University of Leeds

SCHOOL OF COMPUTING

RESEARCH REPORT SERIES

Report 2002.20

Unsupervised Grammar Inference Systems for Natural Language

by

Andrew Roberts[1] & Eric Atwell[2]

December 2002

---

[1] andyr@comp.leeds.ac.uk
[2] eric@comp.leeds.ac.uk

Abstract

In recent years there have been significant advances in the field of Unsupervised Grammar Inference (UGI) for Natural Languages such as English or Dutch. This paper presents a broad range of UGI implementations, where we can begin to see how the theory has been put in to practise. Several mature systems are emerging, built using complex models and capable of deriving natural language grammatical phenomena. The range of systems is classified into: models based on Categorial Grammar (GraSp, CLL, EMILE); Memory Based Learning models (FAMBL, RISE); Evolutionary computing models (ILM, LAgts); and string-pattern searches (ABL, GB). An objectively measurable statistical comparison of performance Of the systems reviewed is not yet feasible. However, their merits and shortfalls are discussed, as well as a look at what the future has in store for UGI.

1   Introduction

Gold's seminal paper (Gold, 1967) showed that it was theoretically impossible to extract a definitive grammar from examples of a target language, unless selected negative counterexamples were also available. Encouragingly, this theoretical hurdle has not deterred research: the natural language learning (NLL) community has witnessed rapid advances in unsupervised grammar inference.

Interesting developments have arisen from the psychological perspective of this task: research has been driven to devise psychologically plausible models of natural language acquisition.

Grammar inference is not just restricted to its classical domain of syntactic pattern recognition, with many useful functions within other levels of Natural Language Processing, such as speech processing. It has expanded in to other important areas, for example, information retrieval (Freltag, 1997; Hong and Clark, 2001) and gene analysis (Dong and Searls, 1994).

This paper focuses on recent UGI implementations. Some are pitched as UGI systems in their own right, others could be classed as solutions to sub-tasks that would be useful to language learning systems. It is worth noting that this is not a comprehensive review of every such system, but more of a snapshot of some of the interesting avenues of research being explored. Highly supervised systems, regardless of their performance, have not been included, nor have visualisation techniques that make it easier for human experts to discover grammar structure from text (Belkin and Goldsmith, 2002; Elliott et al, 2001). We also exclude systems which only infer word-classifications without attempting to learn structure, such as (Atwell, 1983; Hughes and Atwell, 1994; Roberts, 2002).

2   Categorial Grammar

A categorial grammar is a simply a grammar rather than a learning paradigm. However, CG clearly lends itself to unsupervised learning as it has been adopted

$$
\begin{array}{ccc}
\dfrac{\text{Harry}}{\text{NP}} & \dfrac{\text{eats}}{\text{S\textbackslash NP/NP}} & \dfrac{\text{apples}}{\text{NP}}
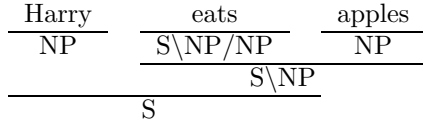\end{array}
$$

Figure 1: A simple sentence parsed in CG

as the foundation for many systems. One of the main reasons for this that the lexicon and the grammar are acquired in the same task; the psychology literature (Bates and Goodman, 1997) suggests that the two are not separate mental processes. This is a bonus for those researchers striving to produce a realistic psychological model of human language acquisition, and also for those who wish to implement simpler and more efficient algorithms for what is still a complex task.

CGs were first proposed by Ajdukiewicz (1935) and have since matured and modified. Steedman's generalised version (Steedman, 1989) serves well for a brief overview. A CG comprises of two components. Firstly, the *categorial lexicon* is essentially a dictionary which associates each word within the lexicon a syntactic and semantic category. Secondly, the *combinatory rules* provide the functional application of the grammar, and allow more complex categories to be created from the simpler ones.

$$
\begin{aligned}
S/b &\longrightarrow a \\
S\backslash a &\longrightarrow b
\end{aligned}
$$

These operators provide the freedom to transform rules, allowing you to isolate and manipulate parts that would otherwise be inaccessible (Adriaans, 1999). Fig. 1, taken from Steedman (1989), illustrates how a simple sentence is parsed.

2.1 GraSp

GraSp (Henrichsen, 2002) is a learning algorithm designed specifically for inducing grammars from large, unlabelled corpora. Its long term goal is to provide insight to the innateness debate. In this instance, the hypothesis is that there is no such linguistic innateness.

Henrichsen used a variant of the Gentzen-Lambek categorial grammar, which was enhanced with non-classical rules for isolating a residue of uninterpretable sequent elements. Empty categories are also permitted in this version which are not normally allowed in CG due to the *principle of adjacency*: combinatory rules may only apply to entities which are linguistically realised and adjacent (Steedman, 1989).

The learning algorithm begins by assigning each word type with its own unique category. The learning process applies changes in the lexicon by adding, removing and manipulating the basic categories using the CG operators (/,

\ or *). The changes are guided by a measure of disorder. $Dis(\Sigma)$ returns the number of uninterpretable atoms in the sequent Σ. The update process is iterative. GraSp monitors the measure of disorder before applying each update, and the process will halt as when the update no longer improves the disorder of the lexicon.

No quantifiable measurements of GraSp's accuracy were published. Therefore, commenting on its performance is obviously difficult. Whilst rigorous evaluation may not have taken place, GraSp clearly has many merits in that it does succeed in learning linguistic features from unlabelled corpora. Henrichsen describes the output being rich in "microparadigms and microstructure", which inter-connect to form a complex grammar.

2.2   CLL

CLL (Watkinson and Manandhar, 2001) is not only concerned with developing a computationally feasible language learner, but one that is also psychologically plausible too. Therefore, the algorithm used by CLL was designed to also make way for a model of human language learning facilities, as well as being a computational learning tool.

CLL is trying to emulate a child with respect to its acquisition of its first language. The psychology influence in the research refers mainly to the environment in which the learner learns. This deals with the type of language a child is likely to encounter and the effect of language teaching. The conclusion reached: "Hence, we have a learner that is unsupervised, positive only and does not have a teacher." Unfortunately, the algorithm was arguably built with too much 'innateness', which reduces its credibility as an unsupervised process. The provision of a complete set of lexical categories was quite justly acknowledged by the authors as being "too strong a bias to be psychologically plausible". Additionally, the algorithm is given a set of closed-class words (with categories) at the start of the learning process. Two different sizes of the initial lexicon were tried, 31 and 348.

The learning algorithm functions by taking an example sentence from a corpus, which is then parsed using a n-best probabilistic chart parser (developed from a standard stochastic CKY algorithm). This can result in a number of possible parses, of which it is then up to the parse selector to decide which one would benefit the lexicon the most. The metric which decides the 'goodness' of a parse is based on which creates the most compressive lexicon. To do this, it must also evaluate the effect of the newly modified lexicon by reparsing any examples that may be affected. Whilst it appears to be a costly approach, it does ensure the most compressive lexicon.

Watkinson and Manandbar created a relatively robust approach to evaluating their results (Watkinson and Manandhar, 2001). As the Penn Treebank corpus was the source of the text to learn, its annotation was translated into CG annotation, so that it could be compared with the output of the learning algorithm. Whilst the newly annotated corpus was considered a gold standard, it was converted automatically, and therefore liable to error. The best perfor-

mance attained by CLL was 51.9% accuracy (this is with an initial 348 word lexicon). While this performance is still relatively low, considering the difficulty of the problem, and using a complex corpus, to perform above 50% is still a recognisable achievement.

2.3 EMILE

EMILE (Adriaans, 1992, 1999) has been around for some time now. It will continue to be with us because it is a well executed algorithm and performs well and efficiently. EMILE has been updated through the years, and is currently at version 4.1 — although this latest version has been implemented by Vervoort (2000).

For EMILE to be in this section, it clearly relies on a categorial grammar. A given input set of example sentences is converted into a CG of basic categories. After applying first order explosion, each sentence is examined to discover how it can be broken up into subexpressions (using the standard CG operators). The resulting set of subexpressions is passed to an oracle. The reason for this is because EMILE uses a teacher/child metaphor. Therefore, the system can ask the oracle which ones are valid.

Any subexpressions that can be substituted into the same contexts, and still be valid are said to be of the same type. Therefore, the next step employed is to cluster to rules passed by the oracle and cluster them into types. The final phase is rule induction. The clustering has resulted in a variety of basic and complex rules, however, they tend to relate to specific types. Thus the rule induction step generalises them to general types, with the outcome being a shallow context-free grammar.

EMILE tends to produce accurate results due to the fact that it waits for enough evidence to be found before constructing grammar rules. However, according to Adriaans' calculations, in order for his system to acquire a language with 50,000 words, it would need learn from a sample of 5 million sentences. Assuming an average of 15 words per sentence, then a 75 million word corpus is required. My initial thoughts was that figure was too large. However, it is quite reasonable considering EMILE is generating a grammar to cope with 50,000 words considering the complexity of the task. It does mean that EMILE is a slow learner (compared to some other systems, such as ABL), as was concluded in van Zaanen and Adriaans (2001). Experiments conducted on the ATIS corpus produced precision of 51.6%.

3 Memory Based Learning (MBL)

The MBL paradigm attempts to discover ways in which you can abstract information from a given data set, whilst maintaining accuracy. The hope is to develop methods that perform at least equal to pure MBL (where all information is retained).

```
_ _ _ a c c e d e  X
_ _ _ a c c e l e  X
_ _ _ a c c e n t  X
_ _ _ a c c e p t  X
_ _ _ a c c e s s  X
```
→
```
                    d     e
_ _ _ a c c e  l   t   X
                    n   s
                    p
                    s
```
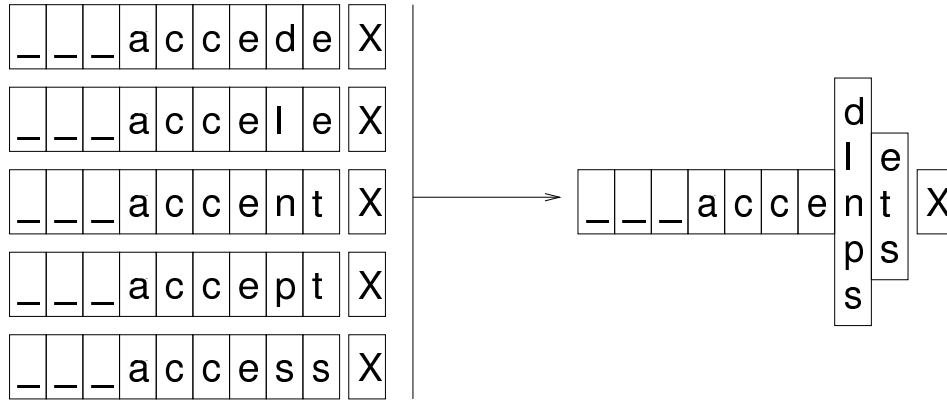
Figure 2: *Example of family creation in FAMBL taken from Van den Bosch (1999). This shows instances of grapheme-phoneme occurrences being merged into a single family expression.*

Pure MBL tends to give the best performance in analysis of unseen data, due to the very fact that it never forgets any training examples. An advantage of computerised systems is that it is often feasible to keep every instance - memory limits are seemingly infinite. So, why bother with MBL? Abstraction should result is smaller and more efficient learning models. After all, humans do not have a capacity for pure MBL, therefore, MBL should offer a more psychologically plausible approach too.

3.1 FAMBL (Family Based Learning)

Classification systems that are equipped with a forgetting facility, use it to not only perform more efficiently, but to avoid over-fitting, which can be detrimental to accuracy. However, default parameters for this process often generalise too much for learning tasks which produces poor performance (Van den Bosch, 1999).

Therefore, the key is to use careful (or weak) abstraction, whereby instances can be abstracted without doing harm (e.g., forgetting exceptions would be careless (Daelemans et al, 1999)). The way in which FAMBL achieves this is to transform the instance base into *instance families*. A family is a cluster that has been classified using k-NN (Nearest Neighbour). The instances surround a given instance are its family. Fig. 2 gives an example of how the instances are then merged into hyper-rectangles that define a family expression.

The FAMBL algorithm randomly selects instances individually, that are not a member of a family. For each one found, its family is determined and a family expression is created from those instances. It continues to generate families until the instance base doesn't contain any instances that do not belong to a family.

FAMBL has not yet been used as a full-scale grammar induction system. It has, however, been applied successfully to a variety of relevant tasks in language

learning, including morphological segmentation, base-NP chunking, PP attachment, and POS tagging. For example, the experimentation with POS resulted in an accuracy of 97.87%, and family-abstraction yielded a reduction of 75% memory compared to pure-MBL.

3.2   RISE (Rule Induction from a Set of Exemplars)

RISE (Domingos, 1995) is a multi-strategy approach, which comprises of MBL and rule induction. Rules begin as being instance specific. It then begins to generalise by looking at each rule and searching for other instances that fall within the same class. Any instances that satisfy this are merged and their rules are generalised.

In order to ensure the rules deduced are productive, RISE estimates the 'goodness' by computing its apparent accuracy using its class prediction strength with Laplace correction. Performance is constantly monitored by the algorithm and generalisation ceases if the apparent accuracy worsens.

4   Artificial Life: Evolutionary Optimisation

Nature has allowed humans to acquire the abilities to learn, understand and communicate using language by process of evolution. With that precedent, it should therefore by possible for us to apply similar techniques to create *Language Acquisition Devices*: LADs. Of course, its feasibility is the matter of debate.

LADs are already complicated, but adding an extra discipline of modelling evolution brings a new dimension of difficulty. The payoff is that once the system is setup, natural selection will take over and allow optimal language learning conditions to emerge.

4.1   Iterated Learning Model (ILM)

The idea behind Kirby's work is to take away the emphasis of biological evolution and he believes too much importance has been placed upon it. The alternative is to treat languages as adaptively evolving systems (Kirby, 2002).

If language is like an organism in its own right, then you begin to see that it has its own set of selected pressures. Humans are its host, and its method of transmission is via human communication. A successful language is one that can be learnt, understood and used, for the benefit of its hosts.

Therefore, Kirby and Hurford (2002) suggests that language is not only subject to biological natural selection, but is the result of three complex adaptive systems, as he illustrated in fig. 3:

> "There clearly are interactions: for example, biological evolution provides the platform on which learning takes place, what can be learnt influences the languages that can persist through cultural evolution, and the structure of the language of a community will

influence the selection pressures on the evolving language users."
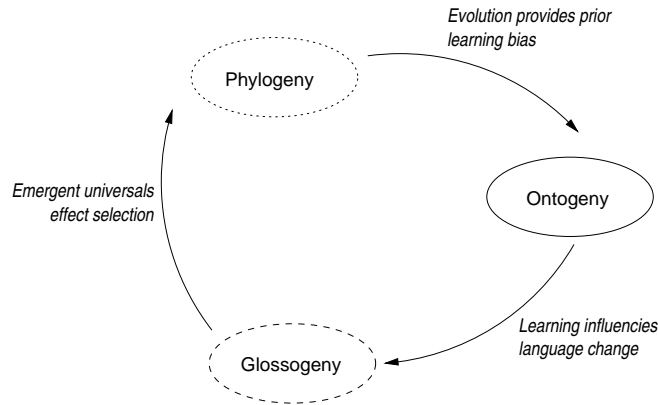(ibid)

Figure 3: *Interactions of the three adaptive systems.*

ILM is composed of four elements:

1. A meaning space

2. A signal space

3. One or more language-learning agents

4. One or more language-using adult agents

The adult agent as a set of meanings for which is must convert into signals to then be transmitted to the learner (both agents have been initialised with random parameters). The speaker makes an assumption that the hearers signal-to-meaning mapping will be approximately that of the speakers. Therefore, the principle behind transmission is the generate signals that maximise the confidence of the given meaning. Once the speaker has finished, it is then redundant, and the learner is promoted. A new learner is added and the cycle continues.

This is a somewhat simplified version of events, but the overall result is after a few hundred generations — after much randomness — meaningful structure emerges.

Kirby's research on the whole is looking at linguistic evolution rather than creating a tool for grammar inference (although one does lead to the other). And as such, there are no large scale runs of using the ILM to acquire grammar from a large corpus, and to evaluate it in any way. However, its inclusion is still relevant because it is an interesting approach - one that could be adapted to evolve grammar inference devices.

4.2 Language Agents (LAgts)

Language agents (LAgts) are born from the research carried out by Briscoe (2000). An equally intriguing approach to grammar induction through the application of alife, populations of LAgts are simulated. LAgts are language learners, generators and parsers.

Each LAgt adopts a Categorial Grammar as its underlying framework, albeit an extended version known as Generalized Categorial Grammar (GCG), (Wood, 1993). They are used to provide a relationship between the LAgt's universal grammar and the specification of a given grammar; Briscoe is clear which side of the 'innateness' fence he stands on, which is why LAgts possess a universal grammar. They are embedded in a default inheritance network, and are represented as a sequence of *p-settings*. Each setting can be encoded as *True, False* or *?* (which indicates that it is yet to be specified). They are partially ordered too, which means atomic categories can be in an arbitrary position, although more complex types, where directionality is significant, then ordering must be preserved within that type. There is also a distinction between Absolute, Default and Unset parameters.

Simulations are run to model the evolving population of LAgts. A successful interaction is generally one where a random agent generates a sentence based on its current grammar, which can be parsed by another randomly selected agent. Such a pair are said to have compatible *p-settings*. Each LAgt has a set lifespan of 10 *interaction cycles*. Between the ages of 4-10, an agent can reproduce new agents, and from aged 5 onwards, an agent stops learning and its grammar is therefore fixed. Agents older than 10 are removed from the simulation.

During their learning phase, it is possible for LAgts to alter any of their parameters that were given Default of Unset attributes from their conception. There is a cost associated with an update, which is why successful agents tend to changed by one point of their initial *p-settings*. This results in the classical strategy of inheritance where parents pass on their genes and not any acquired characteristics.

Once again, performance related information about the LAgts' actual abilities to acquire grammar accurately is difficult to extract. Much of Briscoe's work has been to experiment with the seemingly unlimited number of factors that affect evolving systems (coevolution, migration, acquisition effectiveness etc). However, languages did emerge from the learners that were described as 'full language': that is, 'close to an attested language'. Ideally, Briscoe would have elaborated as to just *how* 'close' this is. There were seven full languages available for experimentation. A language is given to one or more adult LAgts (depending on the simulation) who then communicate with the learners, and so on.

5   String Pattern Searches

Some systems do not fit conveniently into the above groups, and illustrate the greater breadth of approaches to UGI.

5.1   ABL (Alignment Based Learning)

ABL (van Zaanen, 2002) is a learning paradigm in its own right. It is based on the principle of substitutability, whereby two constituents are of the same type, then they could be substituted. Of course, the system is unsupervised, and therefore, does not know types. Thus, the principle is reversed so that if two constituents can be substituted, they are of the same type. Fig. 4 shows an example from van Zaanen and Adriaans (2001) of two segments from two sentences are declared as being of the same type.

What is a family fare
What is the payload of an African Swallow
What is (a family fare)x
What is (the payload of an African Swallow)x

Figure 4: Example of bootstrapping structure in ABL

Much complexity is added due to alignment learning phase finding constituents that overlap each other. This problem is overcome using a *selection learning* phase. This is where the ABL algorithm determines the correct (or at least the best fit) constituent using probabilistic methods. Selection is decided upon calculating the probability of the words in the overlapping constituent and its type. A Verberti-style algorithm is also used to search through all possible combinations of overlapping constituents and select the best one. ABL performs well. The computational demands of its algorithm mean that it is not suitable for large corpora (>100K sentences). However, its greedy nature means it will learn quickly. An accuracy of 62% was recorded when ABL was given the OVIS corpus to process. Versions of the ABL system have also been tested with the Penn Treebank and ATIS corpora.

5.2   GB (Grammatical Bigrams)

GB or Grammatical Bigrams were proposed by Paskin (Paskin, 2001), in the hope of creating a simple language learning model, and therefore, making the actual learning process tractable. Independence assumptions are introduced to reduce complexity, although at the same time it increases the model's bias.

The Grammatical Bigram model uses the Dependency Grammar formalism to describe the relationship between pairs of words. One word in this link is the head, and the other is its dependent. A dependency parse is a directed graph, consisting of a set of such relationships. No word can be dependent on more than one head (the root of a sentence has no dependency). Also, a word

cannot be dependent on itself, making the links acyclic. If the dependents of a head word are completely independent of each other, and their order, then this independence assumption results in a much simpler model of grammar and so the parser is spared of that complexity.

The parser is used to learn grammar from labelled corpora. However, for unsupervised learning, the EM algorithm is used to learn the optimal parameters (probabilities of dependency for a given head). Other statistics are computed using an adapted version of the Inside-Outside algorithm that works in $O(n^3)$ time.

Unfortunately, Grammatical Bigrams are suited more to the generalisation of labelled data than to unsupervised induction. When given an unlabelled corpus of Wall Street Journal articles, and its output evaluated against the annotated Wall Street Journal section of the Penn Treebank, GB yielded an accuracy of only 39.7%. Clearly, the compromise in making the model computationally efficient results in a grammar model that is still too approximate to represent the sorts of structures it sees in the input corpus.

6   Discussion

This review has attempted to analyse the range of underlying algorithms used by various approaches:

1. GraSp (Henrichsen, 2002): minimising "disorder" in CG lexicon;

2. CLL (Watkinson and Manandhar, 2001) : stochastic CKY parser optimisation from a given lexicon;

3. EMILE (Adriaans, 1999): rule induction from candidates, guided by "oracle";

4. FAMBL (Daelemans et al, 1999): weak abstraction of common subexpressions into "families";

5. RISE (Domingos, 1995): Memory Based Learning or patterns, with rule induction;

6. ILM (Kirby and Hurford, 2002): evolutionary optimisation of a language-space;

7. LAgts (Briscoe, 2000): evolutionary optimisation of language-acquisition software agents;

8. ABL (van Zaanen, 2002): rule induction from substitutable subexpressions;

9. GB (Paskin, 2001): stochastic optimisation of dependency-pair sets.

A review ought to include a comparative evaluation of the alternative approaches and systems; but we could find few objective metrics or evaluative features reported across the source literature. Most researchers appeal to a linguist's "looks good to me" evaluation (Hughes and Atwell, 1994), (Jurafsky and Martin, 2000): they demonstrate that their systems can infer some examples of grammatical constructs which seem similar those found in linguists' grammars. Unfortunately, this is a subjective qualitative assessment, and does not yield a percentage score which can be compared.

Early research on unsupervised word-class inference, clustering words into classes e.g. (Atwell, 1983; Atwell and Drakos, 1987; Finch, 1993) also appealed to "looks good to me" evaluation; but more recent research has tried to measure inferred word-classes against a human-tagged corpus (Hughes and Atwell, 1994; Roberts, 2002). Other areas of Language Engineering also try to evaluate rival systems against a "Gold Standard" human-annotated corpus; so why not Unsupervised Grammar Inference? Only four of the projects surveyed report accuracy measured against a human-parsed corpus:

1. CCL (Watkinson and Manandhar, 2001): 51.9% on Penn Treebank;

2. Grammatical Bigrams (Paskin, 2001): 39.7% on Penn Treebank;

3. EMILE (Adriaans, 1992; Vervoort, 2000): 51.6% on ATIS Corpus;

4. ABL (van Zaanen, 2002): 62% on OVIS corpus (lower with Penn and ATIS).

Even these percentage scores cannot be compared meaningfully as they are based on different alignment-measures, and used different corpora and human parsing schemes. Parsing schemes used in human-annotated treebanks can capture a variety of grammatical information, including some or all of the following (Leech et al, 1996): (a) Bracketing of segments; (b) Labelling of segments; (c) Showing dependency relations; (d) Indicating functional labels; (e) Marking sub-classification of syntactic segments; (f) Deep or 'logical' information; (g) Information about the rank of a syntactic unit; (h) Special syntactic characteristics of spoken language. In their review of corpus parsing schemes, Atwell et al (2000) conclude:

> "unlike the tagging schemes, it does not make sense to make an application-independent comparative evaluation. No single standard can be applied to all parsing projects. Even the presumed lowest common denominator, bracketing, is rejected by some corpus linguists and dependency grammarians. The guiding factor in what is included in a parsing scheme appears to be the author's theoretical persuasion or the application they have in mind."

So, an objective measure of alignment against a human-parsed "gold standard" Treebank may not be feasible or even desirable. In fact, one intriguing potential of Unsupervised Grammar Inference is that it may yield analyses

which fit the data better than traditional grammarians' parse-categories; but if we measure against an established Treebank, any such innovation will be penalised. Unsupervised Grammar Inference has potential applications with unknown languages e.g. (Atwell and Elliott, 2001), for which a high score in learning English grammar may be inappropriate.

An alternative possible metric which suggests itself is "how much has been learnt": some measure of the difference between size or scale of the initial assumption or "learning bias" (Jurafsky and Martin, 2000) and the final grammar which has been inferred. For example, Watkinson and Manandhar (2001) contrasted CCL experiments with initial lexicons of 31 and 348 words, and found that the latter yielded a larger grammar. Unfortunately, other sources did not report comparable "starting assumption" and "final grammar" metrics.

7 Future of Grammar Inference

The next big step within the UGI community is to design and develop a robust evaluation procedure. Many of the systems featured in this paper did not publish performance results of their experiments, favouring a '*looks good to me*' approach. This is certainly not an attempt to say expert linguistic evaluation is somehow inferior to an automatic, computerised approach. However, it is rather subjective, and if carried out by the author of the UGI system, likely to be partial to some bias.

An automatic evaluation tool — if designed correctly — would allow consistent comparison between rival systems. To be able to quantify performance would allow GI designers and developers to ensure that future updates actually provide greater accuracy, and quickly, so that research is not led down a dead end if the results of a system looked good, but performance was in fact degrading. However, it is clearly fraught with difficulties, which is the likely reason why many have steered clear.

With respect to UGI itself, we can look forward to great advances in the long term. The computational complexity of the algorithms will become less of a burden with optimisation and increased computational resources. We also look forward to wider applications on different datasets, as Unsupervised Grammar Inference is more widely recognised as a powerful data-mining technique.

References

Adriaans, P.W. *Language Learning from a Categorial Perspective*. Ph.D. thesis, Unversiteit van Amsterdam. 1992.

Adriaans, P.W. *Learning shallow context-free languages under simple distributions*. ILLC Report PP-1999-13, Institute for Logic, Language and Computation, Amsterdam.

Ajdukiewicz, K. *Die syntaktische Konnexität.* Studia Philosophica. 1. 1935. pp. 1-27.

Atwell, E. *Constituent likelihood Grammar*. ICAME Journal, 7, 983, pp. 34-65.

Atwell, E and Drakos, N. *Pattern recognition applied to the acquisition of a grammatical classification system from unrestricted English text*. In Maegaard, B. (Ed.), Proceedings of the Third Conference of European Chapter of the Association for Computational Linguistics, 1987, pp. 56-63.

Atwell, E, Demetriou, G, Hughes, J, Schiffrin, A, Souter, C and Wilcock, S. *A comparative evaluation of modern English corpus grammatical annotation schemes*. ICAME Journal 24, 2000, pp. 7-23.

Atwell, E. and Elliott, J. *A corpus for interstellar communication*. In Rayson, P, Wilson, A, McEnery, T, Hardie, A, and Khoja, S. (Eds.), Proceedings of CL2001: International Conference on Corpus Linguistics. UCREL Technical Paper 13, Lancaster University, 2001, pp. 31-39.

Bates, E. and Goodman, J.C. *On the Inseparability of Grammar and the Lexicon: Evidence from Acquisition, Aphasia, and Real-time Processing*. Language and Cognitive Processes, 12, 1997, pp. 507-584.

Belkin, M. and Goldsmith, J. *Using eigenvectors of the bigram graph to infer morpheme identity*. Proceedings of the Morphology/Phonology Learning Workshop of ACL-02. Association for Computational Linguistics. 2002.

Briscoe, E.J. *Grammatical Acquisition: Inductive Bias and Coevolution of Language and the Language Acquisition Device*. Language, 76(2). 2000. pp. 245-296.

Daelemans, W, Van den Bosch, A and Zavrel, J. *Forgetting exceptions is harmful in language learning*. Machine Learning, 11. 1999. pp. 11-43.

Domingos, P. *The RISE 2.0 system: A case study in multistrategy learning*. Technical Report 95-2, Department of Information and Computer Science, University of California. 1995.

Dong, S and Searls, D.B. *Gene Structure Prediction by Linguistic Methods*. Genomics. 1994.

Elliott, J, Atwell, E and Whyte, W. *Visualisation of Long Distance Grammatical Collocation Patterns in Language*. In IV2001: 5th International Conference on Information Visualisation, London, UK. 2001.

Finch, S. *Finding structure in language*. PhD thesis, Edinburgh University. 1993.

Freltag, D. *Using Grammatical Inference to Improve Precision in Information Extraction*. In Working Papers of the ICML-97 Workshop on Automata Induction, Grammatical Inference and Language Acquisition. 1997.

Gold, E.M. *Language Identification in the Limit*. Information and Control. 10. 1967. pp. 447-474.

Henrichsen, P.J. *GraSp: Grammar Learning from unlabelled speech corpora*. In: Roth, D and Van den Bosch, A. (Eds), Proceedings of CoNLL-2002, Taipei, Taiwan, 2002, pp. 22-28.

Hong, T.W and Clark, K.L. *Using Grammatical Inference to Automate Extraction from the Web*. In Principles of Data Mining and Knowledge Discovery. 2001. pp. 216-227.

Hughes, J and Atwell, E. *The automated evaluation of inferred word classifications*. In Cohn, A. (Ed), Proceedings of ECAI'94: 11th European Conference on Artificial Intelligence. John Wiley, 1994, pp535-539.

Kirby, S and Hurford, J. The emergence of linguistic structure: an overview of the iterated learning model. In: Cangelosi, A and Parisi, D (Eds.) *Simulating the Evolution of Language*. 2002. pp. 121-148.

Kirby, S. *Natural Language from Artificial Life*. Artificial Life, 8(2). 2002. pp. 185-215.

Jurafsky, D and Martin, J. *Speech and Language Processing*. Prentice-Hall, 2000.

Leech, G, Barnett, R and Kahrel, P. *EAGLES Final Report and guidelines for the syntactic annotation of corpora*. European Expert Advisory Group on Language Engineering Standards (EAGLES) Report EAG-TCWG-SASG/1.5, 1996.

Paskin, M.A. *Grammatical Bigrams*. In Dietterich, T, Becker, S, and Gharahmani, Z (eds.), Advances in Neural Information Processing Systems 14. Cambridge, MA: MIT Press. 2001.

Roberts, A. *Automatic acquisition of word classification using distributional analysis of content words with respect to function words* Technical Report, School of Computing, University of Leeds, 2002.

Steedman, M. Constituency and Coordination in a Combinatory Grammar. In: Baltin, M.R and Kroch, A.S (Eds.), *Alternative Conceptions of Phrase Structure*. University of Chicago. 1989. pp. 201-231.

Van den Bosch, A. *Careful abstraction from instance families in memory-based language learning*. Journal of Experimental and Theoretical Artificial Intelligence, 11:3, special issue on Memory-Based Language Processing, Daelemans, W, guest ed. 1999. pp. 339-368.

Van Zaanen, M and Adriaans, P.W. *Comparing Two Unsupervised Grammar Induction Systems: Alignment-Based Learning vs. EMILE*. Technical Report: TR2001.05, School of Computing, University of Leeds. 2001.

Van Zaanen, M. *Bootstrapping Structure into Language: Alignment-Based Learning*. PhD Thesis, School of Computing, University of Leeds. 2002.

Vervoort, M.R. *Games, Ealks and Grammars*. Ph.D thesis, Universiteit van Amsterdam. 2000.

Watkinson, S and Manandhar, S.*A Psychologically Plausible and Computationally Effective Approach to Learning Syntax*, CoNLL'01, the Workshop on Computational Natural Language Learning, ACL/EACL 2001.

Watkinson, S and Manandhar, S.*Translating treebank annotation for evaluation*. In: Proceedings of the Workshop on Evaluation Methodologies for Language and Dialogue Systems, ACL/EACL 2001.

Wood, M. *Categorial Grammars*. Routledge. London. 1993.