Generalizing Automatically Generated Selectional Patterns

Ralph Grishman and John Sterling

Computer Science Department, New York University
715 Broadway, 7th Floor, New York, NY 10003, U.S.A.
{grishman,sterling}@cs.nyu.edu

Abstract

Frequency information on co-occurrence patterns can be automatically collected from a syntactically analyzed corpus; this information can then serve as the basis for selectional constraints when analyzing new text from the same domain. This information, however, is necessarily incomplete. We report on measurements of the degree of selectional coverage obtained with different sizes of corpora. We then describe a technique for using the corpus to identify selectionally similar terms, and for using this similarity to broaden the selectional coverage for a fixed corpus size.

1 Introduction

Selectional constraints specify what combinations of words are acceptable or meaningful in particular syntactic relations, such as subject-verb-object or head-modifier relations. Such constraints are necessary for the accurate analysis of natural language text. Accordingly, the acquisition of these constraints is an essential yet time-consuming part of porting a natural language system to a new domain. Several research groups have attempted to automate this process by collecting co-occurrence patterns (e.g., subject-verb-object patterns) from a large training corpus. These patterns are then used as the source of selectional constraints in analyzing new text.

The initial successes of this approach raise the question of how large a training corpus is required. Any answer to this question must of course be relative to the degree of coverage required; the set of selectional patterns will never be 100% complete, so a large corpus will always provide greater coverage. We attempt to shed to some light on this question by processing a large corpus of text from a broad domain (business news) and observing how selectional coverage increases with domain size.

In many cases, there are limits on the amount of training text available. We therefore also consider how coverage can be increased using a fixed amount of text. The most straightforward acquisition procedures build selectional patterns containing only the specific word combinations found in the training corpus. Greater coverage can be obtained by generalizing from the patterns collected so that patterns with semantically related words will also be considered acceptable. In most cases this has been done using manually-created word classes, generalizing from specific words to their classes [12,1,10]. If a pre-existing set of classes is used (as in [10]), there is a risk that the classes available may not match the needs of the task. If classes are created specifically to capture selectional constraints, there may be a substantial manual burden in moving to a new domain, since at least some of the semantic word classes will be domain-specific.

We wish to avoid this manual component by automatically identifying semantically related words. This can be done using the co-occurrence data, i.e., by identifying words which occur in the same contexts (for example, verbs which occur with the same subjects and objects). From the co-occurrence data one can compute a similarity relation between words [8,7]. This similarity information can then be used in several ways. One approach is to form word clusters based on this similarity relation [8]. This approach was taken by Sekine et al. at UMIST, who then used these clusters to generalize the semantic patterns [11]. Pereira et al. [9] used a variant of this approach, "soft clusters", in which words can be members of different clusters to different degrees.

An alternative approach is to use the word similarity information directly, to infer information about the likelihood of a co-occurrence pattern from information about patterns involving similar words. This is the approach we have adopted for our current experiments [6], and which has also been employed by Dagan et al. [2]. We compute from the co-occurrence data a "confusion matrix", which measures the interchangeability of words in particular contexts. We then use the confusion matrix directly to generalize the semantic patterns.

2 Acquiring Semantic Patterns

Based on a series of experiments over the past two years [5,6] we have developed the following procedure for acquiring semantic patterns from a text corpus:

1. Parse the training corpus using a broad-coverage grammar, and regularize the parses to produce something akin to an LFG f-structure, with explicitly labeled syntactic relations such as SUBJECT and OBJECT.[1]

2. Extract from the regularized parse a series of triples of the form

head	syntactic-relation	head-of-argument /modifier

   We will use the notation $< w_i \; r \; w_j >$ for such a triple, and $< r \; w_j >$ for a relation-argument pair.

3. Compute the frequency F of each head and each triple in the corpus. If a sentence produces N parses, a triple generated from a single parse has weight 1/N in the total.

   We will use the notation $F(< w_i r w_j >)$ for the frequency of a triple, and $F_{head}(w_i)$ for the frequency with which w_i appears as a head in a parse tree.[2]

For example, the sentence

Mary likes young linguists from Limerick.

would produce the regularized syntactic structure

```
(s like (subject (np Mary))
        (object (np linguist (a-pos young)
                             (from (np Limerick)))))
```

from which the following four triples are generated:

like	subject	Mary
like	object	linguist
linguist	a-pos	young
linguist	from	Limerick

Given the frequency information F, we can then estimate the probability that a particular head w_i appears with a particular argument or modifier $< r \; w_j >$:

$$\frac{F(< w_i \; r \; w_j >)}{F_{head}(w_i)}$$

This probability information would then be used in scoring alternative parse trees. For the evaluation below, however, we will use the frequency data F directly.

Step 3 (the triples extraction) includes a number of special cases:

(a) if a verb has a separable particle (e.g., "out" in "carry out"), this is attached to the head (to create the head *carry-out*) and not treated as a separate relation. Different particles often correspond to very different senses of a verb, so this avoids conflating the subject and object distributions of these different senses.

(b) if the verb is "be", we generate a relation *be-complement* between the subject and the predicate complement.

(c) triples in which either the head or the argument is a pronoun are discarded

(d) triples in which the argument is a subordinate clause are discarded (this includes subordinate conjunctions and verbs taking clausal arguments)

(e) triples indicating negation (with an argument of "not" or "never") are ignored

3 Generalizing Semantic Patterns

The procedure described above produces a set of frequencies and probability estimates based on specific words. The "traditional" approach to generalizing this information has been to assign the words to a set of semantic classes, and then to collect the frequency information on combinations of semantic classes [12,1].

Since at least some of these classes will be domain specific, there has been interest in automating the acquisition of these classes as well. This can be done by clustering together words which appear in the same context. Starting from the file of triples, this involves:

1. collecting for each word the frequency with which it occurs in each possible context; for example, for a noun we would collect the frequency with which it occurs as the subject and the object of each verb

2. defining a similarity measure between words, which reflects the number of common contexts in which they appear

3. forming clusters based on this similarity measure

Such a procedure was performed by Sekine et al. at UMIST [11]; these clusters were then manually reviewed and the resulting clusters were used to generalize selectional patterns. A similar approach to word cluster formation was described by Hirschman et al. in 1975 [8]. More recently, Pereira et al. [9] have described a word clustering method using "soft clusters", in which a word can belong to several clusters, with different cluster membership probabilities.

Cluster creation has the advantage that the clusters are amenable to manual review and correction. On the other hand, our experience indicates that successful cluster generation depends on rather delicate adjustment of the clustering criteria. We have therefore

[1] But with somewhat more regularization than is done in LFG; in particular, passive structures are converted to corresponding active forms.

[2] Note that $F_{head}(w_i)$ is different from $F(w_i$ appears as a head in a triple) since a single head in a parse tree may produce several such triples, one for each argument or modifier of that head.

elected to try an approach which directly uses a form of similarity measure to smooth (generalize) the probabilities.

Co-occurrence smoothing is a method which has been recently proposed for smoothing n-gram models [3].[3] The core of this method involves the computation of a co-occurrence matrix (a matrix of confusion probabilities) $P_C(w_j|w_i)$, which indicates the probability of word w_j occurring in contexts in which word w_i occurs, averaged over these contexts.

$$\begin{aligned} P_C(w_j|w_i) &= \sum_s P(w_j|s)P(s|w_i) \\ &= \frac{\sum_s P(w_j|s)P(w_i|s)P(s)}{P(w_i)} \end{aligned}$$

where the sum is over the set of all possible contexts s. In applying this technique to the triples we have collected, we have initially chosen to generalize (smooth over) the first element of triple. Thus, in triples of the form *word1 relation word2* we focus on *word1*, treating *relation* and *word2* as the context:

$$\begin{aligned} & P_C(w_i|w_i') \\ &= \sum_{r,w_j} P(w_i| < r\, w_j >) \cdot P(< r\, w_j > |w_i') \\ &= \sum_{r,w_j} \frac{F(< w_i\, r\, w_j >)}{F(< r\, w_j >)} \cdot \frac{F(< w_i'\, r\, w_j >)}{F_{head}(w_i')} \end{aligned}$$

Informally, we can say that a large value of $P_C(w_i|w_i')$ indicates that w_i is selectionally (semantically) acceptable in the syntactic contexts where word w_i' appears. For example, looking at the verb "convict", we see that the largest values of $P_C(\text{convict}, x)$ are for x = "acquit" and x = "indict", indicating that "convict" is selectionally acceptable in contexts where words "acquit" or "indict" appear (see Figure 4 for a larger example).

How do we use this information to generalize the triples obtained from the corpus? Suppose we are interested in determining the acceptability of the pattern convict-object-owner, even though this triple does not appear in our training corpus. Since "convict" can appear in contexts in which "acquit" or "indict" appear, and the patterns acquit-object-owner and indict-object-owner appear in the corpus, we can conclude that the pattern convict-object-owner is acceptable too. More formally, we compute a smoothed triples frequency F_S from the observed frequency F by averaging over all words w_i', incorporating frequency information for w_i' to the extent that its contexts are also suitable contexts for w_i:

$$F_S(< w_i\, r\, w_j >) = \sum_{w_i'} P_C(w_i|w_i') \cdot F(< w_i'\, r\, w_j >)$$

In order to avoid the generation of confusion table entries from a single shared context (which quite often is the result of an incorrect parse), we apply a filter in generating P_C: for $i \neq j$, we generate a non-zero $P_C(w_j|w_i)$ only if the w_i and w_j appear in at least two common contexts, and there is some common context in which both words occur at least twice. Furthermore, if the value computed by the formula for P_C is less than some threshold τ_C, the value is taken to be zero; we have used $\tau_C = 0.001$ in the experiments reported below. (These filters are not applied for the case $i = j$; the diagonal elements of the confusion matrix are always computed exactly.) Because these filters may yeild an un-normalized confusion matrix (i.e., $\sum_{w_j} P_C(w_j|w_i) < 1$), we renormalize the matrix so that $\sum_{w_j} P_C(w_j|w_i) = 1$.

A similar approach to pattern generalization, using a similarity measure derived from co-occurrence data, has been recently described by Dagan et al. [2]. Their approach differs from the one described here in two significant regards: their co-occurrence data is based on linear distance within the sentence, rather than on syntactic relations, and they use a different similarity measure, based on mutual information. The relative merits of the two similarity measures may need to be resolved empirically; however, we believe that there is a virtue to our non-symmetric measure, because substitutibility in selectional contexts is not a symmetric relation.[4]

4 Evaluation

4.1 Evaluation Metric

We have previously [5] described two methods for the evaluation of semantic constraints. For the current experiments, we have used one of these methods, where the constraints are evaluated against a set of manually classified semantic triples.[5]

For this evaluation, we select a small test corpus separate from the training corpus. We parse the corpus, regularize the parses, and extract triples just as we did for the semantic acquisition phase. We then manually classify each triple as valid or invalid, depending on whether or not it arises from the correct parse for the sentence.[6]

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

[3] We wish to thank Richard Schwartz of BBN for referring us to this method and article.

[4] If w_1 allows a broader range of arguments than w_2, then we can replace w_2 by w_1, but not vice versa. For example, we can replace "speak" (which takes a human subject) by "sleep" (which takes an animate subject), and still have a selectionally valid pattern, but not the other way around.

[5] This is similar to tests conducted by Pereira et al. [9] and Dagan et al. [2]. The cited tests, however, were based on selected words or word pairs of high frequency, whereas our test sets involve a representative set of high and low frequency triples.

[6] This is a different criterion from the one used in our earlier papers. In our earlier work, we marked a triple as valid if it *could* be valid in some sentence in the domain. We found that it was very difficult to apply such a standard consistently, and have therefore changed to a criterion based on an individual sentence.

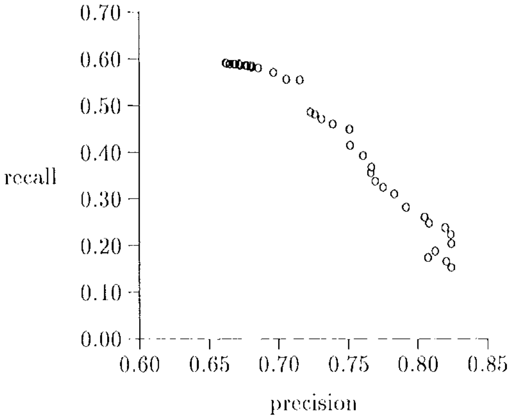

Figure 1: Recall/precision trade-off using entire corpus.

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 $> T$

and then define

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

By varying the threshold, we can select different trade-offs of recall and precision (at high threshold, we select only a small number of triples which appeared frequently and in which we therefore have high confidence, thus obtaining a high precision but low recall; conversely, at a low threshold we admit a much larger number of triples, obtaining a high recall but lower precision).

4.2 Test Data

The training and test corpora were taken from the Wall Street Journal. In order to get higher-quality parses of these sentences, we disabled some of the recovery mechanisms normally used in our parser. Of the 57,366 sentences in our training corpus, we obtained complete parses for 34,414 and parses of initial substrings for an additional 12,441 sentences. These parses were then regularized and reduced to triples. We generated a total of 279,233 distinct triples from the corpus.

The test corpus used to generate the triples which were manually classified consisted of 10 articles, also from the Wall Street Journal, distinct from those in the training set. These articles produced a test set containing a total of 1932 triples, of which 1107 were valid and 825 were invalid.

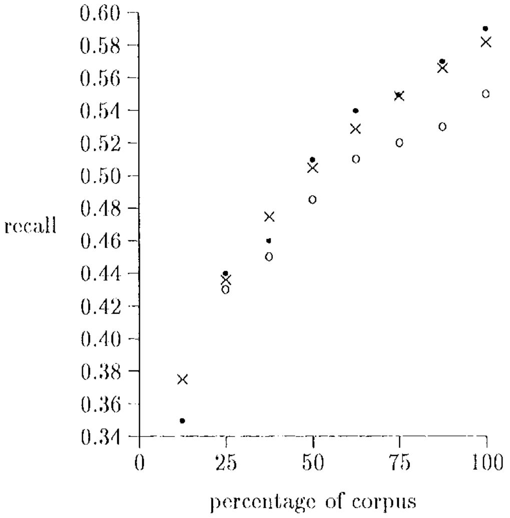

Figure 2: Growth of recall as a function of corpus size (percentage of total corpus used). ○ = at 72% precision; ● = maximum recall, regardless of precision; × = predicted values for maximum recall

4.3 Results

4.3.1 Growth with Corpus Size

We began by generating triples from the entire corpus and evaluating the selectional patterns as described above; the resulting recall/precision curve generated by varying the threshold is shown in Figure 1.

To see how pattern coverage improves with corpus size, we divided our training corpus into 8 segments and computed sets of triples based on the first segment, the first two segments, etc. We show in Figure 2 a plot of recall vs. corpus size, both at a constant precision of 72% and for maximum recall regardless of precision.[7]

The rate of growth of the maximum recall can be understood in terms of the frequency distribution of triples. In our earlier work [4] we fit the growth data to curves of the form $1 - \exp(-\beta x)$, on the assumption that all selectional patterns are equally likely. This may have been a roughly accurate assumption for that application, involving semantic-class based patterns (rather than word-based patterns), and a rather sharply circumscribed sublanguage (medical reports). For the (word-level) patterns described here, however, the distribution is quite skewed, with a small number of very-high-frequency patterns,[8] which results in dif-

[7] No data point is shown for 72% precision for the first segment alone because we are not able to reach a precision of 72% with a single segment.

[8] The number of high-frequency patterns is accentuated by

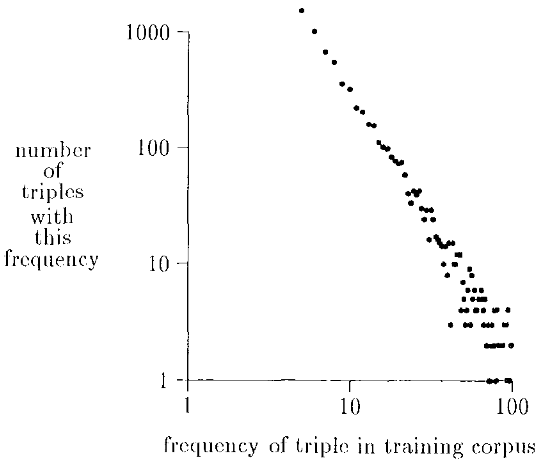

Figure 3: Distribution of frequencies of triples in training corpus. Vertical scale shows number of triples with a given frequency.

ferent growth curves. Figure 3 plots the number of distinct triples per unit frequency, as a function of frequency, for the entire training corpus. This data can be very closely approximated by a function of the form $N(F) = aF^{-\alpha}$, where $\alpha = 2.2$.[9]

To derive a growth curve for maximum recall, we will assume that the frequency distribution for triples selected at random follows the same form. Let $p(T)$ represent the probability that a triple chosen at random is a particular triple T. Let $P(p)$ be the density of triples with a given probability; i.e., the number of triples with probabilities between p and $p + \epsilon$ is $\epsilon P(p)$ (for small ϵ). Then we are assuming that $P(p) = \kappa p^{-\alpha}$, for p ranging from some minimum probability p_{min} to 1. For a triple T, the probability that we would find at least one instance of it in a corpus of τ triples is approximately $1 - e^{-\tau p(T)}$. The maximum recall for a corpus of τ triples is the probability of a given triple (the "test triple") being selected at random, multiplied by the probability that that triple was found in the training corpus, summed over all triples:

$$\sum_{T} p(T) \cdot (1 - e^{-\tau p(T)})$$

which can be computed using the density function

$$\int_{p_{min}}^{1} p \cdot P(p) \cdot (1 - e^{-\tau p}) dp$$
$$= \int_{p_{min}}^{1} \kappa p \cdot p^{-\alpha} (1 - e^{-\tau p}) dp$$

By selecting an appropriate value of κ (and corresponding p_{min} so that the total probability is 1), we can get a good match to the actual maximum recall values; these computed values are shown as × in Figure 2. Except for the smallest data set, the agreement is quite good considering the very simple assumptions made.

w	$P_C(bond\|w)$
eurobond	0.133
foray	0.128
mortgage	0.093
objective	0.089
marriage	0.071
note	0.068
maturity	0.057
subsidy	0.046
veteran	0.046
commitment	0.046
debenture	0.044
activism	0.043
mile	0.038
coupon	0.038
security	0.037
yield	0.036
issue	0.035

Figure 4: Nouns closely related to the noun "bond", ranked by P_C.

4.3.2 Smoothing

In order to increase our coverage (recall), we then applied the smoothing procedure to the triples from our training corpus. In testing our procedure, we first generated the confusion matrix P_C and examined some of the entries. Figure 4 shows the largest entries in P_C for the noun "bond", a common word in the Wall Street Journal. It is clear that (with some odd exceptions) most of the words with high P_C values are semantically related to the original word.

To evaluate the effectiveness of our smoothing procedure, we have plotted recall vs. precision graphs for both unsmoothed and smoothed frequency data. The results are shown in Figure 5. Over the range of precisions where the two curves overlap, the smoothed data performs better at low precision/high recall, whereas the unsmoothed data is better at high precision/low recall. In addition, smoothing substantially extends the level of recall which can be achieved for a given corpus size, although at some sacrifice in precision.

Intuitively we can understand why these curves should cross as they do. Smoothing introduces a certain degree of additional error. As is evident from Figure 4, some of the confusion matrix entries are spurious, arising from such sources as incorrect parses and the conflation of word senses. In addition, some of the triples being generalized are themselves incorrect (note that even at high threshold the precision is below 90%). The net result is that a portion (roughly 1/3 to 1/5) of

the fact that our lexical scanner replaces all identifiable company names by the token *a-company*, all currency values by *a-currency*, etc. Many of the highest frequency triples involve such tokens.

[9] This is quite similar to a Zipf's law distribution, for which $\alpha = 2$.

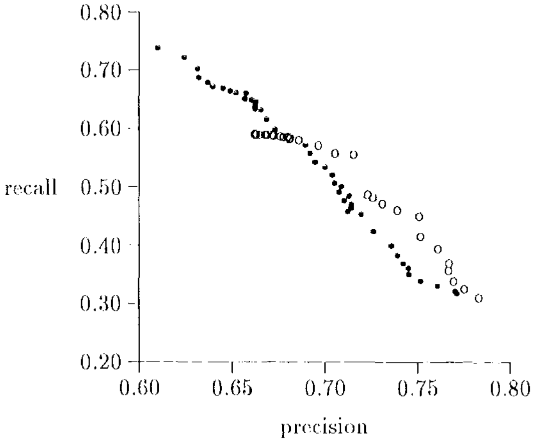

Figure 5: Benefits of smoothing for largest corpus: o = unsmoothed data, • = smoothed data.

the triples added by smoothing are incorrect. At low levels of precision, this produces a net gain on the precision/recall curve; at higher levels of precision, there is a net loss. In any event, smoothing does allow for substantially higher levels of recall than are possible without smoothing.

5 Conclusion

We have demonstrated how selectional patterns can be automatically acquired from a corpus, and how selectional coverage gradually increases with the size of the training corpus. We have also demonstrated that — for a given corpus size — coverage can be significantly improved by using the corpus to identify selectionally related terms, and using these similarities to generalize the patterns observed in the training corpus. This is consistent with other recent results using related techniques [2,9].

We believe that these techniques can be further improved in several ways. The experiments reported above have only generalized over the first (head) position of the triples; we need to measure the effect of generalizing over the argument position as well. With larger corpora it may also be feasible to use larger patterns, including in particular subject-verb-object patterns, and thus reduce the confusion due to treating different words senses as common contexts.

References

[1] Jing-Shin Chang, Yih-Fen Luo, and Keh-Yih Su. GPSM: A generalized probabilistic semantic model for ambiguity resolution. In *Proceedings of the 30th Annual Meeting of the Assn. for Computational Linguistics*, pages 177–184, Newark, DE, June 1992.

[2] Ido Dagan, Shaul Marcus, and Shaul Markovitch. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting of the Assn. for Computational Linguistics*, pages 31–37, Columbus, OH, June 1993.

[3] U. Essen and V. Steinbiss. Cooccurrence smoothing for stochastic language modeling. In *ICASSP92*, pages I–161 – I–164, San Francisco, CA, May 1992.

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

[5] Ralph Grishman and John Sterling. Acquisition of selectional patterns. In *Proc. 14th Int'l Conf. Computational Linguistics (COLING 92)*, Nantes, France, July 1992.

[6] Ralph Grishman and John Sterling. Smoothing of automatically generated selectional constraints. In *Proceedings of the Human Language Technology Workshop*, Princeton, NJ, March 1993.

[7] Donald Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Assn. for Computational Linguistics*, pages 268–275, June 1990.

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

[9] Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Assn. for Computational Linguistics*, pages 183–190, Columbus, OH, June 1993.

[10] Philip Resnik. A class-based approach to lexical discovery. In *Proceedings of the 30th Annual Meeting of the Assn. for Computational Linguistics*, Newark, DE, June 1992.

[11] Satoshi Sekine, Sofia Ananiadou, Jeremy Carroll, and Jun'ichi Tsujii. Linguistic knowledge generator. In *Proc. 14th Int'l Conf. Computational Linguistics (COLING 92)*, pages 560–566, Nantes, France, July 1992.

[12] 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.