A Semantic Approach to IE Pattern Induction

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

Abstract

This paper presents a novel algorithm for the acquisition of Information Extraction patterns. The approach makes the assumption that useful patterns will have similar meanings to those already identified as relevant. Patterns are compared using a variation of the standard vector space model in which information from an ontology is used to capture semantic similarity. Evaluation shows this algorithm performs well when compared with a previously reported document-centric approach.

1 Introduction

Developing systems which can be easily adapted to new domains with the minimum of human intervention is a major challenge in Information Extraction (IE). Early IE systems were based on knowledge engineering approaches but suffered from a knowledge acquisition bottleneck. For example, Lehnert et al. (1992) reported that their system required around 1,500 person-hours of expert labour to modify for a new extraction task. One approach to this problem is to use machine learning to automatically learn the domain-specific information required to port a system (Riloff, 1996). Yangarber et al. (2000) proposed an algorithm for learning extraction patterns for a small number of examples which greatly reduced the burden on the application developer and reduced the knowledge acquisition bottleneck.

Weakly supervised algorithms, which bootstrap from a small number of examples, have the advantage of requiring only small amounts of annotated data, which is often difficult and time-consuming to produce. However, this also means that there are fewer examples of the patterns to be learned, making the learning task more challenging. Providing the learning algorithm with access to additional knowledge can compensate for the limited number of annotated examples. This paper presents a novel weakly supervised algorithm for IE pattern induction which makes use of the WordNet ontology (Fellbaum, 1998).

Extraction patterns are potentially useful for many language processing tasks, including question answering and the identification of lexical relations (such as meronomy and hyponymy). In addition, IE patterns encode the different ways in which a piece of information can be expressed in text. For example, "Acme Inc. fired Jones", "Acme Inc. let Jones go", and "Jones was given notice by his employers, Acme Inc." are all ways of expressing the same fact. Consequently the generation of extraction patterns is pertinent to paraphrase identification which is central to many language processing problems.

We begin by describing the general process of pattern induction and an existing approach, based on the distribution of patterns in a corpus (Section 2). We then introduce a new algorithm which makes use of WordNet to generalise extraction patterns (Section 3) and describe an implementation (Section 4). Two evaluation regimes are described; one based on the identification of relevant documents and another which aims to identify sentences in a corpus which

Proceedings of the 43rd Annual Meeting of the ACL, pages 379–386,
Ann Arbor, June 2005.

are relevant for a particular IE task (Section 5). Results on each of these evaluation regimes are then presented (Sections 6 and 7).

2 Extraction Pattern Learning

We begin by outlining the general process of learning extraction patterns, similar to one 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 unannotated and may be either relevant (contain the description of an event relevant to the scenario) or irrelevant although the algorithm has no access to this information.
2. This corpus is pre-processed to generate the set of all patterns which could be used to represent sentences contained in the corpus, call this set S. The aim of the learning process is to identify the subset of S representing patterns which are relevant to the IE scenario.
3. The user provides a small set of seed patterns, S_{seed}, which are relevant to the scenario. These patterns are used to form the set of currently accepted patterns, S_{acc}, so $S_{acc} \leftarrow S_{seed}$. The remaining patterns are treated as candidates for inclusion in the accepted set, these form the set $S_{cand} (= S - S_{acc})$.
4. A function, f, is used to assign a score to each pattern in S_{cand} based on those which are currently in S_{acc}. This function assigns a real number to candidate patterns so $\forall\ c\ \epsilon\ S_{cand},\ f(c, S_{acc}) \mapsto \Re$. 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 S_{learn}.
5. The patterns in S_{learn} are added to S_{acc} and removed from S_{cand}, so $S_{acc} \leftarrow S_{acc} \cup S_{learn}$ and $S_{cand} \leftarrow S_{acc} - S_{learn}$
6. If a suitable set of patterns has been learned then stop, otherwise go to step 4

2.1 Document-centric approach

A key choice in the development of such an algorithm is step 4, the process of ranking the candidate patterns, which 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 already identified as relevant to a particular IE scenario are likely to contain further relevant patterns. This approach, which can be viewed as being document-centric, operates by associating confidence scores with patterns and relevance scores with documents. Initially seed patterns are given a maximum confidence score of 1 and all others a 0 score. Each document is given a relevance score based on the patterns which occur within it. Candidate patterns are ranked according to the proportion of relevant and irrelevant documents in which they occur, those found in relevant documents far more than in irrelevant ones are ranked highly. After new patterns have been accepted all patterns' confidence scores are updated, based on the documents in which they occur, and documents' relevance according to the accepted patterns they contain.

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 the 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 semantic-similarity approach 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. We now go on to describe a new algorithm which implements this approach.

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 these 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 one. 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.

The algorithm described here relies on identifying patterns with similar meanings. The approach we have developed to do this is inspired by the vector space model which is commonly used in Information Retrieval (Salton and McGill, 1983) and language processing in general (Pado and Lapata, 2003). 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_fired` and `object_ceo`. Each pair consists of either a lexical item or semantic category, and pattern element. Once an appropriate set of pairs has been established a pattern can be represented as a binary vector 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.

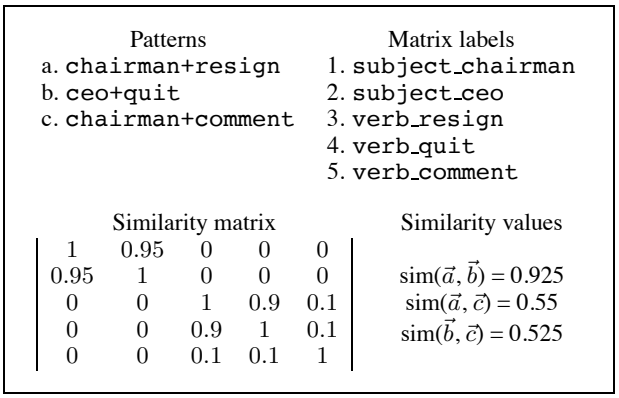

Figure 1: Similarity scores and matrix for an example vector space formed from three patterns

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

The semantic similarity matrix W contains information about the similarity of each pattern element-filler pair stored as 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 $p_1, p_2, ...p_n$. 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 $w_{ij} = w_{ji}$ and, consequently, W is a symmetric matrix. 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.

Figure 1 shows an example using three patterns, `chairman+resign`, `ceo+quit` and `chairman+comment`. This shows how these patterns are represented as vectors and gives a sample semantic similarity matrix. It can be seen that the first pair of patterns are the most similar using the proposed measure.

The measure in Equation 1 is similar to the cosine metric, commonly used to determine the similarity of documents in the vector space model approach

to Information Retrieval. However, the cosine metric will not perform well for our application since it does not take into account the similarity between elements of a vector and would assign equal similarity to each pair of patterns in the example shown in Figure 1.[1] The semantic similarity matrix in Equation 1 provides a mechanism to capture semantic similarity between lexical items which allows us to identify `chairman+resign` and `ceo+quit` as the most similar pair of patterns.

3.2 Populating the Matrix

It is important to choose appropriate values for the elements of W. We chose to make use of the research that has concentrated on computing similarity between pairs of lexical items using the WordNet hierarchy (Resnik, 1995; Jiang and Conrath, 1997; Patwardhan et al., 2003). We experimented with several of the measures which have been reported in the literature and found that the one proposed by Jiang and Conrath (1997) to be the most effective.

The similarity measure proposed by Jiang and Conrath (1997) relies on a technique developed by Resnik (1995) which assigns numerical values to each sense in the WordNet hierarchy based upon the amount of information it represents. These values are derived from corpus counts of the words in the synset, either directly or via the hyponym relation and are used to derive the *Information Content* (IC) of a synset c thus $IC(c) = -\log(\Pr(c))$. For two senses, s_1 and s_2, the *lowest common subsumer*, $lcs(s_1, s_2)$, is defined as the sense with the highest information content (most specific) which subsumes both senses in the WordNet hierarchy. Jiang and Conrath used these elements to calculate the semantic distance between a pair or words, w_1 and w_2, according to this formula (where $senses(w)$ is the set of all possible WordNet senses for word w):

$$\underset{\substack{s_1 \,\epsilon\, senses(w_1),\\ s_2 \,\epsilon\, senses(w_2)}}{ARGMAX} \; IC(s_1)+IC(s_2)-2\times IC(lcs(s_1,s_2)) \tag{2}$$

Patwardhan et al. (2003) convert this distance metric into a similarity measure by taking its multiplicative inverse. Their implementation was used in the experiments described later.

As mentioned above, the second part of a pattern element-filler pair can be either a lexical item or a semantic category, such as company. The identifiers used to denote these categories, i.e. `COMPANY`, do not appear in WordNet and so it is not possible to directly compare their similarity with other lexical items. To avoid this problem these tokens are manually mapped onto the most appropriate node in the WordNet hierarchy which is then used for similarity calculations. 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 in this paper just seven semantic classes were sufficient to annotate the corpus.

3.3 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 (a full description would be too long for this paper) resulting in a scheme where the four highest scoring patterns whose score is within 0.95 of the best pattern are accepted.

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

[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 each pattern element-filler pair being identical to itself but not similar to anything else.

threshold, β, is used to determine the maximum number of documents in which a pattern can occur since these very frequent patterns 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 not learned. 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 by running the text through the named entity identifier in the GATE system (Cunningham et al., 2002). The corpus is then parsed, using a version of MINIPAR (Lin, 1999) adapted to process text marked with named entities, to produce dependency trees from which SVO-patterns are extracted. 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 like transitive verbs for the purposes of this analysis.

An implementation of the algorithm described in Section 3 was completed in addition to an implementation of the document-centric algorithm described in Section 2.1. It is important to mention that this implementation is not identical to the one described by Yangarber et al. (2000). Their system 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 and we do not believe this difference has any effect on the results of our experiments.

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) developed an indirect method which allowed automatic evaluation. In addition to learning a set of patterns, their system also notes the relevance of documents based on the current set of accepted patterns. Assuming the subset of documents relevant to a particular IE scenario is known, it is possible to use these relevance judgements to determine how accurately a given set of patterns can discriminate the relevant documents from the irrelevant. This evaluation is similar to the "text-filtering" sub-task used in the sixth Message Understanding Conference (MUC-6) (1995) in which systems were evaluated according to their ability to identify the documents relevant to the extraction task. The document filtering evaluation technique was used to allow comparison with previous studies.

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 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.

5.1 Evaluation Corpus

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. 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.

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 3.3).

The document-centric approach benefits from a large corpus containing a mixture of relevant and irrelevant documents. We provided this using a subset of the Reuters Corpus Volume I (Rose et al., 2002) which, like the MUC-6 corpus, consists of newswire

COMPANY+appoint+PERSON
COMPANY+elect+PERSON
COMPANY+promote+PERSON
COMPANY+name+PERSON
PERSON+resign
PERSON+depart
PERSON+quit

Table 1: Seed patterns for extraction task

texts. 3000 documents relevant to the management succession task (identified using document meta-data) and 3000 irrelevant documents were used to produce the supplementary corpus. This supplementary corpus yielded 126,942 pattern tokens and 79,473 types with 14,576 of these appearing more than once. Adding the supplementary corpus to the data set used by the document-centric approach led to an improvement of around 15% on the document filtering task and over 70% for sentence filtering. It was not used for the semantic similarity algorithm since there was no benefit.

The set of seed patterns listed in Table 1 are indicative of the management succession extraction task and were used for these experiments.

6 Results

6.1 Document Filtering

Results for both the document and sentence filtering experiments are reported in Table 2 which lists precision, recall and F-measure for each approach on both evaluations. Results from the document filtering experiment are shown on the left hand side of the table and continuous F-measure scores for the same experiment are also presented in graphical format in Figure 2. While the document-centric approach achieves the highest F-measure of either system (0.83 on the 33rd iteration compared against 0.81 after 48 iterations of the semantic similarity approach) it only outperforms the proposed approach for a few iterations. In addition the semantic similarity approach learns more quickly and does not exhibit as much of a drop in performance after it has reached its best value. Overall the semantic similarity approach was found to be significantly better than the document-centric approach ($p < 0.001$, Wilcoxon Signed Ranks Test).

Although it is an informative evaluation, the document filtering task is limited for evaluating IE pattern learning. This evaluation indicates whether the set of patterns being learned can identify documents containing descriptions of events but does not provide any information about whether it can find those events within the documents. In addition, the set of seed patterns used for these experiments have a high precision and low recall (Table 2). We have found that the distribution of patterns and documents in the corpus means that learning virtually any pattern will help improve the F-measure. Consequently, we believe the sentence filtering evaluation to be more useful for this problem.

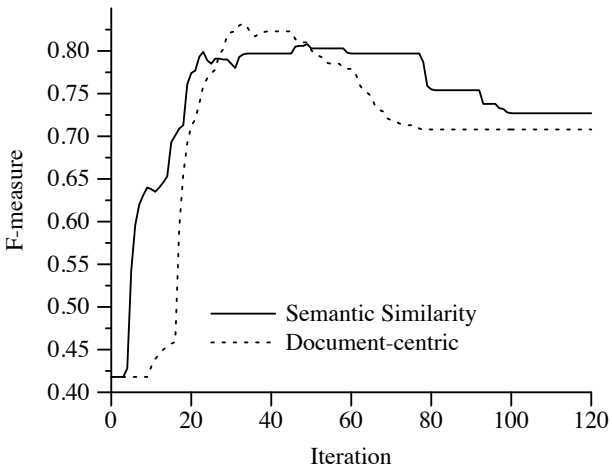

Figure 2: Evaluating document filtering.

6.2 Sentence Filtering

Results from the sentence filtering experiment are shown in tabular format in the right hand side of Table 2[2] and graphically in Figure 3. The semantic similarity algorithm can be seen to outperform the document-centric approach. This difference is also significant ($p < 0.001$, Wilcoxon Signed Ranks Text).

The clear difference between these results shows that the semantic similarity approach can indeed identify relevant sentences while the document-centric method identifies patterns which match relevant documents, although not necessarily relevant sentences.

[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.

Number of Iterations	Document Filtering						Sentence Filtering					
	Document-centric			Semantic similarity			Document-centric			Semantic similarity		
	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	0.81	0.10	0.18	0.81	0.10	0.18
20	0.75	0.68	0.71	0.77	0.78	0.77	0.30	0.29	0.29	0.61	0.49	0.54
40	0.72	0.96	0.82	0.70	0.93	0.80	0.40	0.67	0.51	0.47	0.64	0.55
60	0.65	0.96	0.78	0.68	0.96	0.80	0.32	0.70	0.44	0.42	0.73	0.54
80	0.56	0.96	0.71	0.61	0.98	0.76	0.18	0.71	0.29	0.37	0.89	0.52
100	0.56	0.96	0.71	0.58	0.98	0.73	0.18	0.73	0.28	0.28	0.92	0.42
120	0.56	0.96	0.71	0.58	0.98	0.73	0.17	0.75	0.28	0.26	0.95	0.41

Table 2: Comparison of the different approaches over 120 iterations

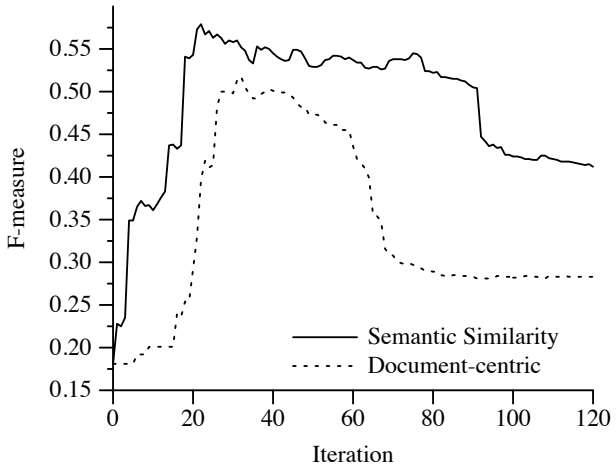

Figure 3: Evaluating sentence filtering.

The precision scores for the sentence filtering task in Table 2 show that the semantic similarity algorithm consistently learns more accurate patterns than the existing approach. At the same time it 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 75%.

7 Discussion

The approach to 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 semantic generalisation approach presented here performs well when compared to a previously reported document-centric method. Differences between the two approaches are most obvious when the results of the sentence filtering task are considered and it seems that this is a more informative evaluation for this problem. 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 best score recorded by the proposed algorithm on the sentence filtering task is an F-measure of 0.58 (22nd iteration). While this result is lower than those reported for IE systems based on knowledge engineering approaches these results should be placed in the context of a weakly supervised learning algorithm which could be used to complement manual approaches. These results could be improved by manual filtering the patterns identified by the algorithm.

The learning algorithm presented in Section 3 includes a mechanism for comparing two extraction patterns using information about lexical similarity derived from WordNet. This approach is not restricted to this application and could be applied to other language processing tasks such as question answering, paraphrase identification and generation or as a variant of the vector space model commonly used in Information Retrieval. In addition, Sudo et al. (2003) proposed representations for IE patterns which extends the SVO representation used here and, while they did not appear to significantly improve IE, it is expected that it will be straightforward to extend the vector space model to those pat-

tern representations.

One of the reasons for the success of the approach described here is the appropriateness of WordNet which is constructed on paradigmatic principles, listing the words which may be substituted for one another, and is consequently an excellent resource for this application. WordNet is also a generic resource not associated with a particular domain which means the learning algorithm can make use of that knowledge to acquire patterns for a diverse range of IE tasks. This work represents a step towards truly domain-independent IE systems. Employing a weakly supervised learning algorithm removes much of the requirement for a human annotator to provide example patterns. Such approaches are often hampered by a lack of information but the additional knowledge in WordNet helps to compensate.

Acknowledgements

This work was carried out as part of the RESuLT project funded by the EPSRC (GR/T06391). Roman Yangarber provided advice on the re-implementation of the document-centric algorithm. We are also grateful for the detailed comments provided by the anonymous reviewers of this paper.

References

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: an Architecture for Development of Robust HLT. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, pages 168–175, Philadelphia, PA.

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.

W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. 1992. University of Massachusetts: Description of the CIRCUS System used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, San Francisco, CA.

D. 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.

S. Pado and M. Lapata. 2003. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 128–135, Sapporo, Japan.

S. Patwardhan, S. Banerjee, and T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conferences on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City.

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 *LREC-02*, pages 827–832, La Palmas, Spain.

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

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

K. Sudo, S. Sekine, and R. Grishman. 2003. An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 224–231.

R. Yangarber, R. Grishman, P. Tapanainen, and S. 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.

R. 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.