Proceedings of the

First Workshop on

Psycho-computational Models of Human Language Acquisition

Held in cooperation with COLING-2004

28-29 August 2004
Geneva, Switzerland

INVITED SPEAKERS:

Walter Daelemans	(University of Antwerp, Belgium and Tilburg University, the Netherlands)
B. Elan Dresher	(University of Toronto, Canada)
Charles Yang	(Yale University, USA)

ORGANIZER:

William Gregory Sakas	(City University of New York, USA)

PROGRAM COMMITTEE:

Robert Berwick	(MIT, USA)
Antal van den Bosch	(Tilburg University, the Netherlands)
Ted Briscoe	(University of Cambridge, UK)
Damir Cavar	(Indiana University, USA)
Morten H. Christiansen	(Cornell University, USA)
Stephen Clark	(University of Edinburgh, UK)
James Cussens	(University of York, UK)
Walter Daelemans	(University of Antwerp, Belgium and Tilburg University, the Netherlands)
Jeffrey Elman	(University of California, San Diego, USA)
Gerard Kempen	(Leiden University, the Netherlands and the Max Planck Institute, Nijmegen)
Vincenzo Lombardo	(University of Torino, Italy)
Larry Moss	(University of Indiana, USA)
Miles Osborne	(University of Edinburgh, UK)
Dan Roth	(University of Illinois at Urbana-Champaign, USA)
Ivan Sag	(Stanford University, USA)
Jeffrey Siskind	(Purdue University, USA)
Mark Steedman	(University of Edinburgh, UK)
Menno van Zaanen	(Tilburg University, the Netherlands)
Charles Yang	(Yale University, USA)

WORKSHOP ASSISTANTS:

Xuan Nga Cao (City University of New York, USA)
Mari Fujimoto (City University of New York, USA)
Lydiya Tornyova (City University of New York, USA)

SPONSOR:

The 20th International Conference on Computational Linguistics

FURTHER INFORMATION:

William Gregory Sakas
Ph.D. Programs in Linguistics and Computer Science
Department of Computer Science, North Bldg1008
Hunter College, City University of New York
695 Park Ave
New York, NY 10021
USA

email: sakas@hunter.cuny.edu
psycho.comp@hunter.cuny.edu

WWW: http://www.colag.cs.hunter.cuny.edu/psychocomp

Introduction

Every day, we use language so effortlessly that we often overlook its complexity. The fact that language *is* complex is indisputable. Indeed, even after decades of scrutiny, highly-trained adult scientists cannot agree on a definitive analysis of the underlying mechanism that ultimately determines how our sounds, words, and sentences go together – but such an effortless task for a child! Children as young as one-and-a-half-years-old (and younger) continually exploit much of language's underpinnings while going about the business of making sense of the linguistic environment that surrounds them. By the time a child reaches kindergarten, he or she has almost full mastery of an elaborate structure that eludes adequate scientific description. How children accomplish this – how they come to acquire 'knowledge' of language's essential organization – is one of the most fundamental, beguiling, and surprisingly open questions of modern science.

This workshop brings together researchers whose (at least one) line of investigation is to computationally model the acquisition process and ascertain substantive interrelationships between a model and linguistic and psycholinguistic theory. Progress in this agenda not only directly informs developmental psycholinguistic and linguistic research, but in my opinion, will also have the long term benefit of informing applied computational linguistics in areas that involve the automated acquisition of knowledge from a human or human-computer linguistic environment.

The level of sophistication and breadth of applied computational linguistics techniques has skyrocketed in the past two decades. There is now a battery of computational formalisms and statistical methods to 'choose from,' all which have yielded remarkable success in many applied domains that involve the computer learning of natural language (e.g. speech recognition, web technologies, corpus analysis, etc). These achievements have dramatically spurred even more research and funding to the point where the evolution of the science of computational linguistics can be seen as quickly outpacing that of psycholinguistics.

However, there are signs that the computational linguistics community has been progressively more aware that language technologies might benefit by incorporating learning strategies employed by humans. Although research involving the psycho-computational modeling of human language acquisition has been long active in the areas of psycholinguistics, cognitive science and formal learning theory, it has, arguably, only recently become a growing part of the computational linguistics agenda. This is evidenced by the occasional special session at an ACL meeting (e.g., ACL-1999 – Thematic Session on Computational Psycholinguistics), current workshops at both COLING-2004 (this workshop) and ACL-2004 (Incremental Parsing: Bringing Engineering and Cognition Together), and regular invitations to developmental psycholinguists to deliver plenary addresses at recent ACL meetings. This cross-discipline attentiveness is clearly very healthy and might well help reduce the possibility that applied research will run into a *psycho-computational bottleneck* – when state-of-the-art computational methods cannot be improved further in the development of user-transparent computer-human language applications – by incorporating theoretical advances in computational psycholinguistics into computational language learning technologies.

This workshop brings together a wide range of computational psycholinguistics research that is involved with the study of language acquisition: 34% of author contributions come from researchers holding positions in computer science or related departments, 33% from linguistics departments, 30% from psychology or cognitive science departments, and 3% from other departments.[1] The articles present investigations involving a broad diversity of formalisms, learning strategies, modeling techniques and linguistic phenomena. Linguistic footings range from (variations on): Universal Grammar, constructionist frameworks, and categorial grammar, to novel formulations of structural representation, to 'none.' Learning strategies include: distributional and corpus techniques, connectionist implementations, cue-based learning, and hybrid models that apply several strategies. Phenomena that are modeled include: the acquisition of semantics, linguistic (principles and) parameter setting, lexical subcategorization, child language production, atypical acquisition, phonological acquisition and morphological acquisition. Several papers involve cross-linguistic research and/or use actual child-directed speech (from corpora).

[1] An "author contribution" is calculated as 1 / the number of authors on a paper.

Notably, most papers (not all) address acquisition at the sub-word, word, or multi-word level. Few models assign structure or meaning to an entire utterance (or discourse) although many papers suggest that a presented model could be (easily) scaled-up – a worthwhile direction for future research. It is also worth remarking on the fact that articles addressing formal learning issues (e.g., PAC learning, identification in the limit, grammar induction, etc.) or that incorporate formalisms from mainstream computational linguistics (e.g., any of the many variants of probabilistic grammars) are underrepresented (the workshop contains one such). Future meetings along the lines of this workshop might benefit from attracting research efforts related to these approaches.

I would sincerely like to thank the program committee for above-and-beyond effort given the tight timetable, the diversity of the papers, and the several frustrating problems caused by spam-blockers; the workshop assistants who were a tremendous help with collating the reviews, organizing the articles for the proceedings, dealing with email and designing the conference web site; and, finally, the members of the COLING-2004 Workshop Program Committee, who were extremely helpful (and patient) on more than one occasion.

William Gregory Sakas
New York City
June 2004

Table of Contents

A Quantitative Evaluation of Naturalistic Models of Language Acquisition; the Efficiency of the Triggering Learning Algorithm Compared to a Categorial Grammar Learner
Paula Buttery……....1

On Statistical Parameter Setting
Damir Ćavar, Joshua Herring, Toshikazu Ikuta, Paul Rodrigues and Giancarlo Schrementi 9

Putting Meaning into Grammar Learning
Nancy Chang 17

Grammatical Inference and First Language Acquisition
Alexander Clark 25

A Developmental Model of Syntax Acquisition in the Construction Grammar Framework with Cross-Linguistic Validation in English and Japanese
Peter Ford Dominey and Toshio Inui 33

On the Acquisition of Phonological Representations
B. Elan Dresher 41

Statistics Learning and Universal Grammar: Modeling Word Segmentation
Timothy Gambell and Charles Yang 49

Modelling Syntactic Development in a Cross-Linguistic Context
Fernand Gobet, Daniel Freudenthal and Julian M. Pine 53

A Computational Model of Emergent Simple Syntax: Supporting the Natural Transition from the One-Word Stage to the Two-Word Stage
Kris Jack, Chris Reed and Annalu Waller 61

On a Possible Role for Pronouns in the Acquisition of Verbs
Aarre Laakso and Linda Smith 69

Some Tests of an Unsupervised Model of Language Acquisition
Bo Pedersen, Shimon Edelman, Zach Solan, David Horn and Eytan Ruppin 77

Modelling Atypical Syntax Processing
Michael S. C. Thomas and Martin Redington 85

Combining Utterance-Boundary and Predictability Approaches to Speech Segmentation
Aris Xanthos 93

On Statistical Parameter Setting

Damir ĆAVAR, Joshua HERRING, Toshikazu IKUTA, Paul RODRIGUES
Linguistics Dept., Indiana University
Bloomington, IN, 46405
dcavar@indiana.edu

Giancarlo SCHREMENTI
Computer Science, Indiana University
Bloomington, IN, 47405
gischrem@indiana.edu

Abstract

We present a model and an experimental platform of a bootstrapping approach to statistical induction of natural language properties that is constraint based with voting components. The system is incremental and unsupervised. In the following discussion we focus on the components for morphological induction. We show that the much harder problem of incremental unsupervised morphological induction can outperform comparable all-at-once algorithms with respect to precision. We discuss how we use such systems to identify cues for induction in a cross-level architecture.

1 Introduction

In recent years there has been a growing amount of work focusing on the computational modeling of language processing and acquisition, implying a cognitive and theoretical relevance both of the models as such, as well as of the language properties extracted from raw linguistic data.[1] In the computational linguistic literature several attempts to induce grammar or linguistic knowledge from such data have shown that at different levels a high amount of information can be extracted, even with no or minimal supervision.

Different approaches tried to show how various puzzles of language induction could be solved. From this perspective, language acquisition is the process of segmentation of non-discrete acoustic input, mapping of segments to symbolic representations, mapping representations on higher-level representations such as phonology, morphology and syntax, and even induction of semantic properties. Due to space restrictions, we cannot discuss all these approaches in detail. We will focus on the close domain of morphology.

Approaches to the induction of morphology as presented in e.g. Schone and Jurafsky (2001) or Goldsmith (2001) show that the morphological properties of a small subset of languages can be induced with high accuracy, most of the existing approaches are motivated by applied or engineering concerns, and thus make assumptions that are less cognitively plausible: a. Large corpora are processed all at once, though unsupervised incremental induction of grammars is rather the approach that would be relevant from a psycholinguistic perspective; b. Arbitrary decisions about selections of sets of elements are made, based on frequency or frequency profile rank,[2] though such decisions should rather be derived or avoided in general.

However, the most important aspects missing in these approaches, however, are the link to different linguistic levels and the support of a general learning model that makes predictions about how knowledge is induced on different linguistic levels and what the dependencies between information at these levels are. Further, there is no study focusing on the type of supervision that might be necessary for the guidance of different algorithm types towards grammars that resemble theoretical and empirical facts about language acquisition, and processing and the final knowledge of language.

While many theoretical models of language acquisition use innateness as a crutch to avoid outstanding difficulties, both on the general and abstract level of I-language as well as the more detailed level of E-language, (see, among others, Lightfoot (1999) and Fodor and Teller (2000), there is also significant research being done which shows that children take advantage of statistical regularities in the input for use in the language-learning task (see Batchelder (1997) and related references within).

In language acquisition theories the dominant view is that knowledge of one linguistic level is bootstrapped from knowledge of one, or even several different levels. Just to mention such approaches: Grimshaw (1981), and Pinker (1984)

[1] See Batchelder (1998) for a discussion of these aspects.

[2] Just to mention some of the arbitrary decisions made in various approaches, e.g. Mintz (1996) selects a small set of all words, the most frequent words, to induce word types via clustering ; Schone and Jurafsky (2001) select words with frequency higher than 5 to induce morphological segmentation.

assume that semantic properties are used to bootstrap syntactic knowledge, and Mazuka (1998) suggested that prosodic properties of language establish a bias for specific syntactic properties, e.g. headedness or branching direction of constituents. However, these approaches are based on conceptual considerations and psycholinguistc empirical grounds, the formal models and computational experiments are missing. It is unclear how the induction processes across linguistic domains might work algorithmically, and the quantitative experiments on large scale data are missing.

As for algorithmic approaches to cross-level induction, the best example of an initial attempt to exploit cues from one level to induce properties of another is presented in Déjean (1998), where morphological cues are identified for induction of syntactic structure. Along these lines, we will argue for a model of statistical cue-based learning, introducing a view on bootstrapping as proposed in Elghamry (2004), and Elghamry and Ćavar (2004), that relies on identification of elementary cues in the language input and incremental induction and further cue identification across all linguistic levels.

1.1 Cue-based learning

Presupposing input driven learning, it has been shown in the literature that initial segmenations into words (or word-like units) is possible with unsupervised methods (e.g. Brent and Cartwright (1996)), that induction of morphology is possible (e.g. Goldsmith (2001), Schone and Jurafsky (2001)) and even the induction of syntactic structures (e.g. Van Zaanen (2001)). As mentioned earlier, the main drawback of these approaches is the lack of incrementality, certain arbitrary decisions about the properties of elements taken into account, and the lack of integration into a general model of bootstrapping across linguistic levels.

As proposed in Elghamry (2004), cues are elementary language units that can be identified at each linguistic level, dependent or independent of prior induction processes. That is, intrinsic properties of elements like segments, syllables, morphemes, words, phrases etc. are the ones available for induction procedures. Intrinsic properties are for example the frequency of these units, their size, and the number of other units they are build of. Extrinsic properties are taken into account as well, where extrinsic stands for distributional properties, the context, relations to other units of the same type on one, as well as across linguistic levels. In this model, extrinsic and intrinsic properties of elementary language units are the cues that are used for grammar induction only.

As shown in Elghamry (2004) and Elghamry and Ćavar (2004), there are efficient ways to identify a kernel set of such units in an unsupervised fashion without any arbitrary decision where to cut the set of elements and on the basis of what kind of features. They present an algorithm that selects the set of kernel cues on the lexical and syntactic level, as the smallest set of words that co-occurs with all other words. Using this set of words it is possible to cluster the lexical inventory into open and closed class words, as well as to identify the subclasses of nouns and verbs in the open class. The direction of the selectional preferences of the language is derived as an average of point-wise Mutual Information on each side of the identified cues and types, which is a self-supervision aspect that biases the search direction for a specific language. This resulting information is understood as derivation of secondary cues, which then can be used to induce selectional properties of verbs (frames), as shown in Elghamry (2004).

The general claim thus is:

- Cues can be identified in an unsupervised fashion in the input.
- These cues can be used to induce properties of the target grammar.
- These properties represent cues that can be used to induce further cues, and so on.

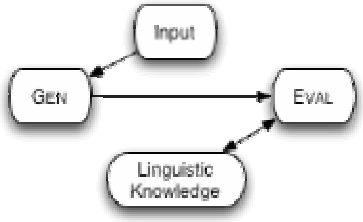
The hypothesis is that this snowball effect can reduce the search space of the target grammar incrementally. The main research questions are now, to what extend do different algorithms provide cues for other linguistic levels and what kind of information do they require as supervision in the system, in order to gain the highest accuracy at each linguistic level, and how does the linguistic information of one level contribute to the information on another.

In the following, the architectural considerations of such a computational model are discussed, resulting in an example implementation that is applied to morphology induction, where morphological properties are understood to represent cues for lexical clustering as well as syntactic structure, and vice versa, similar to the ideas formulated in Déjean (1998), among others.

1.2 Incremental Induction Architecture

The basic architectural principle we presuppose is incrementality, where incrementally utterances are processed. The basic language unit is an utterance, with clear prosodic breaks before and after. The induction algorithm consumes such utterances and breaks them into basic linguistic units, generating for each step hypotheses about

the linguistic structure of each utterance, based on the grammar built so far and statistical properties of the single linguistic units. Here we presuppose a successful segmentation into words, i.e. feeding the system utterances with unambiguous word boundaries. We implemented the following pipeline architecture:



The GEN module consumes input and generates hypotheses about its structural descriptions (SD). EVAL consumes a set of SDs and selects the set of best SDs to be added to the knowledge base. The knowledge base is a component that not only stores SDs but also organizes them into optimal representations, here morphology grammars.

All three modules are modular, containing a set of algorithms that are organized in a specific fashion. Our intention is to provide a general platform that can serve for the evaluation and comparison of different approaches at every level of the induction process. Thus, the system is designed to be more general, applicable to the problem of segmentation, as well as type and grammar induction.

We assume for the input to consist of an alphabet: a non-empty set A of n symbols $\{s_1, s_2,... s_n\}$. A word w is a non-empty list of symbols $w = [s_1,s_2,... s_n]$, with $s \in A$. The corpus is a non-empty list C of words $C = [w_1,w_2,... w_n]$.

In the following, the individual modules for the morphology induction task are described in detail.

1.2.1 GEN

For the morphology task GEN is compiled from a set of basically two algorithms. One algorithm is a variant of Alignment Based Learning (ABL), as described in Van Zaanen (2001).

The basic ideas in ABL go back to concepts of *substitutability* and/or *complementarity*, as discussed in Harris (1961). The concept of *substitutability* generally applies to central part of the induction procedure itself, i.e. substitutable elements (e.g. substrings, words, structures) are assumed to be of the same type (represented e.g. with the same symbol).

The advantage of ABL for grammar induction is its constraining characteristics with respect to the set of hypotheses about potential structural properties of a given input. While a brute-force method would generate all possible structural representations for the input in a first order explosion and subsequently filter out irrelevant hypotheses, ABL reduces the set of possible SDs from the outset to the ones that are motivated by previous experience/input or a pre-existing grammar.

Such constraining characteristics make ABL attractive from a cognitive point of view, both because hopefully the computational complexity is reduced on account of the smaller set of potential hypotheses, and also because learning of new items, rules, or structural properties is related to a general learning strategy and previous experience only. The approaches that are based on a brute-force first order explosion of all possible hypotheses with subsequent filtering of relevant or irrelevant structures are both memory-intensive and require more computational effort.

The algorithm is not supposed to make any assumptions about types of morphemes. There is no expectation, including use of notions like *stem*, *prefix*, or *suffix*. We assume only linear sequences. The properties of single morphemes, being stems or suffixes, should be a side effect of their statistical properties (including their frequency and co-occurrence patterns, as will be explained in the following), and their alignment in the corpus, or rather within words.

There are no rules about language built-in, such as what a morpheme must contain or how frequent it should be. All of this knowledge is induced statistically.

In the ABL Hypotheses Generation, a given word in the utterance is checked against morphemes in the grammar. If an existing morpheme LEX aligns with the input word INP, a hypothesis is generated suggesting a morphological boundary at the alignment positions:

INP (*speaks*) + LEX (*speak*) = HYP [*speak*, *s*]

Another design criterion for the algorithm is complete language independence. It should be able to identify morphological structures of Indo-European type of languages, as well as agglutinative languages (e.g. Japanese and Turkish) and polysynthetic languages like some Bantu dialects or American Indian languages. In order to guarantee this behavior, we extended the Alignment Based hypothesis generation with a pattern identifier that extracts patterns of character sequences of the types:

1. A — B — A
2. A — B — A — B
3. A — B — A — C

This component is realized with cascaded regular expressions that are able to identify and

return the substrings that correspond to the repeating sequences.[3]

All possible alignments for the existing grammar at the current state, are collected in a hypothesis list and sent to the EVAL component, described in the following. A hypothesis is defined as a tuple:

$H = <w, f, g>$, with w the input word, f its frequency in C, and g a list of substrings that represent a linear list of morphemes in w, $g = [m_1, m_2, ... m_n]$.

1.2.2 EVAL

EVAL is a voting based algorithm that subsumes a set of independent algorithms that judge the list of SDs from the GEN component, using statistical and information theoretic criteria. The specific algorithms are grouped into memory and usability oriented constraints.

Taken as a whole, the system assumes two (often competing) cognitive considerations. The first of these forms a class of what we term "time-based" constraints on learning. These constraints are concerned with the processing time required of a system to make sense of items in an input stream, whereby "time" is understood to mean the number of steps required to generate or parse SDs rather than the actual temporal duration of the process. To that end, they seek to minimize the amount of structure assigned to an utterance, which is to say they prefer to deal with as few rules as possible. The second of these cognitive considerations forms a class of "memory-based" constraints. Here, we are talking about constraints that seek to minimize the amount of memory space required to store an utterance by maximizing the efficiency of the storage process. In the specific case of our model, which deals with morphological structure, this means that the memory-based constraints search the input string for regularities (in the form of repeated substrings) that then need only be stored once (as a pointer) rather than each time they are found. In the extreme case, the time-based constraints prefer storing the input "as is", without any processing at all, where the memory-based constraints prefer a rule for every character, as this would assign maximum structure to the input. Parsable information falls out of the tension between these two conflicting constraints, which can then be applied to organize the input into potential syntactic categories. These can then be used to set the parameters for the internal adult parsing system.

Each algorithm is weighted. In the current implementation these weights are set manually. In future studies we hope to use the weighting for self-supervision.[4] Each algorithm assigns a numerical rank to each hypothesis multiplied with the corresponding weight, a real number between 0 and 1.

On the one hand, our main interest lies in the comparison of the different algorithms and a possible interaction or dependency between them. Also, we expect the different algorithms to be of varying importance for different types of languages.

Mutual Information (MI)

For the purpose of this experiment we use a variant of standard Mutual Information (MI), see e.g. MacKay (2003). 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 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, but we relativize the point-wise MI 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:

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

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

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

One way we use this as a metric, is by summing up the left and right MI for each morpheme in a

[3] This addition might be understood to be a sort of *supervision* in the system. However, as shown in recent research on human cognitive abilities, and especially on the ability to identify patterns in the speech signal by very young infants (Marcus et al, 1999) shows that we can assume such an ability to be part of the cognitive abilities, maybe not even language specific

[4] One possible way to self-supervise the weights in this architecture is by taking into account the revisions subsequent components make when they optimize the grammar. If rules or hypotheses have to be removed from the grammar due to general optimization constraints on the grammars as such, the weight of the responsible algorithm can be lowered, decreasing its general value in the system on the long run. The relevant evaluations with this approach are not yet finished.

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, discussed later.

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 because of the low frequency of morphemes. However, as the lexicon is extended during the induction procedure, reliable frequencies are bootstrapping this segmentation evaluation.

Minimum 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) and De Marcken (1996), 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 the so far generated 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 morphemes 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.

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

We have two different methods of calculating the cost of the pointer. The first assigns a variable the cost based 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:

$$floor(\lg(freq\_rank) - 1)$$

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:

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

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 GEN hypothesis only,[5] by summing up the cost of each morpheme in the hypothesis. Those with low description lengths are favored.

Relative Entropy (RE)

We are using 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 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:

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

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:

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

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, that is discussed above, in that both are calculating the distance between our original grammar and the grammar with the inclusion of the new hypothesis. The primary difference is RE also takes into account how the *pmf* differs in the two grammars and that our variation punishes new morphemes based upon their frequency relative to the frequency of other morphemes. Our implementation of MDL does not consider frequency in this way, which is why we are including RE as an independent metric.

Further Metrics

In addition to the mentioned metric, we take into account the following criteria: a. Frequency of

[5] We do not calculate the sizes of the grammars with and without the given hypothesis, just the amount each given hypothesis would add to the grammar, favoring the least increase of total grammar size.

morpheme boundaries; b. Number of morpheme boundaries; c. 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 is, 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.

1.2.3 Linguistic Knowledge

The acquired lexicon is stored in a hypothesis space which keeps track of the words from the input and the corresponding hypotheses. The hypothesis space is defined as a list of hypotheses:

Hypotheses space: $S = [H_1, H_2, ... H_n]$

Further, each morpheme that occurred in the SDs of words in the hypothesis space is kept with its frequency information, as well as bigrams that consist of morpheme pairs in the SDs and their frequency.[6]

Similar to the specification of signatures in Goldsmith (2001), we list every morpheme with the set of morphemes it co-occurs. Signatures are lists of morphemes. Grammar construction is performed by replacement of morphemes with a symbol, if they have equal signatures.

The hypothesis space is virtually divided into two sections, long term and short term storage. Long term storage is not revised further, in the current version of the algorithm. The short term storage is cyclically cleaned up by eliminating the signatures with a low likelihood, given the long term storage.

2 The experimental setting

In the following we discuss the experimental setting. We used the Brown corpus,[7] the child-oriented speech portion of the CHILDES Peter corpus,[8] and Caesar's "De Bello Gallico" in Latin.[9]

From the Brown corpus we used the files ck01 – ck09, with an average number of 2000 words per chapter. The total number of words in these files is 18071. The randomly selected portion of "De Bello Gallico" contained 8300 words. The randomly selected portion of the Peter corpus contains 58057 words.

The system reads in each file and dumps log information during runtime that contains the information for online and offline evaluation, as described below in detail.

The gold standard for evaluation is based on human segmentation of the words in the respective corpora. We create for every word a manual segmentation for the given corpora, used for online evaluation of the system for accuracy of hypothesis generation during runtime. Due to complicated cases, where linguist are undecided about the accurate morphological segmentation, a team of 5 linguists was cooperating with this task.

The offline evaluation is based on the grammar that is generated and dumped during runtime after each input file is processed. The grammar is manually annotated by a team of linguists, indicating for each construction whether it was segmented correctly and exhaustively. An additional evaluation criterion was to mark undecided cases, where even linguists do not agree. This information was however not used in the final evaluation.

2.1 Evaluation

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

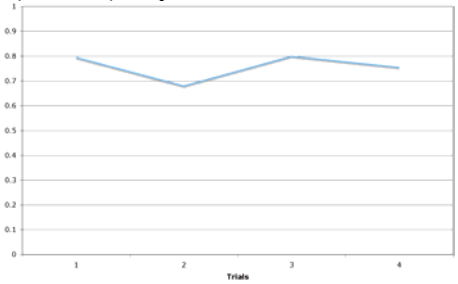
[6] Due to space restrictions we do not formalize this further. A complete documentation and the source code is available at: http://jones.ling.indiana.edu/~abugi/.

[7] The Brown Corpus of Standard American English, consisting of 1,156,329 words from American texts printed in 1961 organized into 59,503 utterances and compiled by W.N. Francis and H. Kucera at Brown University.

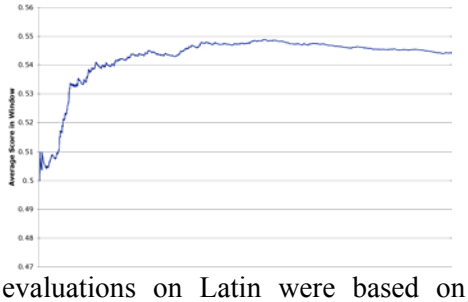
[8] Documented in L. Bloom (1970) and available at http://xml.talkbank.org:8888/talkbank/file/CHILDES/Eng-USA/Bloom70/Peter/.

[9] This was taken from the Gutenberg archive at: http://www.gutenberg.net/etext/10657. The Gutenberg header and footer were removed for the experimental run.

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 then 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.

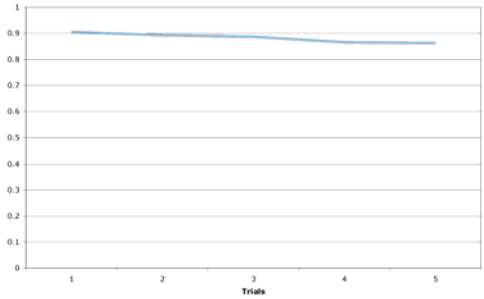


The evaluations on Latin were based on the initial 4000 words of "De Bello Gallico" in a pretest. In the very initial phase we reached a precision of 99.5% and a recall of 13.2%. This is however the preliminary result for the initial phase only. We expect that for a larger corpus the recall will increase much higher, given the rich morphology of Latin, potentially with negative consequences for precision.

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:



3 Conclusion

The evaluations on two related morphology systems show that with a restrictive setting of the parameters in the described algorithm, approx 99% precision can be reached, with a recall higher than 60% for the portion of the Brown corpus, and even higher for the Peter corpus.

We are able to identify phases in the generation of rules that turn out to be for English: a. initially inflectional morphology on verbs, with the plural "s" on nouns, and b. subsequently other types of morphemes. We believe that this phenomenon is purely driven by the frequency of these morphemes in the corpora. In the manually segmented portion of the Brown corpus we identified on the token level 11.3% inflectional morphemes, 6.4% derivational morphemes, and 82.1% stems. In average there are twice as many inflectional morphemes in the corpus, than derivational.

Given a very strict parameters, focusing on the description length of the grammar, our system would need long time till it would discover prefixes, not to mention infixes. By relaxing the weight of description length we can inhibit the

generation and identification of prefixing rules, however, to the cost of precision.

Given these results, the inflectional paradigms can be claimed to be extractable even with an incremental approach. As such, this means that central parts of the lexicon can be induced very early along the time line.

The existing signatures for each morpheme can be used as simple clustering criteria.[10] Clustering will separate dependent (affixes) from independent morphemes (stems). Their basic distinction is that affixes will usually have a long signature, i.e. many elements they co-occur with, as well as a high frequency, while for stems the opposite is true.[11] 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.

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.

This is but a first step to what we hope will eventually finish a platform for a detailed study of various induction algorithms and evaluation metrics.

References

E. O. Batchelder. 1997. *Computational evidence for the use of frequency information in discovery of the infant's first lexicon*. PhD dissertation, CUNY.

E. O. Batchelder. 1998. Can a computer really model cognition? A case study of six computational models of infant word discovery. In M. A. Gernsbacher and S. J. Derry, editors, *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, pages 120–125. Lawrence Erlbaum, University of Wisconsin-Madison.

L. Bloom, L. Hood, and P. Lightbown. 1974. Imitation in language development: If, when and why. *Cognitive Psychology,* 6, 380–420.

M.R. Brent and T.A. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. Cognition 61: 93-125.

H. Déjean. 1998. *Concepts et algorithmes pour la découverte des structures formelles des langues*. Doctoral dissertation, Université de Caen Basse Normandie.

K. Elghamry. 2004. *A generalized cue-based approach to the automatic acquisition of subcategorization frames*. Doctoral dissertation, Indiana University.

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

J. Fodor and V. Teller. 2000. Decoding syntactic parameters: The superparser as oracle. Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society, 136-141.

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

Z.S. Harris. 1961. *Structural linguistics*. University of Chicago Press. Chicago.

J. Grimshaw. 1981. Form, function, and the language acquisition device. In C.L. Baker and J.J. McCarthy (eds.), *The Logical Problem of Language Acquisition*. Cambridge, MA: MIT Press.

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

C.G. de Marcken. 1996. *Unsupervised Language Acquisition*. Phd dissertation, MIT.

G.F. Marcus, S. Vijayan, S. Bandi Rao, and P.M. Vishton. 1999. Rule-learning in seven-month-old infants. *Science* 283:77-80.

R. Mazuka. 1998. The Development of Language Processing Strategies: A cross-linguistic study between Japanese and English. Lawrence Erlbaum.

T.H. Mintz. 1996. *The roles of linguistic input and innate mechanisms in children's acquisition of grammatical categories.* Unpublished doctoral dissertation, University of Rochester.

S. Pinker. 1984. Language Learnability and Language Development, Harvard University Press, Cambridge, MA.

S. Pinker. 1994. *The language instinct*. New York, NY: W. Morrow and Co.

P. Schone and D. Jurafsky. 2001. *Knowledge-Free Induction of Inflectional Morphologies*. In Proceedings of NAACL-2001. Pittsburgh, PA, June 2001.

M.M. Van Zaanen and Pieter Adriaans. 2001. Comparing two unsupervised grammar induction systems: Alignment-based learning vs. EMILE. Tech. Rep. TR2001.05, University of Leeds.

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

[10] Length of the signature and frequency of each morpheme are mapped on a feature vector.

[11] This way, similar to the clustering of words into open and closed class on the basis of feature vectors, as described in Elghamry and Ćavar (2004), the morphemes can be separated into open and closed class.