proposed by Wu and Palmer [1994] who defined similarity in term of the relative depth (i.e. distance from root node) of synsets. Their measure uses the *lowest common subsumer* of a pair of nodes, this is the unique lowest node in the WordNet hierarchy which is a parent of both nodes. Similarity is defined as $sim_{WuP} = \frac{2 \times depth(lcs(s_1,s_2))}{depth(s_1)+depth(s_2)}$ where $depth(s)$ is nodes s's depth in the WordNet hierarchy and $lcs(s_1, s_2)$ denotes the lowest common subsumer of nodes s_1 and s_2.

The second group of measures use corpus frequency counts to represent the informativeness of each node in WordNet, a technique developed by Resnik [1995]. Nodes near the root of the hierarchy are not considered to be informative and have low values while those nearer the leaves have higher values, for example the concept *fish* would be more informative than *animal*. Numerical values representing the informativeness of each node are calculated from frequency counts of the words in that synset. The information content (IC) for a synset, s, is calculated as $IC(s) = -\log(\Pr(s))$ where $\Pr(s)$ is the probability of that synset occurring in the corpus (estimated using word frequency counts). Resnik's similarity measure is provided by $sim_{Res} = IC(lcs(s_1, s_2))$, i.e. the similarity of a pair of nodes is defined to be the informativeness of their lowest common subsumer. Two other measures use variations of this approach, combining different elements in a formula. Jiang and Conrath [1997] define the distance between a pair of synsets as $dist_{JCn} = IC(s_1) + IC(s_2) - 2 \times IC(lcs(s_1, s_2))$ which can be converted to a similarity value by taking the reciprocal i.e. $sim_{JCn} = \frac{1}{dist_{JCn}}$. Lin [1998] combined the same terms in a different formula: $sim_{Lin} = \frac{2 \times IC(lcs(s_1,s_2))}{IC(s_1)+IC(s_2)}$.

Each of these similarity measures has been defined for a pair of WordNet synsets. When a word has several possible synsets (senses), each similarity metric chooses the synset of each word which maximises the similarity value. The WordNet::Similarity package [Pedersen et al., 2004] implements these measures and was used for the experiments described here.

These similarity metrics provide values only for the lexical items contained in WordNet. In order to cope with the semantic categories in our patterns we provided a mapping between these and a nominal synset in WordNet. This mapping is shown in Table 2 where, for example, person#1 denotes the first nominal sense of *person*.

5 Evaluation

Various approaches have been suggested for the evaluation of automatic IE pattern acquisition. Riloff [1996] judged the precision of patterns learned by reviewing them manually. Yangarber *et. al.* [2000] evaluated the patterns learned by their system by manually integrating them into an existing IE system. They also indirectly evaluated the learned patterns using a text filtering task. Patterns were judged according to their ability to distinguish between relevant and irrelevant documents for the extraction task (similar to a MUC-6 sub-task [MUC, 1995]). This approach has the advantage of being easily automated given document relevance information.

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 MUC-6 corpus in which events are marked at the sentence level. 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.

The evaluation corpus used for the experiments was compiled from the training and testing corpus used in MUC-6 where the task was to extract information about the movements of executives from newswire texts. 590 documents from a version of the MUC-6 evaluation corpus described by Soderland [1999] were used. After the pre-processing stages described in Section 4, the MUC-6 corpus produced 15,407 pattern tokens from 11,294 different types. 10,512 patterns appeared just once and these were effectively discarded since our learning algorithm only considers patterns which occur at least twice (see Section 3.2).

The following seed patterns, denoting relevant sentences for the management succession extraction task, were used for these experiments:
`PERSON+resign`,

Table 2: Mapping between semantic categories and WordNet nodes

PERSON ↦ person#1	LOCATION ↦ geographic_area#1
MONEY ↦ money#1	COMPANY ↦ organisation#3
DATE ↦ day#2	ORGANISATION ↦ organisation#3
POST ↦ post#3	

PERSON+depart,
PERSON+quit,
COMPANY+appoint+PERSON,
COMPANY+name+PERSON,
COMPANY+elect+PERSON,
COMPANY+promote+PERSON.

6 Results

A comparison of the sentence filtering results obtained by the learning algorithm using different semantic similarity measures to populate the matrix are shown at 20 iteration intervals in Table 3 and continuously in Figure 2.

Table 3: Sentence filtering results over 120 iterations obtained using different WordNet similarity metrics

	Similarity Metric				
#	LCh	WuP	Res	JCn	Lin
0	0.181	0.181	0.181	0.181	0.181
20	0.353	0.427	0.254	0.543	0.42
40	0.514	0.532	0.282	0.545	0.555
60	0.478	0.499	0.325	0.537	0.476
80	0.402	0.421	0.29	0.522	0.49
100	0.397	0.399	0.291	0.424	0.406
120	0.353	0.397	0.315	0.412	0.357

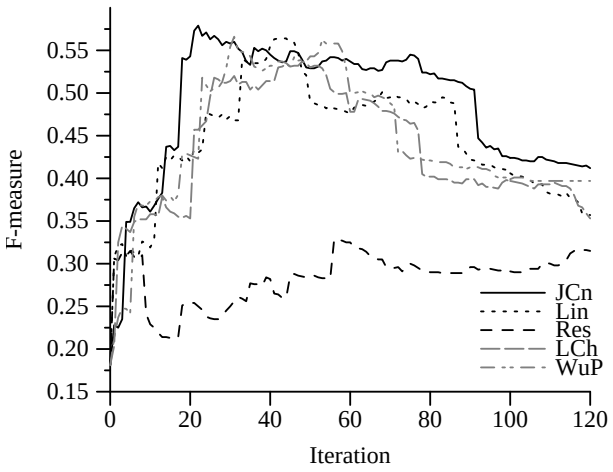

Figure 2: F-measure scores on sentence identification task for several similarity metrics

It can be seen that the highest F-measure for the majority of iterations is recorded for the sim_{JCn} metric. The difference between these scores and sim_{LCh}, sim_{WuP} and sim_{Lin} is often not very large but these measures consistently perform far better than sim_{Res}. The different between these results using the five alternative metrics is significant ($p < 0.001$, Friedman Test). This test also suggests that the best similarity measure is sim_{JCn}, followed by sim_{Lin}, sim_{WuP}, sim_{LCh} and, finally, sim_{Res}.

These results indicate that there is little difference between the two main approaches to measuring semantic similarity, at least for this task. The best- and worst-performing measures (sim_{JCn} and sim_{Res}) both use the node informativeness approach. However, the two measures which combine data about the informativeness of nodes and their lowest common subsumer also perform well.

7 Comparison with Alternative Approach

The approached described here (Section 3) was compared against the document-centric approach (Section 2). The sim_{JCn} measure was used to populate the matrix for the semantic similarity method. The document-centric approach was represented by a re-implementation of the Yangarber *et. al.* [2000] algorithm (described in Section 2). However, this implementation is not identical to the original. In addition to using different parsers, our implementation does not generalise pattern elements by grouping together particular elements. There is no difference between the expressiveness of the patterns learned in either implementation and we do not believe the differences have any effect on the outcome of these experiments. The document-centric approach relies upon a large corpus containing a mixture of documents which are both relevant and irrelevant to the extraction task. Consequently we provided this algorithm with additional training data in the form of 6,000 documents from the Reuters Corpus Volume I [Rose et al., 2002]. Half of these documents were identified as being relevant to the management succession task, using corpus metadata, and the remainder irrelevant. Adding this corpus to the data used by the

document-centric approach improved the maximal F-measure by 70% but did not provide any benefit to the semantic-similarity method and was not used by that algorithm.

The two approaches, semantic similarity and document-centric, were applied to the sentence filtering task and the results over 120 iterations listed in Table 4 where the columns P, R and F list the precision, recall and F-measure values observed at 20 iteration intervals.[2] Figure 3 shows F-measure values for 120 iterations. The semantic similarity algorithm can be seen to significantly outperform the document-centric approach ($p < 0.001$, Wilcoxon Signed Ranks Test).

The precision scores for the sentence filtering task in Table 4 show that the semantic similarity algorithm consistently learns more accurate patterns than the document-centric approach. It also learns patterns with high recall much faster than the document-centric approach, by the 120th iteration the pattern set covers almost 95% of relevant sentences while the document-centric approach covers only 69%.

As previously mentioned, Yangarber *et. al.* [2000] evaluate their system using a document filtering task. In addition to learning a set of patterns, their system also notes the relevance of documents based on the current set of accepted patterns. Stevenson and Greenwood [2005] describe an evaluation in which the document-centric approach is compared against the proposed method on a similar document filtering evaluation to the one used by Yangarber *et. al.* [2000]. It was found that the semantic similarity approach also outperforms the existing approach on this task. These results suggest that the semantic similarity approach, by learning patterns with similar meanings to the seeds, can identify sentences which are relevant to the extraction task while the document-centric approach generates patterns from relevant documents, although these do not necessarily match relevant sentences.

Table 4: Comparison of the differing approaches applied to sentence filtering task over 120 iterations

	Document-centric			Semantic similarity		
#	P	R	F	P	R	F
0	0.813	0.102	0.181	0.813	0.102	0.181
20	0.301	0.219	0.253	0.606	0.492	0.543
40	0.190	0.477	0.272	0.474	0.641	0.545
60	0.195	0.570	0.290	0.423	0.734	0.537
80	0.181	0.609	0.280	0.369	0.891	0.522
100	0.176	0.648	0.277	0.276	0.922	0.424
120	0.171	0.688	0.274	0.264	0.945	0.412

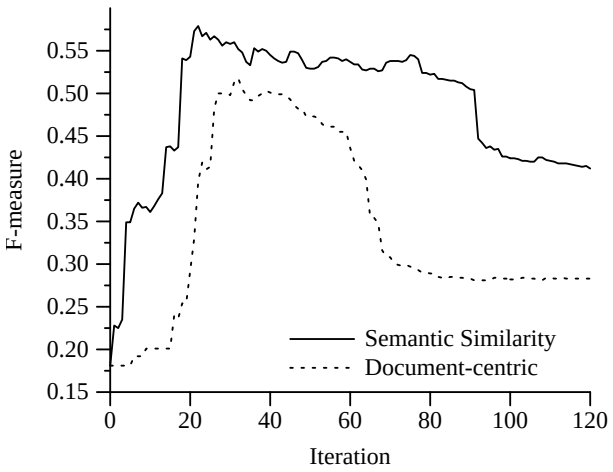

Figure 3: F-measure scores for the sentence filtering task

8 Conclusion

The approach to weakly supervised IE pattern acquisition presented here is related to other techniques but uses different assumptions regarding which patterns are likely to be relevant to a particular extraction task. Evaluation has showed that the approach presented here outperforms the previously reported document-centric method. The semantic similarity approach has the additional advantage of not requiring a large corpus containing a mixture of documents relevant and irrelevant to the extraction task. This corpus is unannotated, and so may not be difficult to obtain, but is nevertheless an additional requirement.

The learning algorithm presented in Section 3 includes a mechanism for comparing two extraction patterns using lexical similarity information from WordNet. This technique could be applied to other language processing tasks including question answering and paraphrase identification and generation. Wong *et. al.* [1985] proposed an extension of the standard vector space model for Information Retrieval which used a matrix to represent similarity values between search terms. They chose to populate this matrix using corpus occurrence statistics of search terms but the technique

[2] The set of seed patterns returns a precision of 0.81 for this task. The precision is not 1 since the pattern `PERSON+resign` matches sentences describing historical events ("Jones resigned last year.") which were not marked as relevant in this corpus following MUC guidelines.

presented here could be integrated into that approach as an alternative method.

Acknowledgements

This work was carried out as part of the RESuLT project funded by the EPSRC (GR/T06391). Roman Yangarber provided advice on the reimplementation of the document-centric approach to pattern learning.

References

N. Català, N. Castell, and M. Martin. 2003. A portable method for acquiring information extraction patterns without annotated corpora. *Natural Language Engineering*, 9(2):151–179.

J. Chai and A. Biermann. 1999. The Use of Word Sense Disambiguation in an Information Extraction System. In *Proceedings of the Eleventh Annual Conference on Innovative Applications of Artificial Intelligence*, pages 850–855, Portland, OR.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and some of its Applications*. MIT Press, Cambridge, MA.

J. Jiang and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan.

C. Leacock and M. Chodrow. 1998. Combining local context and WordNet similarity for word sense identification. In *[Fellbaum, 1998]*, pages 265–283.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine learning (ICML-98)*, Madison, Wisconsin.

Dekan Lin. 1999. MINIPAR: A Minimalist Parser. In *Maryland Linguistics Colloquium*, University of Maryland, College Park.

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

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, San Jose, CA.

P. Resnik. 1995. Using Information Content to evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 448–453, Montreal, Canada.

E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049, Portland, OR.

T. Rose, M. Stevenson, and M. Whitehead. 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)*, pages 827–832, La Palmas de Gran Canaria.

G. Salton and M. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

Stephen Soderland. 1999. Learning Information Extraction Rules for Semi-structured and free text. *Machine Learning*, 31(1-3):233–272.

Mark Stevenson and Mark A. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 379–386, Ann Arbour, MI.

S. Wong, W. Ziarko, and P. Wong. 1985. Generalized Vector Space Model In Information Retrieval. In *Proceedings of the 8th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Conference of the Association for Computational Liinguistics*, pages 133–138.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 940–946, Saarbrücken, Germany.

Roman Yangarber. 2003. Counter-training in the Discovery of Semantic Patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 343–350, Sapporo, Japan.