Toward Unsupervised Whole-Corpus Tagging

Dayne FREITAG
HNC Software, LLC
3661 Valley Centre Drive
San Diego, CA 92130
USA
DayneFreitag@fairisaac.com

Abstract

We present a system for unsupervised tagging of words into classes produced by a distributional clustering technique called *co-clustering*. A hidden Markov model (HMM), trained on the high-frequency terms in the lexicon, is used to "tag" occurrences of low-frequency terms. In experiments using the Wall Street Journal portion of the Penn Treebank, we show that previously reported problems in using Baum-Welch estimation for part-of-speech tagging do not occur in this context. We also show how state-level term emission models can be augmented to account for morphological patterns using features automatically derived from the output of co-clustering. Finally, we consider an alternative means of extending the coverage of the lexicon, in which low-frequency terms are added to the lexicon as *types*, and compare this approach with the *token-level* assignments made by the HMM.

1 Introduction

Part-of-speech (POS) tagging is a necessary prerequisite for many text processing applications, and a wide variety of syntactic lexical resources and taggers are available for languages in which the problem has been studied. However, using a specific tagger and its tag set entails adopting the assumptions it embodies, which may not be appropriate for a target application. In the worst case, the domain of interest may include text in a language not covered by available taggers. Even when a tagger is available, the domain may involve usages substantially different from those in the corpus for which the tagger was developed. Many current taggers are tuned to relatively formal corpora, such as newswire, while many interesting domains, such as email, netnews, or physicians' notes, are replete with elisions, jargon, and neologisms.

Fortunately, using distributional characteristics of term contexts, it is feasible to induce categories having high agreement with part of speech directly from a corpus of sufficient size, as a number of studies have shown (Brown et al., 1992; Schütze, 1995; Clark, 2000). While the categories induced in this way do not always agree perfectly with prior syntactic categories, they are specific to the corpus of interest, reflect the predominant usages in that corpus, handle neologisms seamlessly, and often have a semantic dimension which it should be possible to exploit.

bush peters reagan noriega ...
john robert james david ...
president chairman head owner ...
japan california london chicago ...

Table 1: Sample members of four clusters from the Wall Street Journal corpus.

We employ a method called *co-clustering*, described in Section 2. While we judge our clusters according to their agreement with part of speech, we do not expect unsupervised approaches to POS tagging to replace supervised ones anytime soon. Instead, the near-term promise of these methods is more effective information retrieval and information extraction. Table 1 shows sample terms from several clusters induced from the Penn Treebank's Wall Street Journal corpus. We believe the ability to form such classes automatically will enable effective named entity recognition requiring much lighter supervision than is currently needed. In producing these clusters, we use only the most frequent terms in the corpus, for both efficiency and statistical reliability. As results later in the paper show, quality of cluster assignment decreases with a term's corpus frequency. Nevertheless, to make these induced categories generally useful, we must be able to categorize every token in a corpus.

The remainder of the paper explores several

ways to extend the reach of the *distributional lexicon* (the output of co-clustering). Section 3 describes how we use a second-order hidden Markov model and Baum-Welch re-estimation on a partially labeled corpus to "tag" the whole corpus. Section 4 improves this model with morphological features automatically learned from the distributional lexicon. Finally, in Section 5, we explore an alternative to the HMM: adding low-frequency terms to the lexicon based on their type-level contextual behavior.

2 Co-Clustering

As in Brown, et al (1992), we seek a partition of the vocabulary that maximizes the mutual information between term categories and their contexts. We achieve this in the framework of *information theoretic co-clustering* (Dhillon et al., 2003).

The input to our algorithm is two finite sets of symbols, say $X = \{x_0, x_1, \cdots, x_{N_X}\}$ (e.g., terms) and $Y = \{y_0, y_1, \cdots, y_{N_Y}\}$ (e.g., term contexts), together with a set of co-occurrence count data consisting of a non-negative integer $n_{x_i y_j}$ for every pair of symbols (x_i, y_j) from X and Y. The output is two partitions: $X^* = \{x_0^*, ..., x_{N_{X^*}}^*\}$ and $Y^* = \{y_0^*, ..., y_{N_{Y^*}}^*\}$, where each x_i^* is a subset of X (a "cluster"), and each y_i^* a subset of Y. The co-clustering algorithm chooses the partitions X^* and Y^* to (locally) maximize the mutual information between them, under a constraint limiting the total number of clusters in each partition.

Recall that the *entropy* or *Shannon information* of a discrete distribution is:

$$\mathrm{I}_X = -\sum_x \mathrm{P}(x) \ln \mathrm{P}(x). \qquad (1)$$

This quantifies average improvement in one's knowledge upon learning the specific value of an event drawn from X. The *mutual information* between random variables X and Y can be written:

$$\mathrm{M}_{XY} = \sum_{xy} \mathrm{P}(x,y) \ln \frac{\mathrm{P}(x,y)}{\mathrm{P}(x)\mathrm{P}(y)} \qquad (2)$$

This quantifies the amount that one expects to learn indirectly about X upon learning the value of Y, or vice versa.

2.1 The Algorithm

We perform an approximate maximization of $M_{X^*Y^*}$ using a simulated annealing procedure in which each trial move takes a symbol x or y out of the cluster to which it is tentatively assigned and places it into another. The source cluster, member element, and destination cluster of each candidate move are each chosen uniformly at random. When temperature 0 is reached, all possible moves are repeatedly attempted until no move leads to an increase in the objective function.

2.2 Evaluation Methodology

While the clusters induced by this method display high agreement with part of speech, the resulting set of categories is simultaneously more and less granular than that defined by a typical tagged corpus, such as the Penn Treebank. An example of their greater granularity is the distinction between first and last names shown in Table 1. At the same time, the treebank distinction between "`NN`" (noun) and "`NNP`" (proper noun), which often only amounts to a difference in capitalization, is not generally reflected in the clusters, which are formed from case-normalized data. Schütze (1995) provides a more general discussion of typical discrepancies between prior and induced categories.

Schütze also measures the *accuracy* of his method, by simplifying the problem and manually labeling clusters. We adopt a more replicable methodology. If we treat each token as a joint occurrence of a cluster and tag, the conditional entropy of tag, given cluster, may be computed as follows:

$$\mathrm{I}_{T|C} = \mathrm{I}_T - \mathrm{M}_{TC} \qquad (3)$$

This quantifies the average amount of uncertainty in tag prediction, given a clustering. A slightly more intuitive measure, borrowed by analogy from language modeling, is $\exp(I_{T|C})$, which we call the *cluster-conditional tag perplexity*. Minimum perplexity is 1.0, signifying complete certainty in tag prediction.

2.3 Results

For the purposes of this study, the context of a token was taken to be the tokens immediately to its left and right. Special tokens were inserted to denote the beginnings and ends of sentences. Left and right occurrences of a given contextual token were treated as distinct events. Using data from the Wall Street Journal corpus of the Penn Treebank, we clustered the 5000 most frequent tokens and 5000 most frequent contexts, each into 200 clusters.

Uniform	$1/V$
ML	$N(t)/N(*)$
Interpolated	$\lambda N(t)/N(*) + (1-\lambda)/V$

Table 2: Baseline token emission model variants. $N(t)$ denotes the frequency of term t; $N(*)$ denotes the total frequency of all terms. V is the size of the vocabulary.

The perplexity of tag prediction, in the absence of any additional information, on these 5000 words (which together account for about 91% of all tokens in the corpus) is about 23.6. The perplexity, given their cluster assignments, is 1.57. (Note that, because of polysemy, the best that can be achieved is 1.23.) Part of our success in achieving these numbers is due to an observed tendency of the mutual information criterion to segregate highly frequent closed-class words (e.g., "the" and "to") into singleton or small, cohesive clusters. In categorizing the many open-class unclustered terms, we naturally cannot benefit from this phenomenon.

3 HMM Tagging

3.1 Approach

We train a second-order HMM to perform assignment of novel terms to categories. In doing so, we adopt a standard framework for statistical part of speech tagging (Brants, 2000; Cutting et al., 1992; Abney, 1996), in which states correspond to category bi-grams and each category is associated with a distribution over the terms in the vocabulary. To be precise, we estimate the joint probability of a sequence of tokens (t_i) and categories (c_i) as:

$$\mathrm{P}(T, C) = \prod_{i}^{n} \mathrm{P}(c_i|c_{i-1}, c_{i-2})\mathrm{P}(t_i|c_i) \qquad (4)$$

In all experiments, we use maximum likelihood estimates of transition probabilities. Our baseline experiments estimate emission probabilities using the policies listed in Table 2: *Uniform*, in which each observed token is assigned the same probability in every state; *ML*, in which estimates are directly proportional to the frequency with which a term has been observed in a context; and *Interpolated*, in which the *Uniform* and *ML* estimates are mixed using weights set by deleted interpolation.

Starting with randomly perturbed uniform parameters, each model is trained using Baum-Welch re-estimation. In this procedure, a token of a known (clustered) term is fixed to the corresponding category, the probability of visiting model states corresponding to other categories constrained to be zero. This use of Baum-Welch differs from important earlier uses. In Cutting, et al (1992), for example, a lexicon is presumed, which, for every term, lists possible parts of speech, and Baum-Welch is used to resolve the ambiguity. This approach has been shown to be sensitive to starting conditions, and tagging accuracy typically *decreases* with each iteration of Baum-Welch (Merialdo, 1994; Elworthy, 1994). In contrast, we "know" the categories of the most frequent terms, but know nothing about infrequent terms.[1]

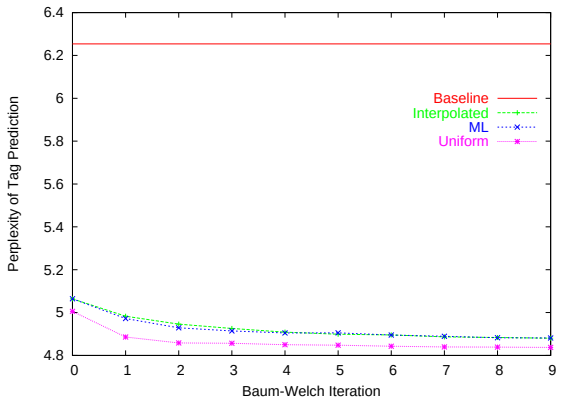

Figure 1: Cluster-conditional tag perplexity on unclustered tokens with increasing iterations of Baum-Welch.

3.2 Results

We trained three models, one for each estimation method in Table 2, on five of the twenty-five partitions of the Wall Street Journal corpus from the Penn Treebank. Baum-Welch was run for ten iterations. We also measured the performance of a baseline approach which always predicts the most frequent category, conditioned on the preceding two categories. Each model was then used to tag the training corpus, and its performance on *unclustered tokens* (i.e., instances of terms not used in clustering—about 9% of all tokens) measured.

Figure 1, which plots cluster-conditional tag perplexity against Baum-Welch iteration, displays two interesting phenomena. First, we do

[1] Note, too, that our work is not directly comparable to similar uses of fully supervised Markov models, such as Brants (2000) and Clark (2000), in which Baum-Welch is not used.

not observe any of the models losing agreement with the external tagging, as Baum-Welch progresses. This effect, which we attribute to structural differences between our problem and comparable results (e.g., (Cutting et al., 1992)), is consistent in all our experiments.

Also of interest is the relatively poor showing of the two multinomial emission policies, *ML* and *Interpolated*. Of course, these naive models have many more parameters than is typical of successful models from the literature. For example, Brants (2000) constructs estimates for all terms occurring fewer than 10 times by pooling terms with similar suffixes. In the following section, we pursue a similar idea, but aim to learn a succinct list of salient morphological features automatically without restricting our attention to suffixes. Again, we want to avoid language-specific commitments, and seek to discover the features of interest, wherever in the word they may occur (note that the German *prefix* "`ge-`" is morphologically significant).

4 Morphological Features

4.1 Learning Affixes

The idea we pursue is based on the observation that, under distributional clustering, terms having similar morphological features tend to cluster together. To the extent this is true, the corresponding features will have high mutual information with cluster assignment. By treating each of the known terms as a joint occurrence of its cluster and each of the features it matches, we can use Equation 2 to rank individual features according to their morphological salience.

Because of redundancy among the highest-scoring features, however, employing a feature set constructed simply by collecting the best features is an inefficient use of model capacity. For example, in English the suffix features `tion$` and `ion$` provide approximately the same syntactic information. Use of one renders the other largely redundant.

We focus, therefore, on finding a good *ensemble* of features. We posit a single categorical-valued meta-feature F which encompasses a list of individual features $[f_1, \cdots, f_n]$. The value of F, given a term, is the index of the first matching feature in its list, evaluated in order, or 0 if none matches. By treating F as a random variable and manipulating the members of its feature list to maximize mutual information with cluster assignment, we produce a *set* of individual features which are morphologically salient and reasonably disjoint.

```
Procedure OptFeatSet(Features, ListSize, NIts)
    List = RandomList(Features, ListSize)
    For It = 1 to NIts
        List = TryOp(SwapInFeature, List)
        List = TryOp(ResizeMember, List)
        List = TryOp(ReOrder, List)
    Return List
```

Table 3: Basic optimization procedure for constructing a list of morphological salient features.

The automatic learning of morphological affixes for use in part of speech tagging has been proposed previously (Brill, 1995; Mikheev, 1997). The approach described here is distinguished from these in its simplicity and, more importantly, in its eschewal of hand-crafted syntactic resources, such as a syntactic lexicon or tagged corpus.

We say a feature is any string of contiguous characters, of length one to five, that matches a cluster member. In addition, whenever such a string is a prefix (or suffix), we define a second feature anchored to the beginning (or end) of the term. For efficiency, we only consider features that match at least 20 cluster terms.

Table 3 lists the procedure used to construct a feature list. We produce an initial list (List) by choosing at random a specified number (ListSize) of distinct features from the candidate set (Features). Then, for a specified number of iterations (NIts), we repeatedly apply three operations, adopting any new list that has higher mutual information than the current list. The following operations are attempted:

- **SwapInFeature**. Choose at random a candidate feature not currently in the list and substitute it for a random list member.
- **ResizeMember**. Add or remove a character from either end of a random list member, as long as this results in a feature from the candidate set.
- **ReOrder**. Swap the list positions of two random list members.

4.2 Extending Emission Models

Given a feature set, in addition to the token observed at a particular point in the corpus, we now have a Boolean feature vector representing the term's morphological characteristics. We consider two principal means of incorporating

Conjunctive $\prod_{i=1}^{n} N(f_i(t))/N(*)$
Conjunctive, with term $N(t)/N(*) \cdot \prod_{i=1}^{n} N(f_i(t))/N(*)$
Mixture $\sum_{\{i\|f_i(t)\}} \lambda_i N(f_i(t))/N(*)$
Mixture, with term $\lambda_0 N(t)/N(*) + \sum_{\{i\|f_i(t)\}} \lambda_i N(f_i(t))/N(*)$

Table 4: Four emission models that incorporate Boolean token features, $f_1, \cdots, f_n$. The expression $f_i(t)$ is true or false, depending on whether term t matches feature f_i. The notation $N(x)$ stands for the number of observed occurrences of x; $N(*)$ means the sum of occurrences of any event in a particular context.

this vector, *conjunctive* and *disjunctive* (or *mixture*), as shown in Table 4. The conjunctive model treats feature measurements as independent simultaneous events, and takes the probability of their joint occurrence as the product of the individual probabilities. The mixture model, on the other hand, reflects a generative process where exactly one of the matching features, with probability equal to λ_i (set using EM), is presumed to be responsible for each observed token. In either principal variation, we can also observe or be blind to the literal term.

The methods we use to account for morphological features are prefigured in the literature. Weischedel, et al (1993), combine estimates conjunctively, while Brants (2000) employs a mixture model over all word endings of various lengths—both in a completely supervised setting. We are aware of no work conducting a comparison of the two approaches in either the supervised or unsupervised setting.

An interesting related piece of work is that of Clark (2003), in which a character-level HMM models word structure during the clustering process. In contrast, our approach produces a succinct list of relevant patterns, which can be reviewed and potentially put to other uses, without wasting capacity on non-informative aspects of word structure (i.e., stems).

4.3 Results

Table 5 shows pattern lists produced using four different list sizes. We take it as confirmation

5	`s$ ion$ ly$ ed$ ing$`
10	`ly$ s$ ed$ e$ t$ ion$ ing$ n$ l r`
15	`al$ s$ ly$ er$ e$ ed$ io y$ t$ n$ ing$ i e r o`
20	`ed$ s$ th st$ ve$ al$ c$ e$ ly$ io r$ nt$ ing$ y$ ^s d$ t$ n$ e a`

Table 5: Feature lists of various sizes automatically derived from Penn Treebank co-clustering output. Each element is the regular expression for the corresponding feature.

Model	Feature list size				Rand
	5	10	15	20	10
Partitions 00–04 (training)					
Conj.	4.06	4.07	4.13	4.05	4.70
... w/term	4.26	4.23	4.22	4.15	4.72
Mixture	4.04	4.36	4.63	4.57	4.89
... w/term	3.99	4.34	4.62	4.55	4.87
Uniform	4.84				
Partitions 10–24 (test)					
Conj.	4.15	4.19	4.22	4.17	
Mixture	4.19	4.50	4.73	4.71	
... w/term	4.07	4.47	4.72	4.67	
Uniform	5.01				

Table 6: Training and test set cluster-conditional tag perplexity of unclustered Penn Treebank tokens, using various emission models and feature sets, after ten iterations of Baum-Welch.

of the procedure that most of the patterns are anchored on the end of a term, and that most have a straightforward morphological interpretation.[2] It is also interesting that larger lists tend to include features present in smaller lists.

Table 6 shows the cluster-conditional tag perplexity of unclustered tokens using various feature sets and emission models. All models incorporating the features out-perform the models that do not use them, improving on the perplexity of the uniform model by almost a point. Beyond this general improvement, however, no clear winners emerge among modeling

[2]It is difficult to determine the import of the very general single-character features at the end of the longer lists; presumably, the procedure selects wide-coverage features with weak syntactic significance, because more specific features would cause a large number of unmatched terms to be grouped into the default category.

alternatives or feature list sizes. Between the two principal modeling variants, there is an interesting discrepancy in the value of the literal token. While the mixture model appears to benefit a small amount from its use, the conjunctive model is clearly better off without it.

The mixture model shows greater sensitivity to the feature set used, while the conjunctive model performs more or less consistently with all feature sets. Perplexity of the mixture model appears to trend upward as a function of feature list size, but this result is not entirely consistent with the results of experiments we have conducted with other corpora. Greater variability (as well as best overall performance) *does* appear to be a characteristic of the mixture model.

In an effort to determine how much of the improvement is due to the syntactic significance of the features, we ran the list optimizer again to produce a list of size ten, this time using mutual information between *terms* (instead of term clusters) and features as the objective function. The result[3] is a feature list that effectively partitions the co-clustering vocabulary but carries little syntactic significance. Perplexities using this feature set are reported in the column labeled "Rand 10." As expected, the performance is close to that of the uniform model.

We also applied the models to a hold-out set of documents from the same corpus, all documents in partitions 10 through 24. Because of the small training set, it is virtually certain that new documents contain tokens that have not previously been encountered (truly *novel*, as opposed to *unclustered*, tokens). The occurrence of such tokens effectively rules out the application of models that cannot smooth over the occurrence of zero-frequency terms, such as the *ML* model or any of the *Conjunctive with Term* variants. The table shows that application to novel data does not alter the ordering among the more robust models, however. In fact, it appears to add, fairly consistently, only about a tenth of a point of perplexity to all results.

In an effort to determine how much can be gained from additional training data, we trained a conjunctive model, (feature list size 10) on the first ten partitions of the corpus–effectively doubling the training set. This yielded a perplexity of 3.85, an improvement over the 4.07 realized on the smaller training set. We surmise that training on a truly large corpus should lead to further improvements.

[3] `ti ^c y in h m u l o a.`

Threshold	Augmented	HMM	Together
10	2.11	3.86	3.37
5	2.35	3.69	3.03
2	2.67	3.38	2.90

Table 7: Perplexity of tag prediction measured on terms categorized using lexicon augmentation, on those categorized using a conjunctive model with 10 features, and on both groups together.

5 Lexicon Augmentation

Instead of using an HMM to extend coverage of the category space to low-frequency terms, an alternative is to add them to the lexicon based on their type-level distributional characteristics. This tends to be less expensive than the initial clustering. Not only do these terms have sparser distributions, resulting in quicker computation, but the process of assignment into a static cluster space is also comparatively inexpensive.

Of course, in contrast to the HMM, this procedure cannot account for the phenomenon of polysemy, but we might expect the prevalence of polysemy among the infrequent terms of a coherent corpus to be low. In exchange for reduced flexibility in categorizing a term, this procedure potentially benefits from a more direct use of a term's corpus-wide behavior. In contrast, the HMM makes assignments based on single occurrences (arguably, however, making more effective use of the wider context in which a token is embedded).

5.1 Approach

Given a specified frequency threshold k, we assign each unclustered term with corpus frequence of at least k to a cluster. Each possible cluster assignment is attempted, and the resulting change in the objective function measured. Whichever assignment maximizes the resulting score becomes the presumed class of the term.

5.2 Results

We applied this procedure to all terms having corpus-wide frequencies of at least 10, 5, and 2, and labeled each newly assigned term with its chosen category. The conjunctive model (feature list size 10) was then trained on the first five partitions of the corpus, labeled in this way.

Table 7 presents the results of this experiment. The perplexities listed in the "Together" column were measured on the same set of tokens as in other experiments involving the five-

partition sub-corpus. In comparison with other results, these numbers show a marked improvement. The other two columns help to explain this improvement. The augmentation procedure is surprisingly reliable at low term frequencies, although the reliability degrades as the frequency threshold is lowered. In spite of this, the added constraint improves the performance of the HMM over previous numbers, even when it is relegated to classifying single-occurrence tokens.

Much remains to be explained and explored here. More interesting combinations of the two techniques are possible. Instead of picking a single class for each term, for example, the augmentation procedure might be used to pick a small set of candidate classes. The HMM could then be constrained to choose from among these classes in processing individual tokens. This procedure would exploit the wider context of instances of the low-frequency terms on which the augmentation procedure is weakest, and might mitigate the polysemy problem.

6 Conclusion

Distributional clustering can be used to infer a corpus's syntactic classes and to categorize high-frequency terms. To make this categorization useful, in applications such as information extraction, a reliable means must be devised to categorize low-frequency terms as well.

We have proposed several methods to achieve this. Training a HMM on data partially tagged with the inferred categories, we are able to account for each novel token. Despite contrary results reported in the literature, repeated iterations of Baum-Welch estimation cause all model variants to correlate increasingly with the manual POS tags. By enhancing token emission models with morphological features automatically derived from the data, we are able to improve this agreement further. Finally, we show how a reliable type-level categorization can be combined with the HMM to realize further improvements.

Acknowledgements

This material is based on work funded in whole or in part by the U.S. Government. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the U.S. Government.

References

S. Abney. 1996. Part-of-speech tagging and partial parsing. In Ken Church, Steve Young, and Gerrit Bloothooft, editors, *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, Dordrecht.

T. Brants. 2000. Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*.

E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, and R.L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

A. Clark. 2000. Inducing syntactic categories by context distribution clustering. In *CoNLL 2000*, September.

A. Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proc. 10th EACL Conference (EACL-03)*, April.

D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*.

I. S. Dhillon, S. Mallela, and D. S. Modha. 2003. Information-theoretic co-clustering. Technical Report TR-03-12, Dept. of Computer Science, U. Texas at Austin.

D. Elworthy. 1994. Does Baum-Welch re-estimation help taggers? In *Proc. 4th ACL Conference on Applied Natural Language Processing*.

B. Merialdo. 1994. Tagging text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.

A. Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

H. Schütze. 1995. Distributional part-of-speech tagging. In *Proc. 7th EACL Conference (EACL-95)*, March.

R. Weischedel, M. Meeter, R. Schwartz, L. Ramshaw, and J. Palmucci. 1993. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19:359–382.