Alignment-Based Learning versus EMILE: A Comparison

Menno van Zaanen[a] Pieter Adriaans[b]

[a] University of Leeds, Woodhouse Lane, LS2 9JT Leeds, UK
[b] University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands

Abstract

In this paper we set out to compare two unsupervised grammar induction systems: Alignment-Based Learning (ABL) and EMILE. Both are motivated from a different background and with a different goal. ABL starts out from the linguistic notion of substitutability [14] aiming to learn a maximum number of correct constituents. On the other hand, EMILE stems from a mathematically sound theory of substitution classes which makes it possible to prove that EMILE's learned string language converges to the string language from which samples are taken.

Both systems will be described briefly, followed by a theoretical comparison between the two. In addition to this, the two systems are applied to two different corpora, the English ATIS corpus and the Dutch OVIS corpus. The properties of the systems as described in the theoretical comparison are reflected in the results on the two corpora.

1 Introduction

In this paper we study unsupervised grammar induction from text. Existing grammar learning methods can be grouped (like other learning methods) into supervised and unsupervised methods, and methods that only learn from positive data versus methods that use complete information (positive as well as negative). Unsupervised methods only use plain (or pre-tagged) sentences, while supervised methods are first initialised with structured sentences.

Since the landmark paper of Gold [12] we know that only a very limited class of languages can be identified from positive data alone. This fact, together with the observation that young children are rarely corrected by their parents when they start to learn their native language, has induced a widespread interest in theoretical as well as experimental research into constraints that make certain more interesting classes of languages (regular, context-free, context-sensitive) learnable.

Apart from this theoretical motivation for our research, there is also a more practical goal. In practice, supervised methods generate better results, since they can adapt their output to the structured examples from the initialisation phase, whereas unsupervised methods do not have any idea what the output should look like. Although unsupervised methods perform worse than supervised methods, unsupervised methods are necessary for the time-consuming and costly creation of treebanks for which no corpus nor grammar yet exists.

Grammar induction tools could also be used in situations where it is in principle impossible to use annotated corpora such as deciphering texts in an unknown language,

or the analysis of non-linguistic data sets such as musical sequences or sequences of error messages. From these facts it is clear that there is a strong practical motivation to develop unsupervised grammar induction algorithms.

The main goal in this paper is the introduction and comparison of two approaches to unsupervised grammar induction: EMILE 4.1 [2] and ABL (Alignment-Based Learning) [25].

2 Previous Work

Early attempts try to induce context-free grammars from text but in 1967 Gold [12] proved that this is in general impossible. Gold's concept of identification in the limit has been amended with the notion of PAC learning (probably approximately correct learning [24]) in the eighties and PACS learning (PAC learning under simple distributions [18]) in the nineties of the last century.

Various authors have worked on the unsupervised grammar induction problem. We will give a (very) short overview here.

There have been several approaches from the machine learning perspective, using for example genetic algorithms [16] or neural networks [15]. Memory based learning (MBL) keeps track of the possible contexts and assigns word types based on that information [10]. [19] contains a method that finds constituent boundaries using mutual information values of the part of speech n-grams within a sentence and in [22] a method is presented that bootstraps syntactic categories using distributional information.

Algorithms that use the minimum description length (MDL) principle build grammars that describe the input sentences using the minimal number of bits. This idea stems from the information theory. Examples of these systems can be found in [11, 13] and [28].

The system in [28] performs a heuristic search while creating and merging symbols directed by an evaluation function. Similarly, [9] describes an algorithm that uses a cost function that can be used to direct search for a grammar. [23] contains a more recent grammar induction method that merges elements of models using a Bayesian framework. In [8] a Bayesian grammar induction method is presented, which is followed by a post-pass using the inside-outside algorithm [3, 17], while the algorithm in [21] applies the inside-outside algorithm to a partially structured corpus.

3 Description of the systems

In this section two grammar induction systems are described. We start out with an explanation of EMILE, followed by an overview of ABL. After describing the two systems, the differences and similarities are discussed.

3.1 The EMILE approach

The general idea behind EMILE is the notion of identification of substitution classes by means of clustering. If a language has a context-free grammar then expressions that are generated from the same non-terminal can be substituted for each other in each context where that non-terminal is a valid constituent. Conversely, if we have a sufficiently rich sample from this language then one expects to find classes of expressions that cluster together in comparable contexts. [2]

EMILE consists of two main stages. In the *clustering phase* all possible contexts and expressions of a sample are gathered in a matrix. Starting with random seeds, clusters of contexts and expressions, that form correct sentences, are created.[1] If a group of contexts and expressions cluster together, they receive a type label. This creates a set of proto-rules.

What is a family fare
What is coach fare for flight 1943

What is {*a family fare* | *coach fare for flight 1943*}

Figure 1: Example clustering expressions in EMILE

In figure 1 we illustrate how EMILE finds clusters and contexts. Actually, this type of clustering can be seen as a form of text compression [13]. The context *What is* occurs with the expressions *a family fare* and *coach fare for flight 1943*. This introduces the proto-rule: *[0]⇒ What is [19]* and *[19] ⇒ a family fare*. The same principle is used for the other expression. The sentence type *[0]* can be rewritten as *What is* concatenated to an expression of type *[19]*.

This finding gives rise to the hypothesis (possibly unjustified) that these two expressions are generated from the same non-terminal. If we find enough traces of a whole group of expressions in a whole group of contexts the probability of this hypothesis grows.

For a sentence of length n the maximal number of different contexts and expressions is $n(n+1)/2$. The complexity of a routine that clusters all contexts and expressions is polynomial in the number of contexts and expressions.

In the *rule induction phase* a concise method for rule creation is used. The matrix is checked for characteristic expressions and new rules are derived by substitution of types for characteristic sub-expressions in typed expressions.

Suppose for instance that the expression *a family fare* is characteristic for type *[19]*. We may then form the rule *[201] ⇒ the price of [19] to Rome* from the rule *[201] ⇒ the price of a family fare to Rome*.

This paper uses the EMILE 4.1 algorithm [27] which is a successor to EMILE 3.0 [1]. It uses a more efficient rule induction algorithm, albeit less sound.

3.2 Alignment-Based Learning

Alignment-Based Learning (ABL) is based on Harris's idea of substitutability [14], which states that if two constituents are of the same type then they can be substituted by each other.

ABL searches for constituents by using a reversed version of Harris's implication: *if parts of sentences can be substituted by each other then they are constituents of the same type.*

(Book Delta 128) from Dallas to Boston
(Give me (all flights) from Dallas to Boston)
Give me (help on classes)

Figure 2: Overlapping constituents

The implementation of the ABL system consists of two distinct steps. (See [25, 26] for a more elaborate overview.)

[1] A set of parameters and thresholds determines the significance of, and the amount of noise in the clusters.

The *alignment learning phase* finds possible constituents by aligning pairs of sentences to each other. Groups of words that are *different* in both sentences are considered possible constituents.

Figure 2 illustrates how ABL finds constituents. In the alignment learning phase, the first sentence is aligned to the second one first. The underlined words indicate similar parts in the sentences. The dissimilar parts (*Book Delta 128* and *Give me all flights*) are now considered constituents.

When the second sentence is aligned to the third, it receives another constituent, which overlaps with the older constituent.

Since the underlying grammar is assumed to be context-free, overlapping constituents are unwanted. The *selection learning phase* eliminates overlapping constituents after the alignment learning phase has finished. The best constituents are selected based on a statistical evaluation function.

The probability for each constituent is computed (C is the set of constituents):

$$P(c|root(c) = r) = \frac{|c' \in C : yield(c') = yield(c) \wedge root(c') = r|}{|c'' \in C : root(c'') = r|}$$

Using these probabilities, the probabilities of all combinations of constituents are computed. Instead of computing the combination probability by multiplying probabilities (as in SCFGs [6]), the geometric mean is used to reduce the "trashing" effect [7]. The set of non-overlapping constituents with the highest probability is chosen to be correct.

3.3 Theoretical comparison between ABL and EMILE

While ABL directly (and greedily) structures sentences, EMILE tries to find grammar rules[2] and it only finds a grammar rule when enough evidence is found. This duality is actually the main difference of the two systems. This results in ABL being slower (taking more time and thus only applicable to smaller corpora) in contrast to EMILE, which is developed to work on much larger corpora (say over 100,000 sentences).

The inner working of the algorithms is completely different. EMILE finds a grammar rule when enough information is found to support the rule. In contrast, ABL stores all possible constituents and then after all possible constituents are found, the "best" constituents are selected.

Given the two sentences *What is a family fare* and *What is coach fare for flight 1943* ABL will align *What is* and *fare*, whereas EMILE will try to cluster *coach fare for flight 1943* and *a family fare* giving a higher probability when it has more proof (i.e. when it sees more occurrences).

One other interesting feature is that both systems can learn recursive structures. To our knowledge, no other systems can do this in a completely unsupervised way.

4 Test Environment

The two systems have been tested on two treebanks: the *Air Traffic Information System (ATIS)* treebank [20] and the *Openbaar Vervoer Informatie Systeem*[3] *(OVIS)* treebank [5].

[2] Using these grammar rules the original corpus is parsed and the resulting structured sentences are used for evaluation.

[3] "Openbaar Vervoer Informatie Systeem" stands for "Public Transport Information System".

The Penn treebank ATIS is an English treebank consisting of 716 sentences. The larger OVIS treebank contains exactly 10,000 Dutch sentences. Removing all sentences consisting of only one word results in a corpus of 6,797 sentences. The sentences of both corpora contain mainly imperatives and questions.

To evaluate the systems, the sentences from the treebanks are stripped of their structure and the plain sentences are fed to the grammar induction systems. The generated treebanks are then compared to the original treebanks.

To compare the treebanks, we use the PARSEVAL measures [4]. The commonly used EVALB program is used to compute the PARSEVAL measures and the EVALB measures. Additionally, the *F-score* is computed.

Note that ABL generates empty constituents, but EMILE does not. To simplify evaluation, we have removed the empty constituents from all treebanks.

To our knowledge, there is no "standard" evaluation test bench for unsupervised grammar induction systems. Instead, we have tried to use as many "standard" components used by others to evaluate their system. This justifies the use of the ATIS treebank, the PARSEVAL measures and the EVALB program.

Instead of testing the unsupervised methods on English text only, we have chosen to use the Dutch OVIS treebank as well. This allows us to see how well the systems perform on a less(er) known language. In our view, unsupervised methods are best used in fields where no structured corpora yet exist (since supervised systems cannot be used there). In addition, it is roughly ten times as large as the ATIS treebank.

Using these components unfortunately does not give results that can be compared to other existing results directly. Hopefully, people evaluating their unsupervised grammar will take an approach similar to ours making further comparison possible.

Table 1: Results: UR=unlabelled recall, UP=unlabelled precision, F=F-score

		UR		UP		F	
ATIS	EMILE	16.814	(0.687)	**51.588**	(2.705)	25.351	(1.002)
	ABL	**35.564**	(0.020)	43.640	(0.023)	**39.189**	(0.021)
OVIS	EMILE	36.893	(0.769)	49.932	(1.961)	41.433	(3.213)
	ABL	**61.536**	(0.007)	**61.956**	(0.008)	**61.745**	(0.007)

Table 2: Results: CB=average crossing brackets, 0 CB=no crossing brackets, $\leq$2 CB=two of fewer crossing brackets

		CB		0 CB		$\leq$2 CB	
ATIS	EMILE	**0.841**	(0.108)	**47.362**	(4.765)	**93.408**	(2.195)
	ABL	2.120	(0.000)	29.050	(0.000)	64.996	(0.069)
OVIS	EMILE	**0.714**	(0.053)	**56.897**	(2.475)	**93.189**	(0.695)
	ABL	1.220	(0.000)	42.046	(0.017)	85.210	(0.000)

5 Results

An overview of the results of both systems on the two treebanks can be found in tables 1 and 2. The figures in the tables represent the mean values of the metric followed by their standard deviations (in brackets). Each result is computed using ten fold cross validation.

Note that the OVIS and ATIS corpora are certainly not characteristic for the underlying grammars. It is therefore impossible to learn a perfect grammar for these corpora from the data in the corpora.[4]

[4]It is our hypothesis that one needs a corpus of at least 50.000.000 sentences to get an acceptable

As can be seen from the results, EMILE only finds structure when enough evidence has been seen. In other words, it does not learn much structure on small corpora (hence the low recall), but the learned constituents are reasonably good (higher precision than recall and also a high 0 CB and $\leq$2 CB measures and a low average crossing).[5] On the other hand, ABL learns faster, resulting in a higher recall and a slightly lower precision on the ATIS corpus. On the OVIS, ABL has a high recall and precision, but the CB, 0 CB and $\leq$2 CB measures are still lower than EMILE's. This indicates that ABL learns greedily, but ABL also tends to learn some incorrect constituents.

Remember that these measures were actually designed to evaluate *supervised* methods. These results cannot be compared to the results of supervised systems under any circumstances. Unsupervised learning of structure is a much harder problem to solve, since the systems do not have a clue whatsoever what the resulting structure should look like.

6 Conclusion

First of all, the results of the system may seem low compared to the results of supervised systems, but this does not mean that unsupervised methods are not useful. Supervised methods need structured corpora to train or initialise, but these resources are often not readily available (for example when analysing unknown languages). Since unsupervised methods like EMILE or ABL do not need these resources, they can still be used, which is a major advantage in many fields.

In general ABL greedily searches for constituents, while EMILE is much more cautious. EMILE needs to have a certain support in the context-expression matrix for the construction of a new type, while ABL only needs to identify expressions in a pair of sentences to learn structure.

In texts with length of the OVIS and ATIS corpora there is not enough information for the EMILE approach to converge to a grammar. In general EMILE is much faster and less greedy than ABL. For large corpora (say >100K sentences) ABL is currently not feasible, while this is well within the possibilities of EMILE.[6]

Acknowledgements

We would like to thank Marco Vervoort for his work in generating the EMILE results on both the ATIS and OVIS corpus.

References

[1] Pieter W. Adriaans. Learning shallow context-free languages under simple distributions. Technical Report PP-1999-13, Institute for Logic, Language and Computation (ILLC), Amsterdam, the Netherlands, 1999.

[2] Pieter Willem Adriaans. *Language Learning from a Categorial Perspective.* PhD thesis, University of Amsterdam, Amsterdam, the Netherlands, November 1992.

grammar of the English language on the basis of the EMILE algorithm.

[5] EMILE first has to learn a grammar and then parses the sentences, while ABL produces structured sentences directly. The current parser of EMILE is non-probabilistic.

[6] However, to date no sufficiently large treebanks exists.

[3] J.K. Baker. Trainable grammars for speech recognition. In J.J. Wolf and D.H. Klatt, editors, *Speech Communication Papers for the Ninety-seventh Meeting of the Acoustical Society of America*, pages 547–550, 1979.

[4] E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of a Workshop—Speech and Natural Language*, pages 306–311, February 19–22 1991.

[5] R. Bonnema, R. Bod, and R. Scha. A DOP model for semantic interpretation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and the 8th Meeting of the European Chapter of the Association for Computational Linguistics (EACL); Madrid, Spain*, pages 159–167, New Brunswick:NJ, USA, July 1997. Association for Computational Linguistics (ACL), Association for Computational Linguistics (ACL).

[6] T. Booth. Probabilistic representation of formal languages. In *Conference Record of 1969 Tenth Annual Symposium on Switching and Automata Theory*, pages 74–81, 1969.

[7] Sharon A. Caraballo and Eugene Charniak. New figures of merit for best-first probabilistic chart parsing. *Computational Linguistics*, 24(2):275–298, 1998.

[8] Stanley F. Chen. Bayesian grammar induction for language modeling. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 228–235, New Brunswick:NJ, USA, 1995. Association for Computational Linguistics (ACL), Association for Computational Linguistics (ACL).

[9] Craig M. Cook, Azriel Rosenfeld, and Alan R. Aronson. Grammatical inference by hill climbing. *Informational Sciences*, 10:59–80, 1976.

[10] Walter Daelemans. Memory-based lexical acquisition and processing. In P. Steffens, editor, *Machine Translation and the Lexicon*, volume 898 of *Lecture Notes in Artificial Intelligence*, pages 85–98. Springer-Verlag, Berlin Heidelberg, Germany, 1995.

[11] Carl G. de Marcken. *Unsupervised Language Acquisition.* PhD thesis, Massachusetts Institute of Technology, Cambridge:MA, USA, September 1996.

[12] E.M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[13] Peter Grünwald. A minimum description length approach to grammar inference. In G. Scheler, S. Wernter, and E. Riloff, editors, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language*, volume 1004 of *Lecture Notes in AI*, pages 203–216. Springer-Verlag, Berlin Heidelberg, Germany, 1994.

[14] Zellig S. Harris. *Structural Linguistics.* University of Chicago Press, Chicago:IL, USA and London, UK, 7th (1966) edition, 1951. Formerly Entitled: Methods in Structural Linguistics.

[15] T. Honkela, V. Pulkki, and T. Kohonen. Contextual relations of words in grimm tales, analyzed by self-organizing map. In *Proceedings of the International Conference on Artificial Neural Networks*, 1995.

[16] M.M. Lankhorts. Grammatical inference with a genetic algorithm. In *Proceedings of the 1994 EUROSIM Conference on Massively Parallel Applications and Development*, pages 423–430, 1994.

[17] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4(35–56), 1990.

[18] M. Li and P. Vitányi. Learning simple concepts under simple distributions. *SIAM Journal of Computing*, pages 911–935, 1991.

[19] David M. Magerman and Mitchell P. Marcus. Parsing a natural language using mutual information statistics. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 984–989. American Association for Artificial Intelligence (AAAI), 1990.

[20] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[21] Fernando Pereira and Yves Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL); Newark:Delaware, USA*, pages 128–135, New Brunswick:NJ, USA, 1992. Association for Computational Linguistics (ACL), Association for Computational Linguistics (ACL).

[22] Martin Redington, Nick Chater, and Steven Finch. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, 22(4):425–469, 1998.

[23] Andreas Stolcke and Stephen Omohundro. Inducing probabilistic grammars by bayesian model merging. In *Proceedings of the Second International Conference on Grammar Inference and Applications; Alicante, Spain*, pages 106–118, 1994.

[24] L.G. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27(11):1134–1142, 1984.

[25] Menno van Zaanen. ABL: Alignment-Based Learning. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING); Saarbrücken, Germany*, pages 961–967. Association for Computational Linguistics (ACL), July 31–August 4 2000.

[26] Menno van Zaanen. Bootstrapping syntax and recursion using Alignment-Based Learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1063–1070, Stanford:CA, USA, June 29–July 2 2000. Stanford University.

[27] Marco R. Vervoort. *Games, Walks and Grammars.* PhD thesis, University of Amsterdam, Amsterdam, the Netherlands, September 2000.

[28] J. Gerard Wolff. Learning syntax and meanings through optimization and distributional analysis. In Y. Levy, I.M. Schlesinger, and M.D.S. Braine, editors, *Categories and Processes in Language Acquisition*, chapter 7, pages 179–215. Lawrence Erlbaum, Hillsdale:NJ, USA, 1988.