SMOOTHING OF AUTOMATICALLY GENERATED SELECTIONAL CONSTRAINTS

Ralph Grishman and John Sterling

Department of Computer Science
New York University
New York, NY 10003

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. Better coverage of the domain can be obtained by appropriate generalization of the specific word patterns which are collected. We report here on an approach to automatically make suitable generalizations: using the co-occurrence data to compute a confusion matrix relating individual words, and then using the confusion matrix to smooth the original frequency data.

1. INTRODUCTION

Semantic (selectional) 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.

However, the patterns collected in this way involve specific word combinations from the training corpus. Unless the training corpus is very large, this will provide only limited coverage of the range of acceptable semantic combinations, even within a restricted domain. In order to obtain better coverage, it will be necessary to generalize from the patterns collected so that patterns with semantically related words will also be considered acceptable. In most cases this has been done by manually assigning words to semantic classes and then generalizing from specific words to their classes. This approach still implies a substantial manual burden in moving to a new domain, since at least some of the semantic word classes will be domain-specific.

In order to fully automate the process of semantic constraint acquisition, we would like to be able to automatically identify semantically related words. This can be done using the co-occurrence data, 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, and then cluster words of high similarity. This approach was taken by Sekine et al. at UMIST, who then used these clusters to generalize semantic patterns [6]. A similar approach to word clustering was reported by Hirschman et al. in 1975 [5].

For our current experiments, we have adopted a slightly different approach. We compute from the co-occurrence data a confusion matrix, which also measures the interchangeability of words in particular contexts. We then use the confusion matrix directly to generalize the semantic patterns.

2. THE NATURE OF THE CONSTRAINTS

The constraints we wish to acquire are local semantic constraints; more specifically, constraints on which words can occur together in specific syntactic relations. These include head-argument relations (e.g., subject-verb-object) and head-modifier relations. Some constraints may be general (domain independent), but others will be specific to a particular domain. Because it is not practical to state all the allowable word combinations, we normally place words into (semantic) word classes and then state the constraints in terms of allowable combinations of these classes.

When these constraints were encoded by hand, they were normally stated as absolute constraints—a particular combination of words was or was not acceptable. With corpus-derived constraints, on the other hand, it becomes possible to think in terms of a probabilistic model. For example, based on a training corpus, we would estimate the probability that a particular verb occurs with a particular subject and object (or with subject and object from particular classes), or that a verb occurs with a particular modifier. Then, using the (obviously crude) assumption of independent probabilities, we would estimate the probability of a particular sentence derivation as the product of the probabilities of all the operations (adding arguments to heads, adding modifiers to heads) required to produce the sentence, and the probability of a sentence as the sum of the probabilities of its derivations.

3. ACQUIRING SEMANTIC PATTERNS

Based on a series of experiments over the past year (as reported at COLING-92) we have developed the following procedure for acquiring semantic patterns from a text corpus:

1. Using unsupervised training methods, create a stochastic grammar from a (non-stochastic) augmented context-free grammar. Use this stochastic grammar to parse the training corpus, taking only the most probable parse(s) of each sentence.
2. Regularize the parses to produce something akin to an LFG f-structure, with explicitly labeled syntactic relations such as SUBJECT and OBJECT.[1]
3. Extract from the regularized parse a series of triples of the form

   head syntactic-relation arg

   where *arg* is the head of the argument or modifier. We will use the notation $< w_i\ r\ w_j >$ for such a triple, and $< r\ w_j >$ for a relation-argument pair.
4. 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.

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 $< r\ w_j >$:[2]

$$\frac{F(< w_i\ r\ w_j >)}{F(w_i \text{ appears as a head in a parse tree})}$$

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.

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

[2] Note that $F(w_i$ appears as a head in a parse tree) 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.

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

4. 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 [7,3].

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 [6]; 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 [5].

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 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 [4].[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. For an n-gram model, for example, the context might be the set of $n-1$ prior words. This matrix can be used to take a basic trigram model $P_B(w_n|w_{n-2}, w_{n-1})$ and produce a smoothed model

$$P_S(w_n|w_{n-2}, w_{n-1}) = \sum_{w'_n} P_C(w_n|w'_n)P_B(w'_n|w_{n-2}, w_{n-1})$$

We have used this method in a precisely analogous way to compute smoothed semantic triples frequencies, F_S. In triples of the form *word1 relation word2* we have initially chosen to smooth over *word1*, treating *relation* and *word2* as the context.

$$\begin{aligned} P_C(w_i|w'_i) &= \sum_{r,w_j} P(w_i| < r\ w_j >) \cdot P(< r\ w_j > |w'_i) \\ &= \sum_{r,w_j} \frac{F(< w_i\ r\ w_j >)}{F(< r\ w_j >)} \\ &\quad \cdot \frac{F(< w'_i\ r\ w_j >)}{F(w'_i \text{ appears as a head of a triple})} \end{aligned}$$

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

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 $\sum_{w_j} P_C(w_j|w_i) = 1$.

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

5. EVALUATION

5.1. Evaluation Metric

We have previously (at COLING-92) 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.

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 (with the exception that we use the non-stochastic grammar in order to generate all grammatically valid parses of each sentence). We then manually classify each triple as semantically valid or invalid (a triple is counted as valid if we believe that this pair of words could meaningfully occur in this relationship, even if this was not the intended relationship in this particular text).

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

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 define

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

w	$P_C(attack\|w)$
harden	0.252
attack	0.251
assault	0.178
dislodge	0.131
torture	0.123
harass	0.114
machinegun	0.096
massacre	0.094
reinforce	0.093
board	0.091
abduct	0.086
specialize	0.076
occupy	0.072
engage	0.068
blow-up	0.064
blow	0.063

w	$P_C(terrorist\|w)$
terrorist	0.309
ally	0.137
job	0.119
world	0.091
ceasefire	0.069
commando	0.058
guerrilla	0.045
urban commando	0.043
coup	0.043
assassin	0.041
individual	0.035
journalist	0.029
offensive	0.029
history	0.026
rebel	0.025
fighter	0.023

Figure 1: Verbs closely related to the verb "attack" and nouns closely related to the noun "terrorist", ranked by P_C. ("harden" appears at the top of the list for "attack" because both appear with the object "position".)

At a given threshold T, our smoothing process should increase recall but in practice will also increase the error rate. How can we tell if our smoothing is doing any good? We can view the smoothing process as moving some triples from v_- to v_+ and from i_- to i_+.[4] Is it doing so better than some random process? I.e., is it preferentially raising valid items above the threshold? To assess this, we compute (for a fixed threshold) the quality measure

$$Q = \frac{\frac{v_+^S - v_+}{v_-}}{\frac{i_+^S - i_+}{i_-}}$$

where the values with superscript S represent the values with smoothing, and those without superscripts represent the values without smoothing. If $Q > 1$, then smoothing is doing better than a random process in identifying valid triples.

5.2. Test Data

The training corpus was the set of 1300 messages (with a total of 18,838 sentences) which constituted the development corpus for Message Understanding Conferences - 3 and 4 [1,2]. These messages are news reports from the Foreign Broadcast Information Service concerning terrorist activity in Central and South America. The average sentence length is about 24 words. In order to get higher-quality parses of these sentences, we disabled several of the recovery mechanisms normally used in parsing, such as longest-substring parsing; with these mechanisms disabled, we obtained parses for 9,903 of the 18,838 sentences. These parses were then regularized and reduced to triples. We generated a total of 46,659 distinct triples from this test corpus.

The test corpus—used to generate the triples which were manually classified—consisted of 10 messages of similar style, taken from one of the test corpora for Message Understanding Conference - 3. These messages produced a test set containing a total of 636 distinct triples, of which 456 were valid and 180 were invalid.

5.3. Results

In testing our smoothing procedure, we first generated the confusion matrix P_C and examined some of the entries. Figure 1 shows the largest entries in P_C for the verb "attack" and the noun "terrorist", two very common words in the terrorist domain. 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 smoothing, we have compared three sets of triples frequency data:

1. the original (unsmoothed) data
2. the data as smoothed using P_C
3. the data as generalized using a manually-prepared classification hierarchy for a subset of the words of the domain

[4] In fact, some triples will move above the threshold and other will move below the threshold, but in the regions we are considering, the net movement will be above the threshold.

generalization strategy	T	v_+	v_-	i_+	i_-	recall	error rate	Q
1. no smoothing	0	139	317	13	167	30%	7%	
2. confusion matrix	0	237	219	50	130	52%	28%	1.39
3. classification hierarchy	0	154	302	18	162	34%	10%	1.58
4. confusion matrix	0.29	154	302	17	163	34%	9%	1.90

Table 1: A comparison of the effect of different generalization strategies.

For the third method, we employed a classification hierarchy which had previously been prepared as part of the information extraction system used for Message Understanding Conference-4. This hierarchy included only the subset of the vocabulary thought relevant to the information extraction task (not counting proper names, roughly 10% of the words in the vocabulary). From this hierarchy we identified the 13 classes which were most frequently referred to in the lexico-semantic models used by the extraction system. If the head (first element) of a semantic triple was a member of one of these classes, the generalization process replaced that 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); to make the results comparable to those with confusion-matrix smoothing, we did not generalize the argument (last element) of the triple.

The basic results are shown in rows 1, 2, and 3 of Table 1. For all of these we used a threshold (T) of 0, so a triple with any frequency > 0 would go into the v_+ or i_+ category. In each case the quality measure Q is relative to the run without smoothing, entry 1 in the table. Both the confusion matrix and the classification hierarchy yield Qs substantially above 1, indicating that both methods are performing substantially better than random. The Q is higher with the classification hierarchy, as might be expected since it has been manually checked; on the other hand, the improvement in recall is substantially smaller, since the hierarchy covers only a small portion of the total vocabulary. As the table shows, the confusion matrix method produces a large increase in recall (about 73% over the base run).

These comparisons all use a T (frequency threshold) of 0, which yields the highest recall and error rate. Different recall/error-rate trade-offs can be obtained by varying T. For example, entry 4 of the table shows the result for T=0.29, the point at which the recall using the confusion matrix and the classification hierarchy is the same (the values without smoothing and the values using the classification hierarchy are essentially unchanged at T=0.29). We observe that, for the same recall, the automatic smoothing does as well as the manually generated hierarchy with regard to error rate. (In fact, the Q value with smoothing (line 4) is much higher than with the classification hierarchy (line 3), but this reflects a difference of only 1 in i_+ and should not be seen as significant.)

6. DISCUSSION

We have demonstrated that automated smoothing methods can be of some benefit in increasing the coverage of automatically acquired selectional constraints. This is potentially important as a step in developing tools for porting natural language systems to new domains. It is still too early to assess the relative merits of different approaches to generalizing these selectional constraints, given our limited testing and the different evaluation metrics of the few others groups experimenting with such acquisition procedures.

Our experimental results are not uniformly positive. We did achieve substantially higher recall levels with smoothing. On the other hand, over the range of recalls obtainable without smoothing, smoothing did not consistently improve the error rate. Therefore at present the principal benefit of the smoothing technique is to raise the recall beyond that possible using unsmoothed data.

In addition, preliminary experiments with smoothing applied to the *argument* position in a triple indicate that the comparison between automated smoothing and manual classification hierarchies is not so favorable. This is not too surprising because when the classification hierarchy was initially created, its primary use was to specify the allowable values of arguments and modifiers in semantic case frames; as a result, while the hierarchy is of benefit in generalizing heads (as described above), it is more effective in generalizing the argument position.

We recognize that the size of the corpus we have used is quite minimal for the task of computing similarities, since to get a fully populated similarity matrix we would require each pair of semantically related words to occur in several common contexts. We hope therefore to repeat these experiments with a substantially larger corpus in the near future. A larger corpus will also allow us 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.

7. Acknowledgement

This material is based upon work supported by the Advanced Research Projects Agency through the Office of Naval Research under Grant No. N00014-90-J-1851.

References

1. *Proceedings of the Third Message Understanding Conference (MUC-3).* Morgan Kaufmann, May 1991.
2. *Proceedings of the Fourth Message Understanding Conference (MUC-4).* Morgan Kaufmann, June 1992.
3. 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.
4. U. Essen and V. Steinbiss. Cooccurrence smoothing for stochastic language modeling. In *ICASSP92*, pages I–161 – I–164, San Francisco, CA, May 1992.
5. Lynette Hirschman, Ralph Grishman, and Naomi Sager. Grammatically-based automatic word class formation. *Information Processing and Management*, 11(1/2):39–57, 1975.
6. 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.
7. 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.