Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON), Philadelphia, July 2002, pp. 11-20. Association for Computational Linguistics.

Unsupervised Learning of Morphology Using a Novel Directed Search Algorithm: Taking the First Step

Matthew G. Snover and **Gaja E. Jarosz** and **Michael R. Brent**
Department of Computer Science
Washington University
St Louis, MO, USA, 63130-4809
{ms9, gaja, brent}@cs.wustl.edu

Abstract

This paper describes a system for the unsupervised learning of morphological suffixes and stems from word lists. The system is composed of a generative probability model and a novel search algorithm. By examining morphologically rich subsets of an input lexicon, the search identifies highly productive paradigms. Quantitative results are shown by measuring the accuracy of the morphological relations identified. Experiments in English and Polish, as well as comparisons with other recent unsupervised morphology learning algorithms demonstrate the effectiveness of this technique.

1 Introduction

There are numerous languages for which no annotated corpora exist but for which there exists an abundance of unannotated orthographic text. It is extremely time-consuming and expensive to create a corpus annotated for morphological structure by hand. Furthermore, a preliminary, conservative analysis of a language's morphology would be useful in discovering linguistic structure beyond the word level. For instance, morphology may provide information about the syntactic categories to which words belong, knowledge which could be used by parsing algorithms. From a cognitive perspective, it is crucial to determine whether the amount of information found in pure speech is sufficient for discovering the level of morphological structure that children are able to find without any direct supervision. Thus, we believe the task of automatically discovering a conservative estimate of the orthographically-based morphological structure in a language independent manner is a useful one.

Additionally, an initial description of a language's morphology could provide a starting point for supervised morphological models, such as the memory-based algorithm of Van den Bosch and Daelemans (1999), which cannot be used on languages for which annotated data is unavailable.

During the last decade several minimally supervised and unsupervised algorithms that address the problem have been developed. Gaussier (1999) describes an explicitly probabilistic system that is based primarily on spellings. It is an unsupervised algorithm, but requires the tweaking of parameters to tune it to the target language. Brent (1993) and Brent et al. (1995), described Minimum Description Length, (MDL), systems. One approach used only the spellings of the words; another attempted to find the set of suffixes in the language used the syntactic categories from a tagged corpus as well. While both are unsupervised, the latter is not knowledge free and requires data that is tagged for part of speech, making it less suitable for analyzing under examined languages.

A similar MDL approach is described by Goldsmith (2001). It is ideal in being both knowledge free and unsupervised. The difficulty lies in Goldsmith's liberal definition of morphology which he uses to evaluate with; a more conservative approach would seem to be a better hypothesis to bootstrap from.

We previously, Snover and Brent (2001), presented a very conservative unsupervised system,

which uses a generative probability model and a hill climbing search. No quantitative studies had been conducted on it, and it appears that the hill-climbing search used limits that system's usefulness. We have developed a system based on a novel search and an extension of the previous probability model of Snover and Brent.

The use of probabilistic models is equivalent to minimum description length models. Searching for the most probable hypothesis is just as compelling as searching for the smallest hypothesis and a model formulated in one framework can, through some mathematical manipulation, be reformulated into the other framework. By taking the negative log of a probability distribution, one can find the number of bits required to encode a value according to that distribution. Our system does not use the minimum description length principle but could easily be reformulated to do so.

Our goal in designing this system, is to be able to detect the final stem and suffix break of each word given a list of the most common words in a language. We do not distinguish between derivational and inflectional suffixation or between the notion of a stem and a base. Our probability model differs slightly from that of Snover and Brent (2001), but the main difference is in the search technique. We find and analyze subsets of the lexicon to find good solutions for a small set of words. We then combine these subhypotheses to form a morphological analysis of the entire input lexicon. We do not attempt to learn prefixes, infixes, or other more complex morphological systems, such as template-based morphology: we are attempting to discover the component of many morphological systems that is strictly concatenative. Finally, our model does not currently have a mechanism to deal with multiple interpretations of a word, or to deal with morphological ambiguity.

2 Probability Model

This section introduces a prior probability distribution over the space of all hypotheses, where a hypothesis is a set of words, each with morphological split separating the stem and suffix. The distribution is based on a seven-model model for the generation of hypothesis, which is heavily based upon the probability model presented in Snover and Brent (2001), with steps 1-3 of the generative procedure being the same. The two models diverge at step 4 with the pairing of stems and suffixes. Whereas the previous model paired individual stems with suffixes, our new model uses the abstract structure of paradigms. A paradigm is a set of suffixes and the stems that attach to those suffixes and no others. Each stem is in exactly one paradigm, and each paradigm has at least one stem. This is an important improvement to the model as it takes into account the patterns in which stems and suffixes attach.

The seven steps are presented below, along with their probability distributions and a running example of how a hypothesis could be generated by this process. By taking the product over the distributions from all of the steps of the generative process, one can calculate the prior probability for any given hypothesis. What is described in this section is a mathematical model and not an algorithm intended to be run.

1. Choose the number of stems, M, according to the distribution:

$$\Pr(M) = \frac{6}{\pi^2}\left(\frac{1}{M}\right)^2 \quad (1)$$

The $6/\pi^2$ term normalizes the inverse-squared distribution on the positive integers. The number of suffixes, X is chosen according to the same probability distribution. The symbols M for steMs and X for suffiXes are used throughout this paper.
Example: M = 5. X = 3.

2. For each stem i, choose its length in letters L_i^m, according to the inverse squared distribution. Assuming that the lengths are chosen independently and multiplying together their probabilities we have:

$$\Pr(L^m \mid M) = \left(\frac{6}{\pi^2}\right)^M \prod_{i=1}^{M}\left(\frac{1}{L_i^m}\right)^2 \quad (2)$$

The distribution for the lengths of the suffixes, L^x, is similar to (2), differing only in that suffixes of length 0 are allowed, by offsetting the length by one.
Example: L^m = 4, 4, 4, 3, 3. L^x = 2, 0, 1.

3. Let Σ be the alphabet, and let $\{p_1 \ldots p_{|\Sigma|}\}$ be a probability distribution on Σ. For each i from 1 to M, generate stem i by choosing L_i^m letters at random, according to the probabilities $\{p_1 \ldots p_{|\Sigma|}\}$. Call the resulting stem set STEM. The suffix set SUFF is generated in the same manner. The probability of any character, l, being chosen is obtained from a maximum likelihood estimate:

$$\hat{p}_l = \frac{c_l}{S}$$

where c_l is the count of l among all the hypothesized stems and suffixes and $S = \sum_l c_l$.

The joint probability of the hypothesized stem and suffix sets is defined by the distribution:

$$\Pr(\text{STEM}, \text{SUFF} \mid M, L^m, X, L^x) = M!X! \prod_{l \in \Sigma} \left(\frac{c_l}{S}\right)^{c_l} \quad (3)$$

The factorial terms reflect the fact that the stems and suffixes could be generated in any order.
Example: STEM = {walk, look, door, far, cat}. SUFF = {ed, ϵ, s}.

4. We now choose the number of paradigms, P, which can range from 1 to M since each stem is in exactly one paradigm, and each paradigm has at least one stem. We pick P according to the following uniform distribution:

$$\Pr(P \mid M) = M^{-1} \quad (4)$$

Example: P = 3.

5. We choose the number of suffixes in the paradigms, D, according to a uniform distribution. The distribution for picking D_i, suffixes for paradigm i is:

$$\Pr(D_i \mid XP) = \frac{1}{X}$$

The joint probability over all paradigms, D is therefore:

$$\Pr(D \mid XP) = \prod_{i=1}^{P} X^{-1} = \left(\frac{1}{X}\right)^P \quad (5)$$

Example: D = {2, 1, 2}.

6. For each paradigm i, choose the set of D_i suffixes, PARA_i^x that the paradigm will represent. The number of subsets of a given size is finite so we can again use the uniform distribution. This implies that the probability of each individual subset of size D_i, is the inverse of the total number of such subsets. Assuming that the choices for each paradigm are independent:

$$\Pr(\text{PARA}^x \mid XPD) = \prod_{i=1}^{P} \binom{X}{D_i}^{-1} = \binom{X}{D_i}^{-P} \quad (6)$$

Example: $PARA_1^x$ = {ϵ, s, ed}. $PARA_2^x$ = {ϵ}. $PARA_3^x$ = {ϵ, s}.

7. For each stem choose the paradigm that the stem will belong in, according to a distribution that favors paradigms with more stems. The probability of choosing a paradigm i, for a stem is calculated using a maximum likelihood estimate:

$$\frac{|\text{PARA}_i^m|}{M}$$

where PARA_i^m is the set of stems in paradigm i. Assuming that all these choices are made independently yields the following:

$$\Pr(\text{PARA}^m \mid MXP) = \prod_{i=1}^{P} \left(\frac{|\text{PARA}_i^m|}{M}\right)^{|\text{PARA}_i^m|} \quad (7)$$

Example: $PARA_1^m$ = {walk, look}. $PARA_2^m$ = {far}. $PARA_3^m$ = {door, cat}.

Combining the results of stages 6 and 7, one can see that the running example would yield the hypothesis consisting of the set of words with suffix breaks, {walk+ϵ, walk+s, walk+ed, look+ϵ, look+s, look+ed, far+ϵ, door+ϵ, door+s, cat+ϵ, cat+s}. Removing the breaks in the words results in the set of input words. To find the probability for this hypothesis just take of the product of the probabilities from equations (1) to (7).

The inverse squared distribution is used in steps 1 and 2 to simulate a relatively uniform probability

distribution over the positive integers, that slightly favors smaller numbers. Experiments substituting the universal prior for integers, developed by Rissanen (1989), for the inverse squared distribution, have shown that the model is not sensitive to the exact distribution used for these steps. Only slight differences in the some of the final hypotheses were found, and it was unclear which of the methods produced superior results. The reason for the lack of effect is that the two distributions are not too dissimilar and steps 1 and 2 are not main contributors to the probability mass of our model. Thus, for the sake of computational simplicity we use the inverse squared distribution for these steps.

Using this generative model, we can assign a probability to any hypothesis. Typically one wishes to know the probability of the hypothesis given the data, however in our case such a distribution is not required. Equation (8) shows how the probability of the hypothesis given the data could be derived from Bayes law.

$$\Pr(\text{Hyp} \mid \text{Data}) = \frac{\Pr(\text{Hyp})\Pr(\text{Data} \mid \text{Hyp})}{\Pr(\text{Data})} \quad (8)$$

Our search only considers hypotheses consistent with the data. The probability of the data given the hypothesis, $\Pr(\text{Data}|\text{Hyp})$, is always 1, since if you remove the breaks from any hypothesis, the input data is produced. This would not be the case if our search considered inconsistent hypotheses. The prior probability of the data is unknown, but is constant over all hypotheses, thus the probability of the hypothesis given the data reduces to $\Pr(\text{Hyp})/c$. The prior probability of the hypothesis is given by the above generative process and, among all consistent hypotheses, the one with the greatest prior probability also has the greatest posterior probability.

3 Search

This section details a novel search algorithm which is used to find the most likely segmentation of the all the words in the input lexicon, L. The input lexicon is a list of words extracted from a corpus. The output of the search is a segmentation of each of the input words into a stem and suffix. The algorithm does not directly attempt to find the most probable hypothesis consistent with the input, but finds a highly probable consistent hypothesis.

The directed search is accomplished in two steps. First sub-hypotheses, each of which is a hypothesis about a subset of the lexicon, are examined and ranked. The N best sub-hypotheses are then incrementally combined until a single sub-hypothesis remains. The remainder of the input lexicon is added to this sub-hypothesis at which point it becomes the final hypothesis.

3.1 Ranking Sub-Hypotheses

We define the set of possible suffixes to be the set of terminal substrings, including the empty string ϵ, of the words in L. Each subset of the possible suffixes has a corresponding sub-hypothesis. The sub-hypothesis, h, corresponding to a set of suffixes SUFF_h, has the set of stems STEMS_h. For each stem m and suffix x, in h, the word $m + x$ must be a word in the input lexicon. STEM_h is the maximal sized set of stems that meets this requirement. The sub-hypothesis, h, is thus the hypothesis over the set of words formed by all pairings of the stems in STEM_h and the suffixes in SUFF_h with the corresponding morphological breaks. One can think of each sub-hypothesis as initially corresponding to a maximally filled paradigm. We only consider sub-hypotheses which have at least two stems and two suffixes.

For each sub-hypothesis, h, there is a corresponding counter hypothesis, $\bar{h}$, which has the same set of words as h, but in which all the words are hypothesized to consist of the word as the stem and ϵ as the suffix.

We can now assign a score to each sub-hypothesis as follows: $\text{score}(h) = \Pr(h)/\Pr(\bar{h})$. This reflects how much more probable h is for those words, than the counter or null hypothesis.

The number of possible sub-hypotheses grows considerably as the number of words increases, causing the examination of all possible sub-hypotheses at very large lexicon sizes to become unreasonable. However since we are only concerned with finding the N best sub-hypotheses, we do not actually need to examine every sub-hypothesis. A variety of different search algorithms can be used to find high scoring sub-hypotheses without significant risk of missing any of the N best sub-hypothesis.

One can view all sub-hypotheses as nodes in a directed graph. Each node, n_i, is connected to another node, n_j if and only if n_j represents a superset of the suffixes that n_i represents, which is exactly one suffix greater in size than the set that n_i represents. By beginning at the node representing no suffixes, one can apply standard graph search techniques, such as a beam search or a best first search to find the N best scoring nodes without visiting all nodes. While one cannot guarantee that such approaches perform exactly the same as examining all sub-hypotheses, initial experiments using a beam search with a beam size equal to N, with a N of 100, show that the N best sub-hypotheses are found with a significant decrease in the number of nodes visited. The experiments presented in this paper do not use these pruning methods.

3.2 Combining Sub-Hypotheses

The highest N scoring sub-hypotheses are incrementally combined in order to create a hypothesis over the complete set of input words. The selection of N should not vary from language to language and is simply a way of limiting the computational complexity of the algorithm. Changing the value of N does not dramatically alter the results of the algorithm, though higher values of N give slightly better results. We let N be 100 in the experiments reported here.

Let S be the set of the N highest scoring sub-hypotheses. We remove from S the sub-hypothesis, s', which has the highest score. The words in s' are now added to each of the remaining sub-hypotheses in S, and their counter hypotheses. Every sub-hypothesis, s, and its counter, $\bar{s}$, in S are modified such that they now contain all the words from s' with the morphological breaks those words had in s'. If a word was already in s and $\bar{s}$ and it is also in s' then it now has the morphological break from s', overriding whatever break was previously attributed to the word.

All of the sub-hypotheses now need to be rescored, as the words in them will likely have changed. If, after rescoring, none of the sub-hypotheses have scores greater than one, then we use s' as our final hypothesis. Otherwise we repeat the process of selecting s' and adding it in. We continue doing this until all sub-hypotheses have scores of one or less or there are no sub-hypotheses left.

The final sub-hypothesis, s', is now converted into a full hypothesis over all the words. All words in L, that are not in s' are added to s' with ϵ as their suffix. This results in a hypothesis over all the words in L.

4 Experiment and Evaluation

4.1 Experiment

We tested three unsupervised morphology learning systems on various sized word lists from English and Polish corpora. For English we used set A of the Hansard corpus, which is a parallel English and French corpus of proceedings of the Canadian Parliament. We were unable to find a standard corpus for Polish and developed one from online sources. The sources for the Polish corpus were older texts and thus our results correspond to a slightly antiquated form of the language. We compared our directed search system, which consists of the probability model described in Section 2 and the directed search described in Section 3 with Goldsmith's MDL algorithm, otherwise known as Linguistica[1] and our previous system (2001), which shall henceforth be referred to as the Hill Climbing Search system. The results were then evaluated by measuring the accuracy of the stem relations identified.

We extracted input lexicons from each corpus, excluding words containing non-alphabetic characters. The 100 most common words in each corpus were also excluded, since these words tend to be function words and are not very informative for morphology. Including the 100 most common words does not significantly alter the results presented. The systems were run on the 500, 1,000, 2,000, 4,000, and 8,000 most common remaining words. The experiments in English were also conducted on the 16,000 most common words from the Hansard corpus.

4.2 Evaluation Metrics

Ideally, we would like to be able to specify the correct morphological break for each of the words in the input, however morphology is laced with ambiguity,

[1] A demo version available on the web, http://humanities.uchicago.edu/faculty/goldsmith/, was used for these experiments. Word-list corpus mode and the method A suffix detection were used. All other parameters were left at their default values.

and we believe this to be an inappropriate method for this task. For example it is unclear where the break in the word, "location" should be placed. It seems that the stem "locate" is combined with the suffix "tion", but in terms of simple concatenation it is unclear if the break should be placed before or after the "t". When "locate" is combined with the suffix "s", simple concatenation seems to work fine, though a different stem is found from "location" and the suffix "es" could be argued for. One solution is to develop an evaluation technique which incorporates the adjustment or spelling change rules, such as the one that deletes the "e" in "locate" when combining with "tion".

None of the systems being evaluated attempt to learn adjustment rules, and thus it would be difficult to analyze them using such a measure. In an attempt to solve this problem we have developed a new measure of performance, which does not specify the exact morphological split of a word. We measure the accuracy of the stems predicted by examining whether two words which are morphologically related are predicted as having the same stem. The accuracy of the stems predicted is analyzed by examining whether pairs of words are morphologically related by having the same immediate stem. The actual break point for the stems is not evaluated, only whether the words are predicted as having the same stem. We are working on a similar measure for suffix identification, which measures whether pairs that have the same suffix are found as having the same suffix, regardless of the actual form of the suffix predicted.

4.2.1 Stem Relation

Two words are related if they share the same immediate stem. For example the words "building", "build", and "builds" are related since they all have "build" as a stem, just as "building" and "buildings" are related as they both have "building" as a stem. The two words, "buildings" and "build" are not directly related since the former has "building" as a stem, while "build" is its own stem. Irregular forms of words are also considered to be related even though such relations would be very difficult to detect with a simple concatenation model.

We say that a morphological analyzer predicts two words as being related if it attributes the same stem to both words, regardless of what that stem actually is. If an analyzer made a mistake and said both "build' and "building" had the stem "bu", we would still give credit to it for finding that the two are related, though this analysis would be penalized by the suffix identification measure. The stem relation precision measures how many of the relations predicted by the system were correct, while the recall measures how many of the relations present in the data were found. Stem relation fscore is an unbiased combination of precision and recall that favors equal scores.

Lexicon Size	English	Polish
500	99	348
1,000	321	891
2,000	1,012	2,062
4,000	2,749	4,352
8,000	6,762	9,407
16,000	15,093	-

Table 1: Correct Number of Stem Relations

The correct number of stem relations for each lexicon size in English and Polish are shown in Table 1. Because Polish has a richer morphology than English, the number of relations in Polish is significantly higher than the number of relations in English at every lexicon size.

4.3 Results

The results from the experiments are shown in Figures 1- 3. All graphs are shown use a log scale for the corpus size. Due to software difficulties we were unable to get Linguistica to run on 500, 1000, and 2000 words in English. The software ran without difficulties on the larger English datasets and on the Polish data.

Figure 1 shows the number of different suffixes predicted by each of the algorithms in both English and Polish. The Hill Climbing Search system found a very small number of suffixes in the English data and was unable to find any suffixes, other than ϵ, in the Polish data. Our directed search algorithm found a relatively constant number of suffixes across lexicon sizes and Linguistica found an increasingly large number of suffixes, predicting over 700 different suffixes in the 16,000 word English lexicon.

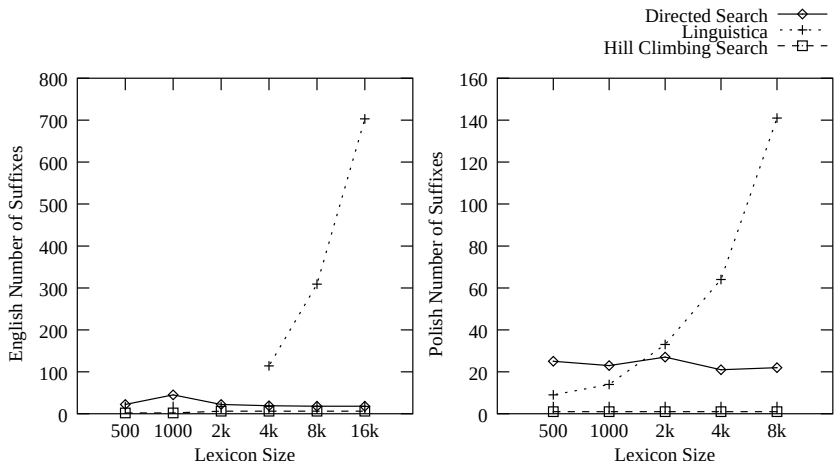

Figure 1: Number of Suffixes Predicted

Figure 2 shows the precision, recall and fscore using the stem relation metric. Figure 3 shows the performance of the algorithms on the Polish input lexicon. The Hill Climbing Search system was unable to learn any morphology on the Polish data sets, and thus has zero precision and recall. The Directed Search maintains a very high precision across lexicon sizes in both languages, whereas the precision of Linguistica decreases considerably at larger lexicon sizes. However Linguistica shows an increasing recall as the lexicon size increases, with our Directed Search having a decreasing recall as lexicon size increases, though the recall of Linguistica in Polish is consistently lower than the Directed Search system's recall. The fscores for the Directed Search and Linguistica in English are very close, and the Directed Search appears to clearly outperform Linguistica in Polish.

Suffixes	Stems
-a -e -ego -ej -ie -o -y	dziwn
ϵ -a -ami -y -ę	chmur siekier
ϵ -cie -li -m -ć	gada odda sprzeda

Table 2: Sample Paradigms in Polish

Table 2 shows several of the larger paradigms found by our directed search algorithm when run on 8000 words of Polish. The first paradigm shown is for the single adjective stem meaning "strange" with numerous inflections for gender, number and case, as well as one derivational suffix, "-ie" which changes it into an adverb, "strangely". The second paradigm is for the nouns, "cloud" and "ax", with various case inflections and the third paradigm paradigm contains the verbs, "talk", "return", and "sell". All suffixes in the third paradigm are inflectional indicating tense and agreement.

As an additional note, Linguistica was dramatically faster than either our Directed Search or the Hill Climbing Search system. Both systems are development oriented software and not as optimized for efficient runtime as Linguistica appears to be.

Of the three systems, the Hill Climbing Search system has poorest performance. The poor performance of the Hill Climbing Search system in Polish is due to a quirk in its search algorithm, which prevents it from hypothesizing stems that are not themselves words. This is not a bug in the software, but a property of the algorithm used. In English this is not a significant difficulty as most stems are also words, but this is almost never the case in Polish, where almost all stems require some suffix.

The differences between the performance of Linguistica and our Directed Search system can most easily be seen in the number of suffixes predicted by each algorithm. The number of suffixes predicted by Linguistica grows linearly with the number of words, in general causing his algorithm to get much higher recall at the expense of precision. The Directed Search algorithm maintains a fairly constant number of suffixes, causing it to generally have higher precision at the expense of recall. This is consistent with our goals to create a conservative sys-

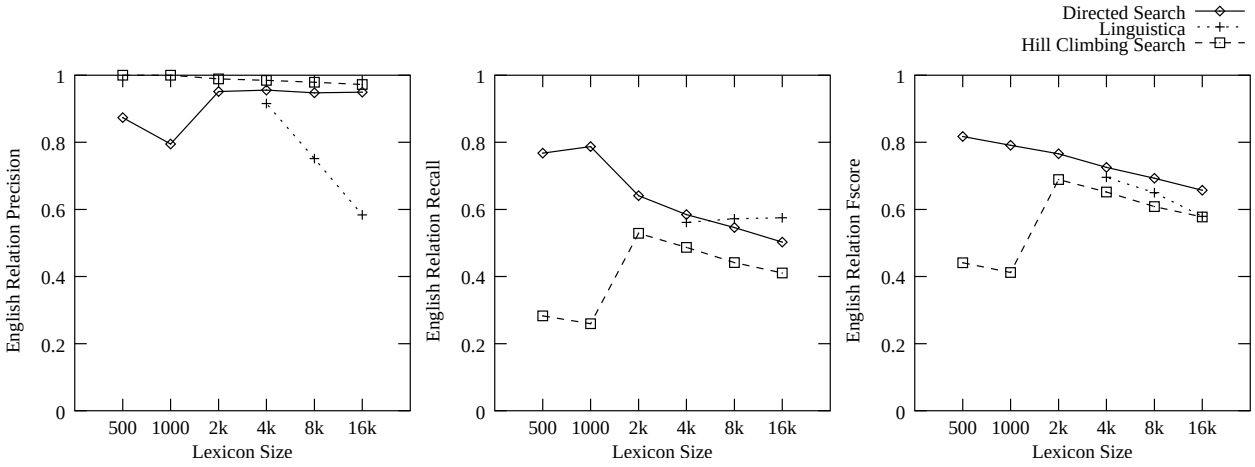

Figure 2: English Results

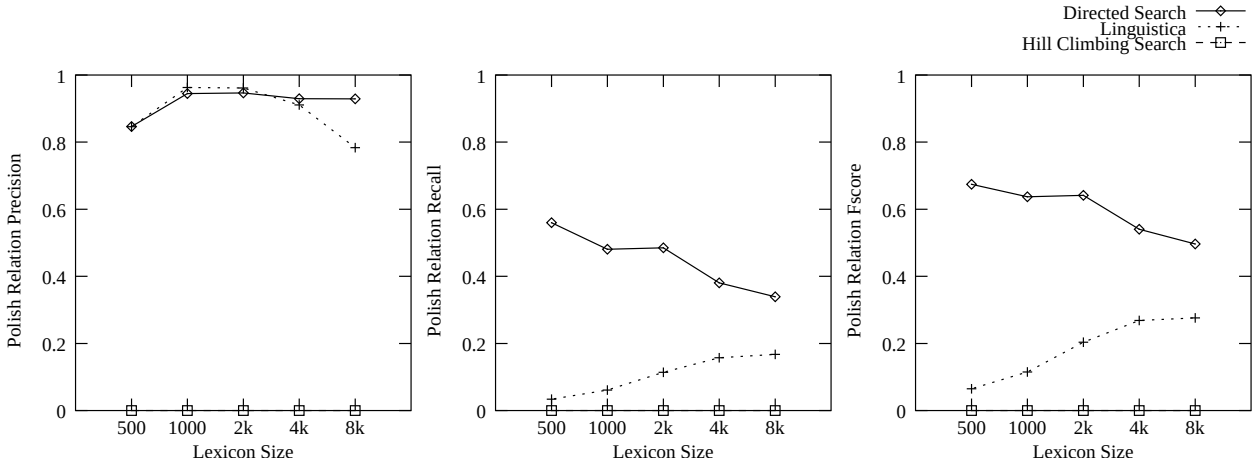

Figure 3: Polish Results

tem for morphological analysis, where the number of false positives is minimized.

Most of Linguistica's errors in English resulted from the algorithm mistaking word compounding, such as "breakwater", for suffixation, namely treating "water" as a productive suffix. While we do think that the word compounding detected by Linguistica is useful, such compounding of words is not generally considered suffixation, and thus should be penalized against.

The Polish language presents special difficulties for both Linguistica and our Directed Search system, due to the highly complex nature of its morphology. There are far fewer spelling change rules and a much higher frequency of suffixes in Polish than in English. In addition phonology plays a much stronger role in Polish morphology, causing alterations in stems, which are difficult to detect using a concatenative framework.

5 Discussion

The superior fscore of our Directed Search system over the Linguistica system has several possible factors which we are currently investigating. It must be noted that Linguistica is designed to leverage off of word frequency in a corpus, and its performance may be enhanced if given a corpus of words, rather than just a lexicon. Similar distributions are used both in the Linguistica model and our Directed Search Model. Rissanen's universal prior for integers is frequently used in Linguistica whereas the inverse squared distribution is used in our model.

Experiments substituting the inverse squared distribution with the universal prior have shown no significant empirical difference in performance. We are currently working on a more detailed comparison of the two systems.

The results obtained from Directed Search algorithm can be significantly improved by incorporating the hill climbing search detailed in Snover and Brent (2001). The hill climbing search attempts to move stems from one paradigm to similar paradigms to increase the probability of the hypothesis. Experiments where the hypothesis outputted by the Directed Search system is used as the starting hypothesis for the hill climbing search, using the probability model detailed in this paper, show an increase in performance, most notably in recall and fscore, over using the Directed Search in isolation.

Many of the stem relations predicted by the Directed Search algorithm, result from postulating stem and suffix breaks in words that are actually morphologically simple. This occurs when the endings of these words resemble other, correct, suffixes. In an attempt to deal with this problem we have investigated incorporating semantic information into the probability model since morphologically related words also tend to be semantically related. A successful implementation of such information should eliminate errors such as "capable" breaking down as "cap"+"able" since "capable" is not semantically related to "cape" or "cap".

Using latent semantic analysis, Schone and Jurafsky (2000) have previously demonstrated the success of using semantic information in morphological analysis. Preliminary results on our datasets using a similar technique, co-occurrence data, which represents each word as a vector of frequencies of co-occurrence with other words, indicates that much semantic, as well as morphological, information can be extracted. When the cosine measure of distance is used in comparing pairs of words in the corpus, the highest scoring pairs are for the most part morphologically or semantically related. We are currently working on correctly incorporating this information into the probability model.

The Directed Search algorithm does not currently handle multiple suffixation or any prefixation; however, some ideas for future work involve extending the model to capture these processes. While such an extension would be a significant one, it would not change the fundamental nature of the algorithm. Furthermore, the output of the present system is potentially useful in discovering spelling change rules, which could then be bootstrapped to aid in discovering further morphological structure. Yarowsky and Wicentowski (2000) have developed a system that learns such rules given a preliminary morphological hypothesis and part of speech tags.

While the experiments reported here are based on an input lexicon of orthographic representations, there is no reason why the Directed Search algorithm could not be applied to phonetically transcribed data. In fact, especially in the case of the English language, where the orthography is particularly inconsistent with the phonology, our algorithm might be expected to perform better at discovering the internal structure of phonologically transcribed words. Furthermore, phonetically transcribed data would eliminate the problems introduced by the lack of one-to-one correspondence of letters to phonemes. Namely, the algorithm would not mistakenly treat sibilants, such as the /ch/ sound in "chat" as two separate units, although these phonemes are often represented orthographically by a two letter sequence. A model of morphology incorporating phonological information such as phonological features could capture morphological phenomena that bridge the morphology-phonology boundary, such as allomorphy, or the existence of multiple variants of morphemes. Simply running the algorithm on phonetic data might not improve performance though, as same structures which were more straight forward in the orthographic data might be more complex in the phonetic representation. Finally, for those interested in the question of whether the language learning environment provides children with enough information to discover morphology with no prior knowledge, an analysis of phonological not orthographic data would be necessary.

The goal of the Directed Search model was to produce a preliminary description, with very low false positives, of the final suffixation, both inflectional and derivational, in a language independent manner. The Directed Search algorithm performed better for the most part with respect to Fscore than Linguistica,

but more importantly, the precision of Linguistica does not approach the precision of our algorithm, particularly on the larger corpus sizes. On the other hand, we feel the Directed Search algorithm has attained the goal of producing an initial estimate of suffixation that could aid other models in discovering higher level structure.

References

Michael R. Brent, Sreerama K. Murthy, and Andrew Lundberg. 1995. Discovering morphemic suffixes: A case study in minimum description length induction. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Laudersdale, FL.

Michael R. Brent. 1993. Minimal generative models: A middle ground between neurons and triggers. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*, pages 28–36, Hillsdale, NJ. Erlbaum.

Éric. Gaussier. 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *ACL '99 Workshop Proceedings: Unsupervised Learning in Natural Language Processing*. ACL.

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

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

Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the Conference on Computational Natural Language Learning*. Conference on Computational Natural Language Learning.

Matthew G. Snover and Michael R. Brent. 2001. A Bayesian Model for Morpheme and Paradigm Identification. In *Proceedings of the 39th Annual Meeting of the ACL*, pages 482–490. Association for Computational Linguistics.

Antal Van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. In *Proc. of the 37th Annual Meeting of the ACL*. ACL.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of ACL-2000*, pages 207–216. ACL.