ACQUISITION OF SELECTIONAL PATTERNS

RALPH GRISHMAN and JOHN STERLING
Computer Science Department
New York University
New York, NY 10003, U.S.A.

1 The Problem

For most natural language analysis systems, one of the major hurdles in porting the system to a new domain is the development of an appropriate set of semantic patterns. Such patterns are typically needed to guide syntactic analysis (as selectional constraints) and to control the translation into a predicate-argument representation. As systems are ported to more complex domains, the set of patterns grows and the task of accumulating them manually becomes more formidable.

There has therefore been increasing interest in acquiring such patterns automatically from a sample of text in the domain, through an analysis of word co-occurrence patterns either in raw text (word sequences) or in parsed text. We briefly review some of this work later in the article. We have been specifically concerned about the practicality of using such techniques in place of manual encoding to develop the selectional patterns for new domains. In the experiments reported here, we have therefore been particularly concerned with the evaluation of our automatically generated patterns, in terms of their completeness and accuracy and in terms of their efficacy in performing selection during parsing.

2 Patterns and Word Classes

In principle, the semantic patterns could be stated in terms of individual words - this verb can meaningfully occur with this subject, etc. In practice, however, this would produce an unmanageable number of patterns for even a small domain. We therefore need to define semantic word classes for the domain and state our patterns in terms of these classes.

Ideally, then, a discovery procedure for semantic patterns would acquire both the word classes and the patterns from an analysis of the word co-occurrence patterns. In order to simplify the task, however, while we are exploring different strategies, we have divided it into separate tasks, that of acquiring word classes and that of acquiring semantic patterns (given a set of word classes). We have previously described [1] some experiments in which the principal word classes for a sublanguge were obtained through the clustering of words based on the contexts in which they occurred, and we expect to renew such experiments using the larger corpora now available. However, the experiments we report below are limited to the acquisition of semantic patterns given a set of manually prepared word classes.

3 Pattern Acquisition

The basic mechanism of pattern acquisition is straightforward. A sample of text in a new domain is parsed using a broad-coverage grammar (but without any semantic constraints). The resulting parse trees are then transformed into a regularized syntactic structure (similar to the f-structure of Lexical-Functional Grammar). This regularization in particular reduces all different clausal forms (active, passive, questions, extraposed forms, relative clauses, reduced relatives, etc.) into a uniform structure with the 'logical' subject and object explicitly marked. For example, the sentence

Fred ate fresh cheese from France.

would produce the regularized syntactic structure

```
(s eat (subject (np Fred))
       (object (np cheese (a-pos fresh)
                          (from (np France)))))
```

We then extract from this regularized structure a series of triples of the form

head	syntactic-function	value

where – if the value is another NP or S – only the head is recorded. For example, for the above sentence we would get the triples

eat	subject	Fred
eat	object	cheese
cheese	a-pos	fresh
cheese	from	France

Finally, we generalize these triples by replacing words by word classes. We had previously prepared, by a purely manual analysis of the corpus, a hierarchy of word classes and a set of semantic patterns for the corpus we were using. From this hierarchy we identified the classes which were most frequently referred to in the manually prepared patterns. The generalization process replaces a word by the most specific class to which it belongs (since we have a hierarchy with nested classes, a word will typically belong to several classes). As we explain in our experiment section below, we made some runs generalizing just the value and others generalizing both the head and the value.

As we process the corpus, we keep a count of the frequency of each head-function-value triple. In addition, we keep separate counts of the number of times each word appears as a head, and the number of times each head-function pair appears (independent of value).

4 Coping with Multiple Parses

The procedure described above is sufficient if we are able to obtain the correct parse for each sentence. However, if we are porting to a new domain and have no semantic constraints, we must rely entirely upon syntactic constraints and so will be confronted with a large number of incorrect parses for each sentence, along with (hopefully) the correct one. We have experimented with several approaches to dealing with this problem:

1. If a sentence has N parses, we can generate triples from all the parses and then include each triple with a weight of 1/N.

2. We can generate a stochastic grammar through unsupervised training on a portion of the corpus [2]. We can then parse the corpus with this stochastic grammar and take only the most probable parse for each sentence. For sentences which still generated $N > 1$ equally-probable parses, we would use a 1/N weight as before.

3. In place of a 1/N weighting, we can refine the weights for alternative parse trees using an iterative procedure analogous to the inside-outside algorithm [3]. We begin by generating all parses, as in approach 1. Then, based on the counts obtained initially (using 1/N weighting), we can compute the probability for the various triples and from these the probabilities of the alternative parse trees. We can then repeat the process, recomputing the counts with weightings based on these probabilities.

All of these approaches rely on the expectation that correct patterns arising from correct parses will occur repeatedly, while the distribution of incorrect patterns from incorrect parses will be more scattered, and so—over a sufficiently large corpus—we can distinguish correct from incorrect patterns on the basis of frequency.

5 Evaluation Methods

To gather patterns, we analyzed a series of articles on terrorism which were obtained from the Foreign Broadcast Information Service and used as the development corpus for the Third Message Understanding Conference (held in San Diego, CA, May 1991) [4]. For pattern collection, we used 1000 such articles with a total of 14,196 sentences and 330,769 words. Not all sentences parsed, both because of limitations in our grammar and because we impose a limit on the search which the parser can perform for each sentence. Within these limits, we were able to parse a total of 7,455 sentences.[1]

The most clearly definable function of the triples we collect is to act as a selectional constraint: to differentiate between meaningful and meaningless triples in new text, and thus identify the correct analysis.

We used two methods to evaluate the effectiveness of the triples we generated. The first

[1] For these runs we disabled several heuristics in our system which increase the number of sentences which can be parsed at some cost in the average quality of parses; hence the relatively low percentage of sentences which obtained parses.

method involved a comparison with manually-classified triples. We took 10 articles (not in the training corpus), generated all parses, and produced the triples from each parse. These triples were stated in terms of words, and were not generalized to word classes. We classified each triple as semantically valid or invalid (a triple was counted as valid if we believed that this pair of words could meaningfully occur in this relationship, even if this was not the intended relationship in this particular text). This produced a test set containing a total of 1169 distinct triples, of which 716 were valid and 453 were invalid.

We then established a threshold T for the weighted triples counts in our training set, and defined

v_+ number of triples in test set which were classified as valid and which appeared in training set with count $> T$

v_- number of triples in test set which were classified as valid and which appeared in training set with count $\leq T$

i_+ number of triples in test set which were classified as invalid and which appeared in training set with count $> T$

i_- number of triples in test set which were classified as invalid and which appeared in training set with count $\leq T$

and then defined

$$\begin{aligned} \text{recall} &= \frac{v_+}{v_+ + v_-} \\ \text{precision} &= \frac{v_+}{v_+ + i_+} \\ \text{error rate} &= \frac{i_+}{i_+ + i_-} \end{aligned}$$

By varying the threshold, we can plot graphs of recall vs. precision or recall vs. error-rate. These plots can then be compared among different strategies for collecting triples and for generalizing triples. The precision figures are somewhat misleading because of the relatively small number of invalid triples in the test set: since only 39% of the triples are invalid, a filter which accepted all the triples in the test set would still be accounted as having 61% precision. We have therefore used the error rate in the figures below (plotting recall against 1–error-rate).

The second evaluation method involves the use of the triples in selection and a comparison of the parses produced against a set of known correct parses. In this case the known correct parses were prepared manually by the University of Pennsylvania as part of their "Tree Bank" project. For this evaluation, we used a set of 317 sentences, again distinct from the training set. In comparing the parser output against the standard trees, we measured the degree to which the tree structures coincide, stated as recall, precision, and number of crossings. These measures have been defined in earlier papers [5,6,7].

6 Results

Our first set of experiments were conducted to compare three methods of coping with multiple parses. These methods, as described in section 4, are (1) generating all N parses of a sentence, and weighting each by $1/N$; (2) selecting the N most likely parses as determined by a stochastic grammar, and weighting those each by $1/N$; (3) generating all parses, but assigning weights to alternative parses using a form of the inside-outside procedure. These experiments were conducted using a smaller training set, a set of 727 sentences drawn from 90 articles. We generated a set of triples using each of the three methods and then evaluated them against our hand-classified triples, as described in section 5. We show in Figure 1 the threshold vs. recall curves for the three methods; in Figure 2 the recall vs. 1–error rate curves.

These experiments showed only very small differences between the three methods (the inside-outside method showed slightly better accuracy at some levels of recall). Based on this, we decided to use method 2 (statistical grammar) for subsequent experiments. Other things being equal, method 2 has the virtue of generating far fewer parses (an average of 1.5 per sentence, vs. 37 per sentence when all parses are produced), and hence a far smaller file of regularized parses (about 10 MB for our entire training corpus of 1000 articles, vs. somewhat over 200 MB which would have been required if all parses were generated). Using method 2, therefore, we generated the triples for our 1000-article training corpus.

Our second series of experiments compared three different ways of accumulating data from the triples:

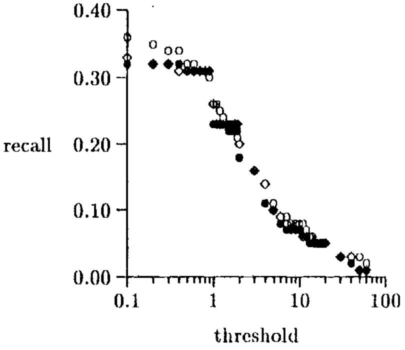

Figure 1: Comparison of methods for dealing with multiple parses in pattern collection, using training corpus of 90 articles. Threshold vs. recall for ○ = all parses; ◇ = all parses + inside-outside; • = most probable parses from stochastic grammar.

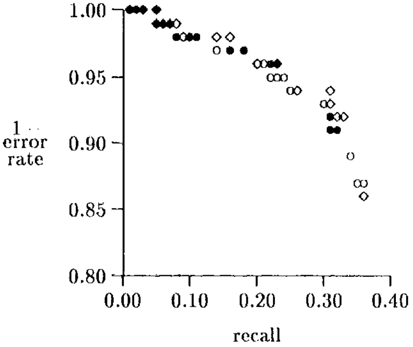

Figure 2: Comparison of methods for dealing with multiple parses in pattern collection, using training corpus of 90 articles. Recall vs. 1-error rate for ○ = all parses; ◇ = all parses + inside-outside; • = most probable parses from stochastic grammar.

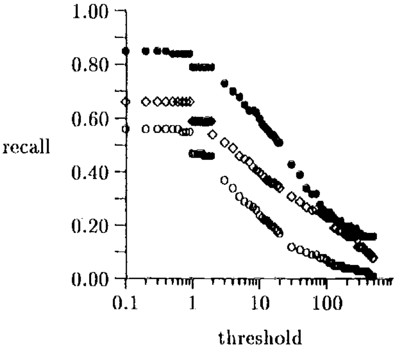

Figure 3: Comparison of pattern generalization techniques, using training corpus of 1000 articles. Threshold vs. recall for ○ = triples without generalized heads; ◇ = triples with generalized heads; • = pairs.

1. generalizing the value in a head-function-value triple to a word class, but not generalizing the head

2. generalizing both the value and the head

3. ignoring the value field entirely in a head-function-value triple, and accumulating counts of head-function pairs (with no generalization applied to the head); a match with the hand-marked triples is therefore recorded if the head and function fields match

Again, we evaluated the patterns produced by each method against the hand-marked triples. Figure 3 shows the threshold vs. recall curves for each method; Figure 4 the recall vs. 1-error rate curves. Figure 3 indicates that using pairs yields the highest recall for a given threshold, triples with generalized heads an intermediate value, and triples without generalized heads the lowest recall. The error rate vs. recall curves of figure 4 do not show a great difference between methods, but they do indicate that, over the range of recalls for which they overlap, using triples without generalized heads produces the lowest error rate.

Finally, we conducted a series of experiments to compare the effectiveness of the triples in selecting the correct parse. In effect, the selection procedure works as follows. For each sentence in the test corpus, the system generates all possible

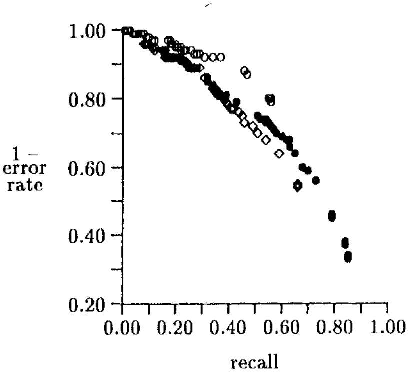

Figure 4: Comparison of pattern generalization techniques, using training corpus of 1000 articles. Recall vs. 1–error rate for ○ = triples without generalized heads; ◇ = triples with generalized heads; • = pairs.

parses and then generates a set of triples from each parse. Each triple is assigned a score; the score for the parse is the product of the scores of the triples obtained from the parse (the use of products is consistent with the idea that the score for a triple to some degree reflects the probability that this triple is semantically valid). The parse or parses with the highest total score are then selected for evaluation.

We tested three approaches to assigning a score to a triple:

1. We used the frequency of head-function-value triples relative to the frequency of the head as an estimate of the probability that this head would appear with this function-value combination. We used the "expected likelihood estimate" [8] in order to assure that triples which do not appear in the training corpus are still assigned non-zero probability; this simple estimator adds 1/2 to each observed frequency:

$$\text{score} = \frac{\text{freq. of triple} + 0.5}{\text{freq. of head} + 0.5}$$

2. We applied a threshold to our set of collected triples: if a triple appeared with a frequency above the threshold it was assigned one score; if at or below the threshold, a lower score. We selected a threshold of 0.9, so that any triple which appeared unambiguously in at least one sentence of the training corpus was included. For our scores, we used the results of our previous set of experiments. These experiments showed that at a threshold of 0.9, 82% of the triples above the threshold were semantically valid, while 47% of the triples below the threshold were valid.[2] Thus we used

$$\begin{aligned}\text{score} = \ & 0.82 \text{ if freq. of triple} > 0.9\\ & 0.47 \text{ if freq. of triple} \leq 0.9\end{aligned}$$

3. We expanded on method 2 by using both triples and pairs information. To assign a score to a head-function-value triple, we first ascertain whether this triple appears with frequency $> T$ in the collected patterns; if so, we assign a high score to the triple. If not, we determine whether the head-function pair appears with frequency $> T$ in the collected patterns. If so, we assign an intermediate score to the triple; if not, we assign a low score to the triple. Again, we chose a threshold of 0.9 for both triples and pairs. Our earlier experiments indicated that, of those head-function-value triples for which the triple was below the threshold for triples frequency but the head-function pair was above the threshold for pair frequency, 52% were semantically valid. Of those for which the head-function pair was below the threshold for pair frequency, 40% were semantically valid. Thus we used

$$\begin{aligned}\text{score} = \ & 0.82 \text{ if freq. of triple} > 0.9\text{, else}\\ & 0.52 \text{ if freq. of pair} > 0.9\text{, else}\\ & 0.40 \text{ if freq. of pair} \leq 0.9\end{aligned}$$

Using these three scoring functions for selection, we parsed our test set of sentences and then scored the resulting parses against our "standard parses". As a further comparison, we also parsed the same set using selectional constraints which had been previously manually prepared for this domain. The parses were scored against the standard in terms of average recall, precision, and number of crossings; the results are shown in Table 1.[3] A better match to the correct parses

[2] The actual value of the scores only matters in cases where one parse generates more triples than another.

[3] These averages are calculated only over the subset of test sentences which yielded a parse with our grammar within the edge limit alloted.

selection strategy	crossings	recall	precision
1. frequency-based	2.00	75.70	71.86
2. triples-threshold	2.17	73.57	70.22
3. triples-and-pairs	2.09	74.33	70.94
3. hand-generated	2.04	74.34	70.79

Table 1: A comparison of the effect of different selection strategies on the quality of parses generated.

is reflected in higher recall and precision and lower number of crossings. These results indicate that the frequency-based scores performed better than either the threshold-based scores or the manually-prepared selection.

7 Related Work

At NYU we have long been interested in the possibilities of automatically acquiring sublanguage (semantic) word classes and patterns from text corpora. In 1975 we reported on experiments — using a few hundred manually prepared regularized parses — for clustering words based on their co-occurrence patterns and thus generating the principal sublanguage word classes for a domain [1]. In the early 1980's we performed experiments, again with relatively small corpora and machine-generated (but manually selected) parses, for collecting sublanguage patterns, similar to the work reported here [9]. By studying the growth curves of size of text sample vs. number of patterns, we attempted to estimate at that time the completeness of the sublanguage patterns we obtained.

More recently there has been a surge of interest in such corpus-based studies of lexical co-occurrence patterns (e.g., [10,11,12,13]). The recent volume edited by Zernik [14] reviews many of these efforts. We mention only two of these here, one seeking a similar range of patterns, the other using several evaluation methods.

Velardi et al. [11] are using co-occurence data to build a "semantic lexicon" with information about the conceptual classes of the arguments and modifiers of lexical items. This information is closely related to our selectional patterns, although the functional relations are semantic or conceptual whereas ours are syntactic. They use manually-encoded coarse-grained selectional constraints to limit the patterns which are generated. No evaluation results are yet reported.

Hindle and Rooth [10] have used co-occurrence data to determine whether prepositional phrases should be attached to a preceding noun or verb. Unambiguous cases in the corpus are identified first; co-occurrence statistics based on these are then used iteratively to resolve ambiguous cases. A detailed evaluation of the predictive power of the resulting patterns is provided, comparing the patterns against human judgements over a set of 1000 sentences, and analyzing the error rate in terms of the type of verb and noun association.

8 Conclusion

We have described two different approaches to evaluating automatically collected selectional patterns: by comparison to a set of manually-classified patterns and in terms of their effectiveness in selecting correct parses. We have shown that, without any manual selection of the parses or patterns in our training set, we are able to obtain selectional patterns of quite satisfactory recall and precision, and which perform better than a set of manual selectional patterns in selecting correct parses. We are not aware of any comparable efforts to evaluate a full range of automatically acquired selectional patterns.

Further studies are clearly needed, particularly of the best way in which the collected triples can be used for selection. The expected likelihood estimator is quite crude and more robust estimators should be tried, particularly given the sparse nature of the data. We should experiment with better ways of combining of triples and pairs data to give estimates of semantic validity. Finally, we need to explore ways of combining these automatically collected patterns with manually generated selectional patterns, which will probably have narrower coverage but may be more precise and complete for the verbs covered.

9 Acknowledgements

This report is based upon work supported by the Defense Advanced Research Projects Agency under Grant N00014-90-J-1851 from the Office of Naval Research and by the National Science Foundation under Grant IRI-89-02304.

References

[1] Lynette Hirschman, Ralph Grishman, and Naomi Sager. Grammatically-based automatic word class formation. *Information Processing and Management*, 11(1/2):39–57, 1975.

[2] Mahesh Chitrao and Ralph Grishman. Statistical parsing of messages. In *Proceedings of the Speech and Natural Language Workshop*, pages 263–266, Hidden Valley, PA, June 1990. Morgan Kaufmann.

[3] J. K. Baker. Trainable grammars for speech recognition. In D. H. Klatt and J. J. Wolf, editors, *Speech Communication Papers for the 97th Meeting of the Acoustic Society of America*, 1979.

[4] Beth Sundheim. Third message understanding evaluation and conference (MUC-3): Phase 1 status report. In *Proceedings of the Speech and Natural Language Workshop*, pages 301–305, Pacific Grove, CA, February 1991. Morgan Kaufmann.

[5] Ezra Black, Steven Abney, Dan Flickenger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Fred Jelinek, Judith Klavans, Mark Liberman, Mitch Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English. In *Proceedings of the Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, CA, February 1991. Morgan Kaufmann.

[6] Philip Harrison, Steven Abney, Ezra Black, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Donald Hindle, Robert Ingria, Mitch Marcus, Beatrice Santorini, and Tomek Strzalkowski. Evaluating syntax performance of parser/grammars. In *Proceedings of the Natural Language Processing Systems Evaluation Workshop*, Berkeley, CA, June 1991. To be published as a Rome Laboratory Technical Report.

[7] Ralph Grishman, Catherine Macleod, and John Sterling. Evaluating parsing strategies using standardized parse files. In *Proc. Third Conf. on Applied Natural Language Processing*, Trento, Italy, April 1992.

[8] William Gale and Kenneth Church. Poor estimates of context are worse than none. In *Proceedings of the Speech and Natural Language Workshop*, pages 283–287, Hidden Valley, PA, June 1990. Morgan Kaufmann.

[9] R. Grishman, L. Hirschman, and N.T. Nhan. Discovery procedures for sublanguage selectional patterns: Initial experiments. *Computational Linguistics*, 12(3):205–16, 1986.

[10] Donald Hindle and Mats Rooth. Structural ambiguity and lexical relations. In *Proceedings of the 29th Annual Meeting of the Assn. for Computational Linguistics*, pages 229–236, Berkeley, CA, June 1991.

[11] Paola Velardi, Maria Teresa Pazienza, and Michela Fasolo. How to encode semantic knowledge: A method for meaning representation and computer-aided acquisition. *Computational Linguistics*, 17(2):153–170, 1991.

[12] Frank Smadja. From n-grams to collocations: An evaluation of xtract. In *Proceedings of the 29th Annual Meeting of the Assn. for Computational Linguistics*, pages 279–284, Berkeley, CA, June 1991.

[13] Nicolette Calzolari and Remo Bindi. Acquisition of lexical information from a large textual italian corpus. In *Proc. 13th Int'l Conf. Computational Linguistics (COLING-90)*, pages 54–59, Helsinki, Finland, August 1990.

[14] Uri Zernik, editor. *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon.* Lawrence Erlbaum Assoc., Hillsdale, NJ, 1991.