# ON UNSUPERVISED GRAMMAR INDUCTION FROM UNTAGGED CORPORA

DAMIR ĆAVAR, JOSHUA HERRING, TOSHIKAZU IKUTA, PAUL RODRIGUES, GIANCARLO SCHREMENTI
*Indiana University, Bloomington*

ABSTRACT

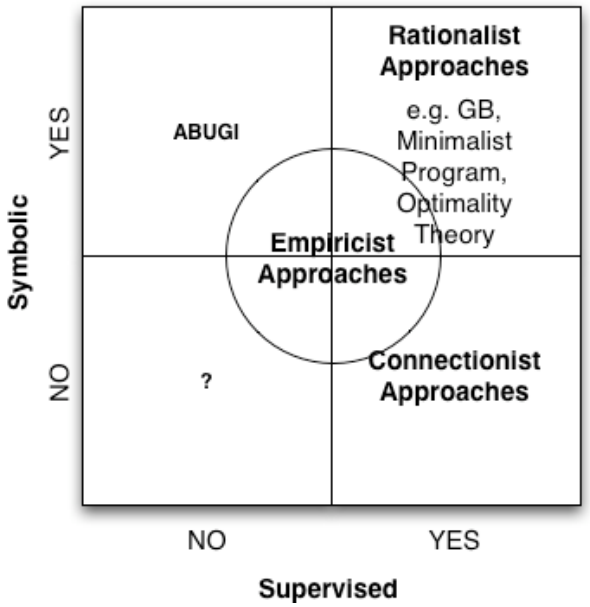
In this paper we describe the theoretical motivation and the implementation of an unsupervised morphology induction algorithm. We show that we can induce morphologies with very high precision, using simple unsupervised statistical algorithms. The results are not only relevant from a theoretical point of view, they also have various potential applications in natural language processing.

## 1. Introduction

Theoretical approaches to the language faculty, and language acquisition in particular, differ with respect to two fundamental assumptions:

a. to what extent they use symbolic mechanisms and
b. to what extent they are supervised

Symbolic approaches are those for which the language faculty is part of a cognitive system that is based on symbols and symbol manipulation operations. In such systems linguistic units are categorical types, e.g. lexical items of a certain type, represented by a symbol for the category. Grammatical rules and principles specify legal operations on these categories. The alternative, typically connectionist, approaches do not assume any symbols but rather understand every linguistic event to be an activation pattern of neurons and connections between neurons in a network. In connectionist approaches there is typically no grammar since there is no symbolic level and thus no possibility to express linguistic dependencies in terms of category manipulation.
Supervised models of the language faculty are those that have access to or are otherwise built upon detailed information about the properties of natural language. They are deductive in nature, using an innate set of rules or principles. Unsupervised models, in contrast, rely on general learning strategies that are not language-specific, typically using statistical induction. Both generative and connectionist approaches are typically supervised. It should be noted that although connectionist models refer to themselves as statistical models, they are not necessarily inductive. Neither should one assume that there are no statistical symbolic approaches that model language faculty and the language acquisition process. Along these lines, current approaches to human language acquisition can be classified into four broad groups as follows:




Rationalist approaches such as the Minimalist Program or Optimality Theory are not only completely supervised, i.e. all rules, constraints, and elementary linguistic categories are assumed to be innate, they are also purely symbolic models of the language faculty. Connectionist approaches are supervised in the sense that the neural networks are specialized on particular tasks, designed to learn from a specific input signal. Their architectural properties are hard-wired and biased towards a very specific task. Further, postulated systems are allowed to access knowledge of how well they are able to reproduce the input and update their performance on the basis of that. Typically no symbolic level is assumed. In between one will find empiricist approaches that vary with respect to the type and amount of innate language specific knowledge, or to the amount of use of discrete categories and symbol manipulation mechanisms.

To our knowledge, clearly non-symbolic and non-supervised approaches do not exist.

The approach we present in the following is located in the top left corner of the graph and is an unsupervised symbolic and statistical approach to language acquisition, or more generally, an empiricist approach. Our main research hypotheses are that certain grammatical properties can be induced from raw natural language input. The questions we try to answer are the following: What kind of learning algorithms can induce what kind of grammatical knowledge given what kind of input in terms of quality and quantity? How can this induced knowledge be used to induce further knowledge of language? And in particular, if we have to assume supervision for certain grammar components, i.e. some innate principles or constraints, what kind of properties do they have to have? These questions, we believe, can be partially answered given the growing amount of natural language corpora and insights in machine learning algorithms and their role in cognitive modeling of language acquisition and processing. With respect to questions of whether symbols are used in the cognitive component of the language faculty, we would like to point out that this issue is most probably completely irrelevant, since potentially a neural network might be used to simulate symbolic processing.

In the following, we focus on the induction of the morphological level of grammar. Morphological properties of lexical items are relevant for clustering of lexical items, i.e. for their categorization. Lexical categorization is the prerequisite for grammar induction on the morphological, as well as the syntactic level. Morphological properties are cues for higher-level grammatical properties.

## 2. Cue-based Learning

We assume that every learning procedure relies on cues about the properties of the system to be learned. Cues for language acquisition are properties of the language stimulus that correlate with specific properties of the grammar, where grammar here is not understood to be the set of possible utterances of a certain language, but the abstract set of rules and principles that govern the generation process of natural language utterances. Different approaches, as mentioned above, have different notions of cues. For purely empiricist approaches, cues are observable properties of language units. Such properties could be inherent to the individual linguistic unit, i.e. its duration, length, relative frequency. They could be relational properties that affect the unit in relation to others, i.e. collocations, relative distance, position in the utterance etc. We assume here that language properties, from our empiricist perspective, are determined along these lines, i.e. as inherent and distributional properties of its elementary units.

As suggested in Ćavar and Elghamry (2004), we assume here that language is acquired incrementally. Acquisition proceeds in stages, with knowledge acquired in earlier, lower-level stages, providing the basis for conclusions (either deductions or inductions) at later, higher-level stages. The language faculty thus emerges piecemeal, increasing in complexity and expressive power as each new bit of learning is added.

This process is crucially understood as an interactive one: innate knowledge and prior conclusions are tested on and formed against the background of linguistic experience. Salient examples form linguistic "cues" which serve to expand on or otherwise change the body of knowledge being acquired.

Our view on cue-based learning is that in the initial stage of learning, a set of cues K identifies the setting of a particular parameter P(1) given some input. These cues and parameter(s) are then applied to new input to induce new sets of cues K' and parameters P(2), P(3), etc. This process continues recursively until all relevant parameters have been set, at which point the grammar settles on a fixed state.

### 2.1. Cue Identification

It is the problem of cue identification to which we primarily address ourselves in this paper. Specifically, how do learners identify some set of cues and use it in the induction process? This is a question to which standard generative theories do not, as yet, provide an adequate answer. Such approaches necessarily assume the prior existence of an innate and specific knowledge of all possible particular language stimuli for the identification of salient (i.e. parameter-setting) examples, cf. Lightfoot (1999). But such a theory is unsatisfying: if such knowledge exists in this form in the brain then the purpose of the acquisition phase for grammatical structure is unclear.

Here we propose a possible explanation for these seeming contradictions. Specifically, while we agree that certain innate mechanisms are necessary to account for language acquisition, we disagree that these forms should themselves necessarily be linguistic in nature. While the human linguistic ability may indeed be cognitively "exceptional" in its functioning and structure, there is no reason to assume that *initial* parameter settings need be based on linguistic knowledge. We purport to show that ordinary statistical abilities of that kind common to e.g. perception in general are sufficient to account for the initial assignment of significance by the learner to points in the input stream that then become anchors for later higher-level bootstrapping to full linguistic competence.

In the following, we will see how the morphological rules and paradigms of a language can be statistically induced without presupposing language specific knowledge, and how this can be used to derive word types using clustering algorithms.

## 3. Morphology Induction[1]

Identifying constituents on the morphological level, i.e. the morphemes in words, is comparable to the task of identifying constituents in the syntactic level. In this paper, we present an algorithm for morphology induction, called the Alignment-Based Unsupervised Grammar Inducer (ABUGI) that is inspired by the morphology induction work of Goldsmith (2001), in turn also by the syntactic induction algorithm presented in Van Zaanen (2001).

[1] The ABUGI Python source code is published under the GNU Public License and can be downloaded at http://jones.ling.indiana.edu/~abugi/.

The basic strategy used in ABL goes back to Harris (1961). The assumption is that matching substrings in two strings are potentially constituents at the respective linguistic level. The generated hypotheses are statistically evaluated on the basis of the statistical corpus properties, as well as some incrementally generated grammar fragment at the moment of hypothesis generation.
The main algorithm of the experimental grammar induction platform is an iterative procedure that takes utterances as input, generates hypotheses for the structural descriptions, and adds the most accurate hypotheses to the grammar and/or lexicon, compressing the grammar with every increment.
The components of the induction algorithm are described in detail in the following.

### 3.1 Hypothesis Generation using Alignment-Based Learning

The hypothesis generation algorithm is based on ABL. While different alternative algorithms are implemented in the current version of ABUGI, e.g. Longest Common Subsequence as well as randomized or statistical segmentation algorithms, ABL is used in the experiments discussed in the following. The output of the hypothesis generation phase is a set of structural hypotheses for the given input. It can be the result of the application of ABL alone, or combination of hypothesis generation algorithms. In the experiments discussed in the following, the hypothesis list is a set of potential sub-morphemes for the input word.
The ABL hypothesis generation algorithm operates by identifying substrings within a word, only if these substrings are already considered to be morphemes in the grammar/language, i.e. if they have been seen in the input before. A hypothesis is created by splitting the word at the edges of the matching substring, and forming a tuple of morphemes. For example, consider the word 'conformed.' The algorithm, if it had prior knowledge of the prefix 'con,' would generate a hypothesis ('con', 'formed'). If both 'con' and 'ed' were known or if 'form' was known then the algorithm would return ('con', 'form', 'ed') as a hypothesis. This process is referred to as alignment because the known morphemes are aligned against the target word at each character to determine if the substring starting with that character in the target word matches the current morpheme being searched for.
During the initial bootstrapping phase, if the word contains no known morphemes, it is not split, and the entire word is added as a morpheme.
Hypotheses can also be generated by using pattern matching with a set of predefined patterns. Common morphological patterns such as, ABA, ABBA, ABAB, etc. are compared against the word to see if substrings match up to the terminals in the pattern. If a word matches one of the patterns then the word is split into morphemes along the boundaries of the terminals in the pattern to create a hypothesis. These hypotheses differ from the previous kind in that none of the morphemes identified need have been seen before. This type processing is in principle only speeding up the general learning procedure (i.e. it is not crucial for the subsequent induction algorithm).

### 3.2. Hypothesis Evaluation

The evaluation of generated hypotheses involves the competitive interaction of a number of metrics that can be divided into two categories: those that favor conservation of memory space at the expense of processing time and those that favor decreasing processing time at the expense of memory. Hypotheses are selected based upon how they perform on all of the metrics. Winning hypotheses tend to be those that lie in the middle of the spectrum. These metrics can be thought of as constraints, preventing the hypotheses from being too extreme with regard to compression or processing time.
The interaction between the constraints is determined by weights assigned to each metric. So, for example, if highly compressed grammars are desired then metrics concerned with minimizing memory use can be given a high weight and those concerned with minimizing processing time a low weight. These weights form a constraint space with different types of grammars occupying different positions in the space. The challenge is to identify what weighting scheme is appropriate for the grammar being learned. We are currently investigating ways of learning these weights through optimization techniques such as genetic algorithms. One of the goals of this project is also identify where different languages lie in this constraint space and how languages move through the space as they evolve over time.
Each of our principles of evaluation, memory conservation and processing time conservation, have metrics associated with them that evaluate a hypothesis with regards to that principle. The memory-conserving

metrics are concerned with creating a highly compressed grammar composed of linguistic elements that are used frequently. These metrics favor identifying commonly used morphemes and in minimizing the size of the encoding for a word. These metrics also have a tendency to prefer more morphemes in a sentence so that morphemes can be reused more often which, as will be discussed later, is in direct contrast to the time-oriented metrics that try to minimize the number of morphemes. We tested three different metrics that favor compressed grammars and memory conservation: description length, mutual information and relative entropy.

### 3.2.1. Memory based constraints

#### 3.2.1.1. Description Length (DL)

The principle of Minimum Description Length (MDL), as used in recent work on grammar induction and unsupervised language acquisition, e.g. Goldsmith (2001), De Marcken (1996), and Grünwald (1998), explains the grammar induction process as an iterative minimization procedure of the grammar size, where the smaller grammar corresponds to the *best* grammar for the given data/corpus.
The description length metric, as we use it here, tells us how many bits of information would be required to store a word given a hypothesis of the morpheme boundaries, using our grammar. For each morpheme in the hypothesis that doesn't occur in the grammar we need to store the string representing the morpheme. For morphemes that do occur in our grammar we just need to store a pointer to that morpheme's entry in the grammar. We use a simplified calculation, taken from Goldsmith (2001), of the cost of storing a string that takes the number of bits of information required to store a letter of the alphabet and multiply it by the length of the string.

(1) $$\lg(len(alphabet)) * len(morpheme)$$

We have two different methods of calculating the cost of the pointer. The first takes a cue from Morse code and gives a variable cost depending on the frequency of the morpheme that it is pointing to. So first we calculate the frequency rank of the morpheme being pointed to, (e.g. the most frequent has rank 1, the second rank 2, etc.). We then calculate:

(2) $$\text{floor}(\lg(\text{frequency rank}) - 1)$$

to get a number of bits similar to the way Morse code assigns lengths to various letters.
The second is simpler and only calculates the entropy of the grammar of morphemes and uses this as the cost of all pointers to the grammar. The entropy equation is as follows:

(3) $$\sum_{x \in G} p(x) \lg \frac{1}{p(x)}$$

The second equation doesn't give variable pointer lengths, but it is preferred since it doesn't carry the heavy computational burden of calculating the frequency rank.
We calculate the description length for each hypothesis only,[2] by summing up the cost of each morpheme in the hypothesis. Those with low description lengths are favored.

#### 3.2.1.2. Mutual Information (MI)

For the purpose of this experiment we use a variant of standard Mutual Information (MI), see Charniak (1996) and MacKay (2003) for some use cases. Information theory tells us that the presence of a given morpheme restricts the possibilities of the occurrence of morphemes to the left and right, thus lowering the

[2] We do not calculate the sizes of the grammars with and without the given hypothesis.

amount of bits needed to store its neighbors. Thus we should be able to calculate the amount of bits needed by a morpheme to predict its right and left neighbors respectively. To calculate this, we have designed a variant of mutual information that is concerned with a single direction of information.
This is calculated in the following way. For every morpheme *y* that occurs to the right of *x* we sum the point-wise MI between *x* and *y*, relativizing it by the probability that *y* follows *x*, given that *x* occurs. This then gives us the expectation of the amount of information that *x* tells us about which morpheme will be to its right. Note that $p(< xy >)$ is the probability of the bigram $< xy >$ occurring and is not equal to $p(< yx >)$ which is the probability of the bigram $< yx >$ occurring.

We calculate the MI on the right side of $x \in G$ by:

(4) $$\sum_{y \in \{<Yx>\}} p(< yx > | x) \lg \frac{p(< xy >)}{p(x)}$$

and the MI on the left of $x \in G$ respectively by:

(5) $$\sum_{y \in \{<Yx>\}} p(< yx >) \lg \frac{p(< yx >)}{p(y)p(x)}$$

One way we use this as a metric, is by summing up the left and right MI for each morpheme in a hypothesis. We then look for the hypothesis that results in the maximal value of this sum. The tendency for this to favor hypotheses with many morphemes is countered by our criterion of favoring hypotheses that have fewer morphemes, which is discussed later. In practice, MI favors the reuse of morphemes, similar to MDL, because common morphemes will tend to have higher mutual information values.
Another way to use the left and right MI is in judging the quality of morpheme boundaries. In a good boundary, the morpheme on the left side should have high right MI and the morpheme on the right should have high left MI. Unfortunately, MI is not reliable in the beginning of the incremental learning procedure, because of the low frequency of words. Removing hypotheses with poor boundary scores prevents the algorithm from bootstrapping itself, as all boundaries are poor in the beginning. We are currently experimenting with phasing this in as MI is deemed more reliable in making these judgments.

3.2.1.3. Relative Entropy (RE)

We use RE as a measure for the cost of adding a hypothesis to the existing grammar. We look for hypotheses that when added to the grammar will result in a low divergence from the original grammar.
We calculate RE as a variant of the Kullback-Leibler Divergence, see Charniak (1996) or MacKay (2003). Given grammar $G_1$, the grammar generated so far, and $G_2$ the grammar with the extension generated for the new input increment, $P(X)$ is the probability mass function (*pmf*) for grammar $G_2$, and $Q(X)$ the *pmf* for grammar $G_1$:

(6) $$\sum_{x \in X} P(x) \lg \frac{P(x)}{Q(x)}$$

Note that with every new iteration a new element can appear, that is not part of $G_1$. Our variant of RE takes this into account by calculating the costs for such a new element *x* to be the point-wise entropy of this element in $P(X)$, summing up over all new elements:

(7) $$\sum_{x \in X} P(x) \lg \frac{1}{P(x)}$$

These two sums then form the RE between the original grammar and the new grammar with the addition of the hypothesis. Hypotheses with low RE are favored.
This metric behaves similarly to description length (discussed above), in that both calculate the distance between our original grammar and the grammar with the inclusion of the new hypothesis. The primary difference is that RE also takes into account how the probability mass function differs in the two grammars. Our variation also punishes new morphemes based upon their frequency relative to the frequency of other morphemes. MDL does not consider frequency in this way, which is why we are including RE as metric. We are currently investigating this to identify under what conditions they behave differently.

### 3.2.2. Time based constraints

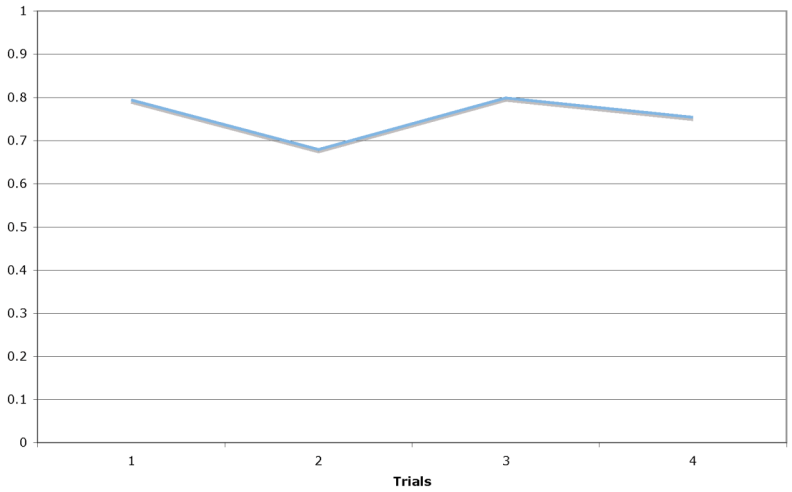
Metrics that conserve processing time are designed to prevent the grammar from becoming too compressed; a situation which can be thought of as the formation of too many morphological rules (which are time consuming to process). These metrics favor few morphological boundaries so that the word can be accessed rapidly with minimal time spend composing the word from its morphemes. The metrics used to evaluate this are: frequency of morpheme boundaries, number of morpheme boundaries, and length of morphemes.
In addition to the memory based metric, we take into account the following criteria:

- Frequency of Morpheme Boundaries
- Number of Morpheme Boundaries
- Length of Morphemes

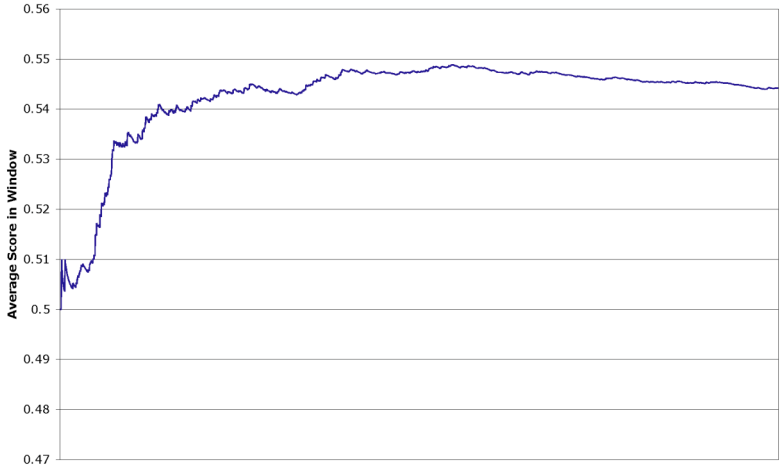
The frequency of morpheme boundaries is given by the number of hypotheses that contain this boundary. The basic intuition is that the higher this number, i.e. the more alignments are found at a certain position within a word, the more likely this position represents a morpheme boundary. We favor hypotheses with high values for this criterion.
The number of morpheme boundaries indicates how many morphemes the word was split into. To prevent the algorithm from degenerating into the state where each letter is identified as a morpheme, we favor hypotheses with low number of morpheme boundaries.
The length of the morphemes is also taken into account. We favor hypotheses with long morphemes to prevent the same degenerate state as the above criterion.

## 4. Results

We used two methods to evaluate the performance of the algorithm. The first analyzes the accuracy of the morphological rules produced by the algorithm after an increment of *n* words. The second looks at how accurately the algorithm parsed each word that it encountered as it progressed through the corpus.
The morphological rule analysis looks at each grammar rule generated by the algorithm and judges it on the correctness of the rule and the resulting parse. A grammar rule consists of a stem and the suffixes and prefixes that can be attached to it, similar to the signatures used in Goldsmith (2001). The grammar rule was then marked as to whether it consisted of legitimate suffixes and prefixes for that stem, and also as to whether the stem of the rule was a true stem, as opposed to a stem plus another morpheme that wasn't identified by the algorithm. The number of rules that were correct in these two categories were summed, and precision and recall figures were calculated for the trial. The trials described in the graph below were run on three increasingly large portions of the general fiction section of the Brown Corpus. The first trial was run on one randomly chosen chapter, the second trial on two chapters, and the third on three chapters. The graph shows the harmonic average (F-score) of precision and recall.




The second analysis is conducted as the algorithm is running and examines each parse the system produces. The algorithm's parses are compared with the "correct" morphological parse of the word using the following method to derive a numerical score for a particular parse. The first part of the score is the distance in characters between each morphological boundary in the two parses, with a score of one point for each character space. The second part is a penalty of two points for each morphological boundary that occurs in one parse and not the other. These scores were examined within a moving window of words that progressed through the corpus as the algorithm ran. The average scores of words in each such window were calculated as the window advanced. The purpose of this method was to allow the performance of the algorithm to be judged at a given point without prior performance in the corpus affecting the analysis of the current window. The following graph shows how the average performance of the windows of analyzed words as the algorithm progresses through five randomly chosen chapters of general fiction in the Brown Corpus amounting to around 10,000 words. The window size for the following graph was set to 40 words.

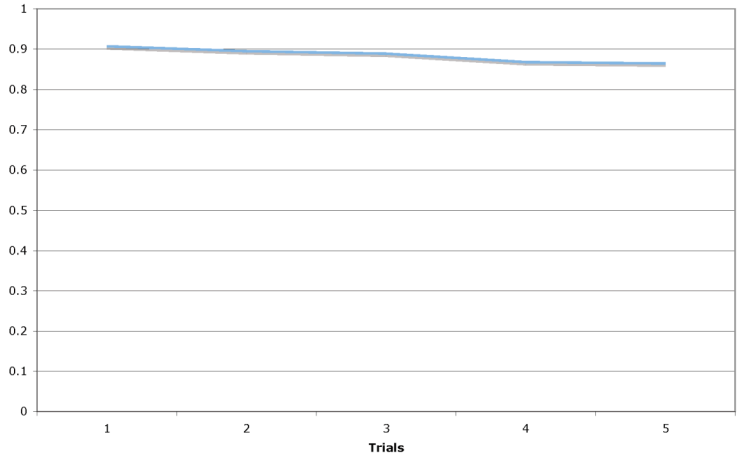



The evaluations on Latin were based on the initial 4000 words of "De Bello Gallico" in a pretest. In the very initial phase we reached a precision of 99.5% and a recall of 13.2%. This is however the preliminary result. We expect that for a larger corpus the recall will increase with a delay, given the rich morphology of Latin. The richer the morphology, the more data we need to reach statistical significance with some evaluation metrics.

The results on the Peter corpus are shown in the following table:

| After file | precision | recall |
|---|---|---|
| 01 | .9957 | .8326 |
| 01-03 | .9968 | .8121 |
| 01-05 | .9972 | .8019 |
| 01-07 | .9911 | .7710 |
| 01-09 | .9912 | .7666 |

We notice a more or less stable precision value with decreasing recall, due to a higher number of words. The Peter corpus contains also many very specific transcriptions and tokens that are indeed unique, thus it is rather surprising to get such results at all. The following graphics shows the F-score for the Peter corpus:




## 5. Conclusion

The rules we generate match well with those that we know to exist in English: among other things, the system correctly identifies prefixes on verbs and the "-s" plural ending for nouns. This is driven entirely by their frequency in the corpora. In the manually segmented portion of the Brown corpus we identified on the token level 11.3% of the inflectional morphemes, 6.4% of the derivational morphemes, and 82.1% of stems. On average there are twice as many inflectional morphemes in the corpus as derivational.

Given very strict parameters, focusing on the description length of the grammar, our system would need a long time to discover prefixes, not to mention infixes. By relaxing the weight of the description length constraint we can inhibit the generation and identification of prefixing rules. This is done, however, at the cost of precision.

Given these results, it is clear that inflectional paradigms are extractable even with an incremental approach. This means that central parts of the lexicon can be induced very early along the time line.

Additionally, we provide a method by which to test the roles different statistical algorithms play in this process. By adjusting the weights of the contributions made by various constraints, we can approach an understanding of the optimal ordering of algorithms that play a role in the computational framework of language acquisition.

### 5.1. Using Morphological Signatures

Given the extremely compact and precise morphological grammar we were able to generate with ABUGI, lexical classification can be performed on the basis of the resulting signatures. One immediate consequence is that the notion of stem or suffix is a side effect of the signature properties. The signature of a stem is short, for English it contains not more than a dozen elements, with every element being highly frequent. For affixes, however, the signatures will contain many low frequent elements, i.e. usually stems. A clustering algorithm (e.g. K-means) groups these elements into two clusters, using optimization of the error calculated on the basis of the length of the signature and the frequency of the contained elements as the clustering criterion. The type of signature allows for separation of lexical classes, e.g. verbs and nouns will have different signatures, as well as most other lexical classes. Along these lines, morphemes with a similar signature can be replaced by symbols, expressing the same type information and compressing the grammar further. This type information, especially for rare morphemes, is essential in subsequent induction of syntactic structure. Due to space limitations, we cannot discuss in detail subsequent steps in the cross-level induction procedures. Nevertheless, the model presented here provides an important pointer to the mechanics of how grammatical parameters might come to be set.

On the other hand, different algorithms make use of similar morphological classification in Part-of-speech (POS) tagging, cf. Brants (2000), Lee et al (2002).

For example, of the words in the WSJ section of Penn that end in "able," 98% are adjectives, and only 2% are nouns (e.g. "cable", "variable") (see Brants, 2000). In these cases, this suffix highly predicts the categorization of the word. This information could be used to aid POS taggers. Samuelsson (1994) introduced an algorithm to utilize these end-of-word substring "suffixes" to categorize words into POSs by taking probabilities of substring word endings of 7 characters or less and smoothing this over by averaging in the probability with one less character each iteration. TnT (Brants, 2000), a statistical n-gram POS tagger, uses an implementation of this algorithm as the primary component of its algorithm to tag words not seen in the training corpus. Brants (2000) reports 89.0% accuracy on these unknown words using the Penn Treebank as a corpus.

Lee et al (2002) performed a similar experiment on Korean. Lee et al's (2002) approach uses a morpheme pattern database to automatically tag the complex agglutinative morphology of Korean eojeols. After assigning all possible morpheme tags to a morpheme, the text is run through a statistical POS tagger, which uses the Viterbi algorithm to assign word categories. This is then run through a correction layer, using a rule-based correction system. Even though 10% of the words were unknown, Lee reports a 97% tagging accuracy.

Precision and Recall of the morphologic component of TnT was not unreported. Lee reported a 94.9% recall and 89.7% precision on the Korean data. ABUGI's precision and less reliance on programmer-supplied knowledge, makes it an attractive replacement for either of these systems. However, a real comparison on Korean data would require tests with ABUGI on the same data, which has not been done yet.

## REFERENCES


Brants, T. 2000. "TnT - A Statistical Part-of-Speech Tagger." In proceedings of ANLP 2000, 6th Applied Natural Language Processing Conference, April 29 - May 4, Seattle, Washington, USA. 224-231.

Ćavar, D. and K. Elghamry 2004. *Bootstrapping cues for cue-based bootstrapping*. Mscr. Indiana University.

Charniak, E. 1996. *Statistical language learning*. MIT Press, Cambridge, Mass.

De Marcken, C.G. 1996. Unsupervised Language Acquisition. PhD dissertation, MIT, Cambridge, Massachusetts.

Eklund, R. 1994. Proceedings of the 9th Nordiska Datalingvistikdagarna (NODALIDA 1993). Stockholm. 3-5 June 1993. Stockholm University, ISBN-91-7153-262-5 Retrieved from http://www.ida.liu.se/~g-robek/nodalida93/nodalida93/NODA93-21/NODA93-21.html9.

Goldsmith, J. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27.2. 153-198.

Grünwald, P. 1998. The Minimum Description Length Principle and Reasoning under Uncertainty. Doctoral dissertation, Universiteit van Amsterdam.

Geunbae, G. and J.-H. Lee and C. Jeongwon. 2002. Syllable-Pattern-Based Unknown-Morpheme Segmentation and Estimation for Hybrid Part-of-Speech Tagging of Korean. *Computational Linguistics* 28.1. 53-70.

Harris, Z.S. 1961. *Structural linguistics*. University of Chicago Press. Chicago. Published in 1951 under title: Methods in structural linguistics.

Lee, G.G. and J.-H. Lee and J. Cha. 2002. Syllable-Pattern-Based Unknown-Morpheme Segmentation and Estimation for Hybrid Part-of-Speech Tagging of Korean. *Computational Linguistics*. 28.1. 53-70.

MacKay, D.J.C. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press.

Samuelsson, C. 1994. Morphological Tagging Based Entirely on Bayesian Inference. In Eklund, R. (ed.).

Van Zaanen, M. 2001. Bootstrapping Structure into Language: Alignment-Based Learning. Doctoral dissertation, The University of Leeds.