Using Morphology and Syntax Together in Unsupervised Learning

Yu Hu and **Irina Matveeva**
Department of Computer Science
The University of Chicago
Chicago IL 60637
yuhu@cs.uchicago.edu
matveeva@uchicago.edu

John Goldsmith
Departments of Linguistics and Computer Science
The University of Chicago
Chicago IL 60637
ja-goldsmith@uchicago.edu

Colin Sprague
Department of Linguistics
The University of Chicago
Chicago IL 60637
sprague@uchicago.edu

Abstract

Unsupervised learning of grammar is a problem that can be important in many areas ranging from text preprocessing for information retrieval and classification to machine translation. We describe an MDL based grammar of a language that contains morphology and lexical categories. We use an unsupervised learner of morphology to bootstrap the acquisition of lexical categories and use these two learning processes iteratively to help and constrain each other. To be able to do so, we need to make our existing morphological analysis less fine grained. We present an algorithm for collapsing morphological classes (signatures) by using syntactic context. Our experiments demonstrate that this collapse preserves the relation between morphology and lexical categories within new signatures, and thereby minimizes the description length of the model.

1 Introduction

Our long term goal is the development of methods which will allow one to produce optimal analyses from arbitrary natural language corpora, where by optimization we understand an MDL (minimum description length; Rissanen, 1989) interpretation of the term: an optimal analysis is one which finds a grammar which simultaneously minimizes grammar length and data compression length. Our specific and primary focus is on *morphology*, and on how knowledge of morphology can be a useful step towards a more complete knowledge of a language's linguistic structure.

Our strategy is based on the following observation: knowing the rightmost suffix of a word is very useful information in inferring (or guessing) a word's part of speech (POS), but due to the ambiguity of many suffixes, it is even better to know both a word's suffix *and* the range of other suffixes that the word's stem appears with elsewhere, i.e., its signature. As we will see below, this conjunction of "better" information is what we call the *signature transform*, and in this paper, we explore how knowledge of signature transform can be merged with knowledge of the context vector to draw conclusions about morphology and syntax.

In the distant future, we would like to be able to use the signature transform in a general process of grammar induction, but that day is not here; we therefore *test* our experiments by seeing how well we are able to predict POS as assigned by an available tagger (TreeTagger; Schmid 1994). In particular, we wish to decrease the uncertainty of a word's POS through the morphological analysis described here. This decrease of uncertainty will enter into our calculation through an *increase* in the probability assigned to our test corpus once the corpus has been augmented with TreeTagger assigned POS tags. But to be clear on our

process: we *analyze* a completely raw text morphologically, and use the POS tags from TreeTagger only to evaluate the signature transforms that we generate.

We assume without argument here that any adequate natural language grammar will contain a lexicon which includes both lexical stems which are specified for morphological properties, such as the specific affixes with which they may occur, and affixes associated with lexical categories. We also explicitly note that many affixes are homophonous: they are pronounced (or written) identically, but have different morphological or syntactic characteristics, such as the English plural *–s* and the verbal 3rd person singular present *–s*.

We focus initially on unsupervised learning of *morphology* for three reasons: first, because we already have a quite successful unsupervised morphological learner; second, the final suffix of a word is typically the strongest single indicator of its syntactic category; and third, analysis of a word into a stem T plus suffix F allows us (given our knowledge that the suffix F is a stronger indicator of category than the stem T) to collapse many distinct stems into a single cover symbol for purposes of analysis, simplifying our task, as we shall see.[1] We eschew the use of linguistic resources with hand-(i.e., human-)assigned morphological information in order for this work to contribute, eventually, to a better theoretical understanding of human language acquisition.

We present in this paper an algorithm that modifies the output of the morphology analyzer by combining redundant signatures. Since we ultimately want to use signatures and signature transforms to learn syntactic categories, we developed an algorithm that uses the syntactic contextual information. We evaluate the changes to the morphological analysis from the standpoint of efficient and adequate representation of lexical categories. This paper presents a test conducted on English, and thus can only be considered a preliminary step in the eventually development of a language-independent tool for grammar induction based on morphology. Nonetheless, the concepts that motivate the process are language-independent, and we are optimistic that similar results would be found in tests based on texts from other languages.

In section 2 we discuss the notion of signature and signature transform, and section 3 present a more explicit formulation of the general problem. In section 4 we present our algorithm for signature collapse. Section 5 describes the experiments we ran to test the signature collapsing algorithm, and section 6 presents and discusses our results.

2 Signatures and signature transforms

We employ the unsupervised learning of morphology developed by Goldsmith (Goldsmith, 2001). Regrettably, some of the discussion below depends rather heavily on material presented there, but we attempt to summarize the major points here.

Two critical terms that we employ in this analysis are *signature* and *signature transform*. A *signature* found in a given corpus is a pair of lists: a *stem*-list and a *suffix*-list (or in the appropriate context, a *prefix*-list). By definition of signature σ, the concatenation of every stem in the stem-list of σ with every suffix in the suffix-list of σ is found in the corpus, and a morphological analysis of a corpus can be viewed as a set of signatures that uniquely analyze each word in the corpus. For example, a corpus of English that includes the words *jump, jumps, jumped, jumping, walk, walks, walked,* and *walking* might include the signature σ_1 whose stem list is { *jump, walk* } and whose suffix list is { Ø, ed, ing , s }. For convenience, we label a signature with the concatenation of its suffixes separated by period '.'. On such an analysis, the word *jump* is analyzed as belonging to the signature *Ø.ed.ing.s*, and it bears the suffix Ø. We say, then, that the signature transform of *jump* is *Ø.ed.ing.s_ Ø*, just as the signature transform of *jumping* is *Ø.ed.ing.s_ing*; in general, the signature transform of a word W, when W is morphologically analyzed as stem T followed by suffix F, associated with signature σ, is defined as σ_F.

[1] See Higgins 2002 for a study similar in some ways; Higgins uses morphology as a bootstrap heuristic in one experimental set-up. This paper is heavily indebted to prior work on unsupervised learning of position categories such as Brown et al 1992, Schütze 1997, Higgins 2002, and others cited there.

In many of the experiments described below, we use a corpus in which all words whose frequency rank is greater than 200 have been replaced by their signature transforms. This move is motivated by the observation that high frequency words in natural languages tend to have syntactic distributions poorly predictable by any feature other than their specific identity, whereas the distribution properties of lower frequency words (which we take to be words whose frequency rank is 200 or below) are better predicted by category membership.

In many cases, there is a natural connection between a signature transform and a lexical category. Our ultimate goal is to exploit this in the larger context of grammar induction. For example, consider the signature *Ø.er.ly*, which occurs with stems such as *strong* and *weak*; in fact, words whose signature transform is *Ø.er.ly_ Ø* are adjectives, those whose signature transform is *Ø.er.ly_er* are comparative adjectives, and those whose signature transform is *Ø.er.ly_ly* are adverbs.

The connection is not perfect, however. Consider the signature *Ø.ed.ing.s* and its four signature transforms. While most words whose σ -transform is *Ø.ed.ing.s_s* are verbs (indeed, 3rd person singular present tense verbs, as in *he walks funny*), many are in fact plural nouns (e.g., *walks* in *He permitted four walks in the eighth inning* is a plural noun). We will refer to this problem as the *signature purity problem*–it is essentially the reflex of the ambiguity of suffixes.

In addition, many 3rd person singular present tense verbs are associated with other signature transforms, such as *Ø.ing.s_s*, *Ø.ed.s_s*, and so forth; we will refer to this as the *signature-collapsing problem*, because all other things being equal, we would like to *collapse* certain signatures, such as *Ø.ed.ing.s* and *Ø.ed.ing*, since a stem that is associated with the latter signature *could* have appeared in the corpus with an -s suffix; removing the *Ø.ed.ing* signature and reassigning its stems to the *Ø.ed.ing.s* signature will in general give us a better linguistic analysis of the corpus, one that can be better used in the problem of lexical category induction. This is the reflex of the familiar data sparsity concern.[2]

Since we ultimately want to use signatures and signature transforms to learn syntactic categories, we base the similarity measure between the signatures on the context.

3 A more abstract statement of the problem

A minimum description length (MDL) analysis is especially appropriate for machine learning of linguistic analysis because simultaneously it puts a premium both on analytical simplicity and on goodness of fit between the model and the data (Rissanen 1989).

We will present first the mathematical statement of the MDL model of the morphology, in (1), following the analysis in Goldsmith (2001), followed by a description of the meaning of the terms of the expressions, and then present the modified version which includes additional terms regarding part of speech (POS) information, in (2) and (3).

(1) Morphology

a. Grammar g =

$$\arg\min_{g \in G} [Length(g) - \log prob(Data \mid g)]$$

b. $Length(g) =$

$$\sum_{t \in T = set\ of\ stems} \left[\log \frac{[W]}{[\sigma(t)]} + \sum_{0 \le i < |t|} \log \frac{1}{freq\, t[i]} \right]$$

$$+ \sum_{f \in F = set\ of\ affixes} \sum_{0 \le i < |f|} \log \frac{1}{freq\, f[i]}$$

$$+ \sum_{\sigma \in \Sigma} \sum_{f \in \sigma} \left[\log \frac{[\sigma]}{[\sigma \cap f]} + \log \frac{[W]}{[f]} \right]$$

[2] The signature-collapsing problem has another side to it as well. An initial morphological analysis of English will typically give rise to a morphological analysis of words such as *move, moves, moved, moving* with a signature whose stems include *mov* and whose affixes are *e.ed.es.ing*. A successful solution to the signature-collapsing problem will collapse *Ø.ed.ing.s* with *e.ed.es.ing*, noting that *Ø ~ e*, *ed ~ed*, *es ~ s*, and *ing ~ ing* in an obvious sense.

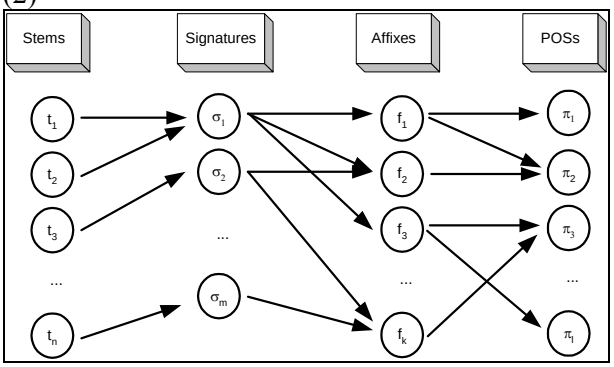
c. $\log prob(Data \mid g) =$

$$\sum_{\substack{w \in Data \\ w = t+f,\sigma}} \begin{bmatrix} \log prob(\sigma) \\ +\log prob(t \mid \sigma) \\ +\log prob(f \mid t, \sigma) \end{bmatrix}$$

Equation (1a) states that our goal is to find the (morphological) grammar that simultaneously minimizes the sum of its own length and the compressed length of the data it analyzes, while (1b) specifies the grammar length (or model length) as the sum of the lengths of the links between the major components of the morphology: the list of letters (or phonemes) comprising the morphemes, the morphemes (stems and affixes), and the signatures. We use square brackets "[.]" to denote the token counts in a corpus containing a given morpheme or word. The first line of (1b) expresses the notion that each stem consists of a pointer to its signature and a list of pointers to the letters that comprise it; $\sigma(t)$ is the signature associated with stem *t,* and we take its probability to be $\frac{[\sigma(t)]}{[W]}$, the empirical count of the words associated with $\sigma(t)$ divided by the total count of words in the data. The second line expresses the idea that the morphology contains a list of affixes, each of which contains a list of pointers to the letters that comprise it. The third line of (1b) expresses the notion that a signature consists of a list of pointers to the component affixes. (1c) expresses the compressed length of each word in the data.[3]

We now consider extending this model to include part of speech labeling, as sketched in (2). The principal innovation in (2) is the addition of part of speech tags; each affix is associated with one or more POS tags. As we have seen, a path from a particular signature σ to a particular affix f constitutes what we have called a particular signature transform σ_f; and we condition the probabilities of the POS tags in the data on the preceding signature transformation. As a result, our final model takes the form in (3).

(2)



(3)

a. Grammar g =

$$\arg\min_{g \in G} \left[Length(g) - \log prob(Data \mid g) \right]$$

b. $Length(g) =$

$$\sum_{t \in T = set\ of\ stems} \left[\log \frac{[W]}{[\sigma(t)]} + \sum_{0 \le i < |t|} \log \frac{1}{freq\, t[i]} \right]$$

$$+ \sum_{f \in F = set\ of\ affixes} \sum_{0 \le i < |f|} \log \frac{1}{freq\, f[i]}$$

$$+ \sum_{\sigma \in \Sigma} \sum_{f \in \sigma} \left[\begin{array}{l} \log \frac{[\sigma]}{[\sigma \cap f]} + \log \frac{[W]}{[f]} + \\ \sum_{\pi \in \Pi} \log \frac{[f \cap \sigma]}{[f \cap \sigma \cap \pi]} \end{array} \right]$$

c. $\log prob(Data \mid g) =$

$$\sum_{\substack{w \in Data \\ w = t+f,\sigma}} \begin{bmatrix} \log prob(\sigma) + \log prob(t \mid \sigma) \\ +\log prob(f \mid t, \sigma) \\ +\log prob(\pi \mid \sigma, f) \end{bmatrix}$$

The differences between the models are found in the added final term in (3b), which specifies the information required to predict, or specify, the part of speech given the signature

[3] We do not sum over all occurrences of a word in the corpus; we count the compressed length of each word type found in the corpus. This decision was made based on the observation that the (compressed length of the) data term grows much faster than the length of the grammar as the corpus gets large, and the loss in ability of the model to predict word frequencies overwhelms any increase in model simplicity when we count word tokens in the data terms. We recognize the departure from the traditional understanding of MDL here, and assume the responsibility to explain this in a future publication.

transform, and the corresponding term in the corpus compression expression (3c).

The model in (3) implicitly assumes that the true POSs are known; in a more complete model, the POSs play a direct role in assigning a higher probability to the corpus (and hence a smaller compressed size to the data). In the context of such a model, an MDL-based learning device searches for the best assignment of POS tags over all possible assignments. Instead of doing that in this paper, we employ the TreeTagger (Schmid, 1994) based tags (see section 5 below), and make the working assumption that optimization of description length over all signature-analyses and POS tags can be approximated by optimization over all signature-analyses, given the POS tags provided by TreeTagger.

4 The collapsing of signatures

We describe in this section our proposed algorithm, using context vectors to collapse signatures together, composed of a sequence of operations, all but the first of which may be familiar to the reader:

Replacement of words by signature-transforms: The input to our algorithm for collapsing signatures is a modified version of the corpus which integrates the (unsupervised) morphological analyses in the following way. First of all, we leave unchanged the 200 most frequent words (word types). Next, we *replace* words belonging to the K most reliable signatures (where K=50 in these experiments) by their associated *signature transforms*, and we in effect *ignore* all other words, by replacing them by a distinguished "dummy" symbol. In the following, we refer to our high frequency words and signature transforms together as *elements*—so an *element* is any member of the transformed corpus other than the "dummy".

Context vectors based on mutual information: By reading through the corpus, we populate both left and right context vectors for each element (=signature-transform and high-frequency word) by observing the elements that occur adjacent to it. The feature indicating the appearance of a particular word on the *left* is always kept distinct from the feature indicating the appearance of the same word on the *right*. The features in a context vector are thus associated with the members of the element vocabulary (and indeed, each member of the element vocabulary occurs as two features: one on the left, one on the right). We assign the value of each feature y of x's context vector as the pointwise mutual information of the corresponding element pair (x, y), defined as $\log \frac{pr(x,y)}{pr(x)pr(y)}$.

Simplifying context vectors with "idf": In addition, because of the high dimensionality of the context vector and the fact that some features are more representative than others, we trim the original context vector. For each context vector, we sort features by their values, and then keep the top N (in general, we set N to 10) by setting these values to 1, and all others to 0. However, in this resulting *simplified context vector,* not all features do equally good jobs of distinguishing syntactical categories. As Wicentowski (2002) does in a similar context, we assign a weight w_{f_i} to each feature f_i in a fashion parallel to inverse document frequency (*idf;* see Sparck Jones 1973), or $\log \frac{\#total\ distinct\ elements}{\#elements\ this\ feature\ appears\ in}$. We view these as the diagonal elements of a matrix M (that is, $m_{i,i} = w_{f_i}$). We then check the similarity between two simplified context vectors by computing the weighted sum of the dot product of them. That is, given two simplified context vectors c and d, their *similarity* is defined as $c^T M d$. If this value is larger than a threshold θ that is set as one parameter, we deem these two context vectors to be similar. Then we determine the similarity between elements by checking whether both left and right simplified context vectors of them are similar (i.e., their weighted dot products exceed a threshold θ). In the experiments we describe below, we explore four settings θ for this threshold: 0.8 (the most "liberal" in allowing greater signature transform collapse, and hence greater signature collapse), 1.0, 1.2, and 1.5.

Calculate signature similarity: To avoid considering many unnecessary pairs of signatures, we narrow the candidates into signature pairs in which the suffixes of one constitute a subset of suffixes of the other, and we set a limit to the permissible difference in the

lengths of the signatures in the collapsed pairs, so that the difference in *number* of affixes cannot exceed 2. For each such pair, if all corresponding *signature transforms* are similar in the sense defined in the preceding paragraph, we deem the two *signatures* to be similar.

Signature graph: Finally, we construct a *signature graph*, in which each signature is represented as a vertex, and an edge is drawn between two signatures iff they are similar, as just defined. In this graph, we find a number of cliques, each of which, we believe, indicates a cluster of signatures which should be collapsed. If a signature is a member of two or more cliques, then it is assigned to the largest clique (i.e., the one containing the largest number of signatures).[4]

5 Experiments

We obtain the morphological analysis of the Brown corpus (Kučera and Francis, 1967) using the Linguistica software (http://linguistica.uchicago.edu), and we use the TreeTagger to assign a Penn TreeBank-style part-of-speech tag to each token in the corpus. We then carry out our experiment using the Brown corpus modified in the way we described above. Thus, for each token of the Brown corpus that our morphology analyzer analyzed, we have the following information: its stem, its signature (i.e., the signature to which the stem is assigned), the suffix which the stem attains in this occurrence of the word (hence, the signature-transform), and the POS tag. For example, the token *polymeric* is analyzed into the stem *polymer* and the suffix *ic*, the stem is assigned to the signature *Ø.ic.s*, and thus this particular token has the signature transform *Ø.ic.s_ic*. Furthermore, it was assigned POS-tag *JJ*, so that we have the following entry: "polymeric JJ Ø.ic.s_ic".

Before performing signature collapsing, we calculate the description length of the morphology and the compressed length of the words that our algorithm analyzes and call it *baseline description length* (DL_0).

Now we apply our signature collapsing algorithm under several different parameter settings for the similarity threshold θ, and calculate the description length DL^{θ} of the resulting morphological and lexical analysis using (3). We know that the smaller the set of signatures, the smaller is the cost of the model. However, a signature collapse that combines signatures with different distributions over the lexical categories will result in a high cost of the data term (3c). The goal was therefore to find a method of collapsing signatures such that the reduction in the model cost will be higher than the increase in the compressed length of the data so that the total cost will decrease.

As noted above, we perform this operation iteratively, and refer to the description length of the ith iteration, using a threshold θ, as $DL^{\theta}_{iter=i}$.

We used random collapsing in our experiments to ensure the expected relationship between appropriate collapses and description length. For each signature collapsing, we created a parallel situation in which the *number* of signatures collapsed is the same, but their *choice* is random. We calculate the description length using this "random" analysis as DL^{θ}_{random}. We predict that this random collapsing will not produce an improvement in the total description length.

[4] Our parameters are by design restrictive, so that we declare only few signatures to be similar, and therefore the cliques that we find in the graph are relatively small. One way to enlarge the size of collapsed signatures would be to loosen the similarity criterion. This, however, introduces too many new edges in the signatures graph, leading in turn to spurious collapses of signatures. We take a different approach, and apply our algorithms iteratively. The idea is that if in the first iteration, two cliques did not have enough edges between their elements to become a single new signature, they may be more strongly connected in the second iteration if many of their elements are sufficiently similar. On the other hand, cliques that were dissimilar in the first iteration remain weakly connected in the second.

6 Results and discussion

Table 1 presents the description length, broken into its component terms (see (3)), for the baseline case and the alternative analyses resulting from our algorithm. The table shows the total description length of the model, as well as the individual terms: the signature term DL(σ), the suffix term DL(F), the lexical categories term, DL(P), total morphology, DL(M), and the compressed length of the data, DL(D). We present results for two iterations for four threshold values (θ=0.8,1.0,1.2,1.5) using our collapsing algorithm.

Table 2 presents DL^{θ}_{random} derived from the random collapsing, in a fashion parallel to Table 1. We show the results for only one iteration of random collapsing, since the first iteration already shows a substantial increase in description length.

Figure 1 and Figure 2 present graphically the total description length from Tables 1 and 2 respectively. The reader will see that all collapsing of signatures leads to a shortening of the description length of the morphology *per se*, and an increase in the compressed length of the data. This is an inevitable formal consequence of the MDL-style model used here. The empirical question that we care about is whether the combined description length increases or decreases, and what we find is that when collapsing the signatures in the way that we propose to do, the combined description length decreases, leading us to conclude that this is, overall, a superior linguistic description of the data. On the other hand, when signatures are collapsed randomly, the combined description length increases. This makes sense; randomly decreasing the formal simplicity of the grammatical description should *not* improve the overall analysis. Only an increase in the formal simplicity of a grammar that is grammatically sensible should have this property. Since our goal is to develop an algorithm that is completely data-driven and can operate in an

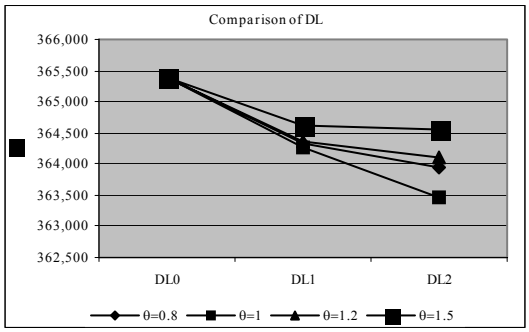


Figure 1 Comparison of DL, 2 iterations and 4 threshold values

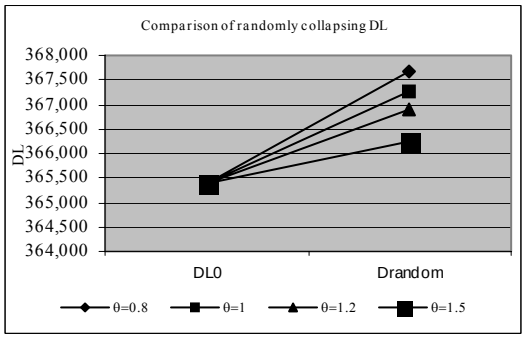


Figure 2 Comparison of DLs with random collapse of signatures (see text)

	DL_0	$DL^{\theta=0.8}_{iter=1}$	$DL^{\theta=0.8}_{iter=2}$	$DL^{\theta=1.0}_{iter=1}$	$DL^{\theta=1.0}_{iter=2}$	$DL^{\theta=1.2}_{iter=1}$	$DL^{\theta=1.2}_{iter=2}$	$DL^{\theta=1.5}_{iter=1}$	$DL^{\theta=1.5}_{iter=2}$
#σ	50	41	35	41	34	44	42	46	45
DL(σ)	47,630	45,343	42,939	45,242	43,046	44,897	44,355	46,172	45,780
DL(F)	160	156	156	153	143	158	147	163	164
DL(P)	2,246	2,087	1,968	2,084	1,934	2,158	2,094	2,209	2,182
DL(M)	50,218	47,768	45,244	47,659	45,304	47,395	46,777	48,724	48,306
DL(D)	315,165	316,562	318,687	316,615	318,172	316,971	317,323	315,910	316,251
Total DL	365,383	364,330	363,931	364,275	363,476	364,367	364,101	364,635	364,558

Table 1. DL and its individual components for baseline and the resulting cases when collapsing signatures using our algorithm.

	DL_0	$DL_{random}^{\theta=0.8}$	$DL_{random}^{\theta=1.0}$	$DL_{random}^{\theta=1.2}$	$DL_{random}^{\theta=1.5}$
#σ	50	41	41	44	46
DL(σ)	47,630	44,892	45,126	45,788	46,780
DL(F)	160	201	198	187	177
DL(P)	2,246	2,193	2,195	2,212	2,223
DL(M)	50,218	47,468	47,700	48,369	49,362
DL(D)	315,165	320,200	319,551	318,537	316,874
Total DL	365,383	367,669	367,252	366,907	366,237

Table 2. DL and its individual components for baseline and the resulting cases when collapsing signatures randomly.

unsupervised fashion, we take this evidence as supporting the appropriateness of our algorithm as a means of collapsing signatures in a grammatically and empirically reasonable way.

We conclude that the collapsing of signatures on the basis of similarity of context vectors of signature transforms (in a space consisting of high frequency words and signature transforms) provides us with a useful and significant step towards solving the signature collapsing problem. In the context of the broader project, we will be able to use signature transforms as a more effective means for projecting lexical categories in an unsupervised way.

As Table 1 shows, we achieve up to 30% decrease in the number of signatures through our proposed collapse. We are currently exploring ways to increase this value through powers of the adjacency matrix of the signature graph.

In other work in progress, we explore the equally important *signature purity* problem in graph theoretic terms: we *split* ambiguous signature transforms into separate categories when we can determine that the edges connecting left-context features and right-context features can be resolved into two sets (corresponding to the distinct categories of the transform) whose left-features have no (or little) overlap and whose right features have no (or little) overlap. We employ the notion of minimum cut of a weighted graph to detect this situation.

References

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

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics,* 27(2): 153-198.

Higgins, Derrick. 2002. *A Multi-modular Approach to Model Selection in Statistical NLP*. University of Chicago Ph.D. thesis.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees.. *International Conference on New Methods in Language Processing*

Kucera, Henry and W. Nelson Francis. 1967. *Computational Analysis of Present-day American English*. Brown University Press.

Rissanen, Jorma. 1989. *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific.

Schütze, Hinrich. 1997. *Ambiguity Resolution in Language Learning*. CSLI Publications. Stanford CA.

Sparck Jones, Karen. 1973. Index term weighting. *Information Storage and Retrieval* 9:619-33.

Wicentowski, Richard. 2002. *Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. Johns Hopkins University Ph.D. thesis.