ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-293

Unsupervised Grammar Induction Using a Parent Based Constituent Context Model

Seyed Abolghasem Mirroshandel and **Gholamreza Ghassem-Sani**[1]

Abstract. Grammar induction is one of attractive research areas of natural language processing. Since both supervised and to some extent semi-supervised grammar induction methods require large treebanks, and for many languages, such treebanks do not currently exist, we focused our attention on unsupervised approaches. Constituent Context Model (CCM) seems to be the state of the art in unsupervised grammar induction. In this paper, we show that the performance of CCM in free word order languages (FWOLs) such as Persian is inferior to that of fixed order languages such as English. We also introduce a novel approach, called parent-based constituent context model (PCCM), and show that by using some history notion of context and constituent information of each span's parent, the performance of CCM, especially in dealing with FWOLs, can be significantly improved.

1 INTRODUCTION

Based on the type of corpora that different unsupervised grammar induction methods use, these methods are divided into three major categories [1]: supervised, unsupervised, and semi-supervised. Supervised methods normally rely on the correct parse of training sentences via a full-parsed and tagged treebank. Semi-supervised methods use less supervision information than supervised ones. Unsupervised methods rely only on tagged sentences without any bracketing.

Although current supervised methods highly outperform unsupervised methods, there are important motives to continue the work on unsupervised methods [2, 3, 4], because producing the necessary training data (corpora) of supervised methods is a time consuming, hard, and expensive work. Besides, it is very difficult to adapt supervised methods for new tasks, languages, and domains. Consequently, it is the corpus availability that directs the research in this area. Not only unsupervised methods do not need such training data, but also they can be used in many applications: in primary phases of constructing large treebanks, in language modeling, and in some NLP research areas that do not require an exact grammar of sentences.

Constituent Context Model (CCM) [2, 3] seems to be the state of the art in unsupervised grammar induction. In this paper, we show that the performance of CCM in free word order languages (FWOLs) such as Persian is inferior to that of fixed order languages such as English. We also introduce a novel approach, called parent-based constituent context model (PCCM), and show that by using some history notion of context and constituent information of each span's parent, the performance of CCM, especially in dealing with FWOLs, can be significantly improved.

The remainder of the paper is arranged as follows: Section 2 is about previous approaches to unsupervised grammar induction. Section 3 explains original constituent context model, and our improved method. Section 4 demonstrates the evaluation of the proposed algorithm. Finally, section 5 includes paper's conclusion.

2 PREVIOUS WORKS

There is a lot of ongoing research on unsupervised grammar induction (UGI) methods. These methods can be divided into three groups: 1) Likelihood based; 2) Compression based, and 3) Distribution based. These groups are discussed in the next three sub-sections.

2.1 Likelihood Based Methods

This group of UGI selects maximum likelihood model using a probabilistic context free grammar (PCFG). Likelihood based methods, also known as inside-outside (IO), work using the expectation maximization (EM) algorithm [5, 6, 7]. There are some researches in amendment of these methods [8]. IO algorithms produce a grammar in Chomsky normal form (CNF). These algorithms often converge toward a local optimum state by iteratively re-estimating the probabilities in a manner that maximizes the likelihood of the training corpus, given the grammar. They would nearly always converge to a linguistically improper grammar [9]. These methods have also been implemented using genetic algorithms [10]. One algorithm in this group, which added the parent of each non-terminal as the conditioning information to the IO grammar rules, is history-based IO (HIO) [11]. In HIO, grammar rules are in CNF, but HIO replaces ordinary CNF ($X \rightarrow AB$) with a pseudo CNF which adds the parent of each non-terminal in the left hand side of the rules ($X, Parent(X) \rightarrow AB$). HIO showed some improvement in UGI, especially in Persian [11].

2.2 Compression Based Methods

These methods work using the minimum description length (MDL) principle. Several methods based on this approach have been proposed, none of which showed a satisfactory result [6]. One of these methods uses the Bayesian model selection criterion for hidden Markov model (HMM) and PCFGs, but it only works in small and artificial languages [12]. Another method [13] works on regular languages rather than context free languages. Since the only factor with which these methods work is the compression of

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, emails: {mirroshandel@ce.sharif.edu, sani@sharif.edu}

the most frequent sequence of tags (sequences with high mutual information), their results are not satisfactory. For example, the sequences IN DT and DT NN have high mutual information in both English and Persian languages. In Penn treebank, the sequence IN DT and DT NN have 1.3675 and 1.266 pointwise mutual information. Thus these methods incorrectly divide the sequence IN DT NN into IN DT and DT NN.

2.3 Distribution Based Methods

These methods are based on a simple idea: the sequences of words or tags that construct the same constituents appear in analogous contexts. There are several methods based on this approach: Distributional clustering was one of the first attempts in this regard [14]. This method used a conditional entropy measure to identify constituents. Another method in this group used Kullback-Leibler (KL) divergence measure of contexts to extract probable rules [15]. In this algorithm, only sequences with two tags are compared with sequences with a single tag. Thus, the method cannot induce sequences that do not correspond to a single tag sequence.

There are other approaches that use distributional clustering in finding similar tag sequences appeared in similar contexts [16]. These methods generate some linguistically plausible clusters, but at the same time find many implausible ones. Therefore, they cannot induce acceptable grammars.

Context distributional clustering (CDC) is another distributed method that uses distributional clustering of sequence of tags. However, in CDC, only clusters that satisfy the mutual information (MI) measure are regarded as valid clusters. In other words, the MI measure is used to prune linguistically implausible clusters. CDC also incorporates the MDL to extract grammars [5, 6].

At present, the most successful UGI algorithm is the so-called constituent context model (CCM). CCM is a parameter search algorithm [4] that, by using some distributional information in an EM method, can induce a grammar. Section 3.1 explains this algorithm in more detail.

All these methods use local distributional context. However, there are two techniques that use the whole sentence as the context: Alignment-Based Learning (ABL) and EMILE [17, 18]. These techniques look for minimal pairs. In fact, they search for pairs of sentences that, except for a particular phrase, look the same. These two algorithms have reasonable results only in restricted and artificial languages [17].

3 PARENT-BASED CCM

In this section, a novel method based on CCM is introduced, which improves CCM's performance, especially in dealing with free word order languages (FWOL). Before describing the new method, CCM is briefly explained in next sub-section.

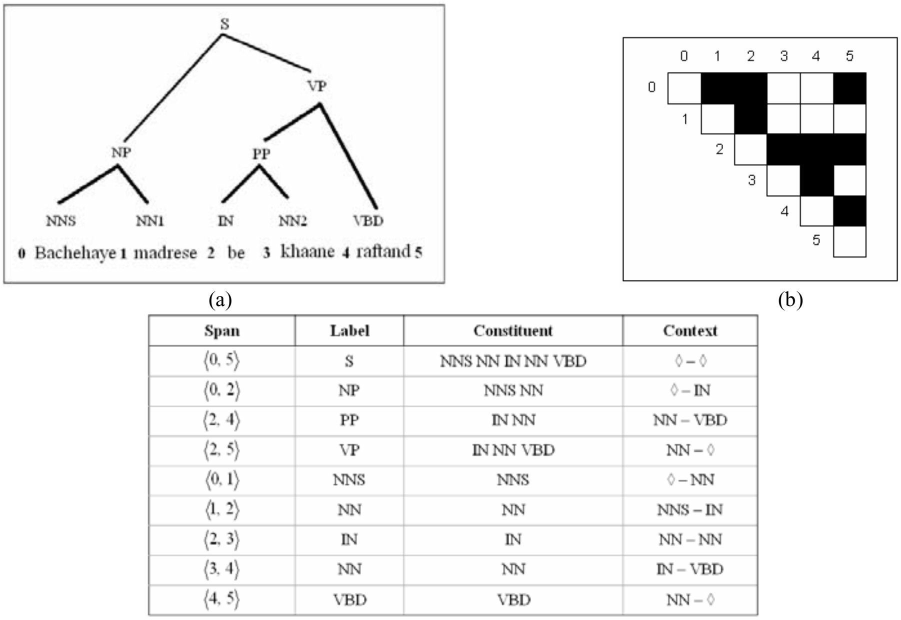
3.1 CCM

As mentioned before, CCM is a distribution based method and works on the basis of a weakened version of the classic linguistic constituency tests [19]: constituents occur in their contexts. CCM is designed to transmit the constituency of a sequence (it works with part-of-speech tag sequences) directly to its context, which is intended to pressure new sequence in that context. This pressure directs a new sequence to be parsed as a constituent in the following step. In fact, this method is a distributional clustering with no-overlap constraint.

3.1.1 Constituents and Contexts

In CCM, all sequences of tags, i.e. spans, are modeled, regardless of being constituent or non-constituent. Contexts in CCM are local linear contexts, which means that context of a word is the pair of words immediately adjacent to its left and right. For example, in the sentence "Factory payrolls fell in September", the word "payrolls" occurs in the context "Factory–fell".

A bracketing of a sentence is a boolean matrix where a true element indicates that the related span is a constituent, and conversely a false element corresponds to a non-constituent (called distituent).



(a) (b)

Span	Label	Constituent	Context
⟨0, 5⟩	S	NNS NN IN NN VBD	◊ – ◊
⟨0, 2⟩	NP	NNS NN	◊ – IN
⟨2, 4⟩	PP	IN NN	NN – VBD
⟨2, 5⟩	VP	IN NN VBD	NN – ◊
⟨0, 1⟩	NNS	NNS	◊ – NN
⟨1, 2⟩	NN	NN	NNS – IN
⟨2, 3⟩	IN	IN	NN – NN
⟨3, 4⟩	NN	NN	IN – VBD
⟨4, 5⟩	VBD	VBD	NN – ◊

(c)

Figure 1. A parse tree for the Persian sentence (a), Its related bracketing (b), and the constituents and context associated with the bracketing (c).

Figure 1 demonstrates a parse tree of a sentence in Persian language, its related bracketing, and constituent-contexts of the parse tree. Representation of words in the sentence is based on [20].

A bracketing "B" is non-crossing if at most one of the two crossing brackets is a constituent in "B". A non-crossing bracketing that satisfies these rules is a tree-equivalent bracketing: 1) all unit spans (i.e., spans including just one word) of a sentence are constituent; 2) the span containing full sentence is constituent, and 3) all zero size spans of a sentence are distituent. If a bracketing corresponds to a binary tree, then the bracketing is binary, too.

The performance of CCM depends on two simple properties: 1) only binary bracketings are valid, and 2) constituents occur in constituent contexts. It has been shown that without the first assumption, CCM cannot produce valuable results [2, 3].

The generative CCM over sentences S has two steps. First, according to some distribution *P(B)*, a bracketing *B* is chosen, and then given that bracketing, the corresponding sentence is generated [4]:

$$P(S,\ B) = P(B)P(S|B) \tag{1}$$

Contexts and constituents are independent. These are generated by using the following equation:

$$\begin{aligned} P(S|B) &= \prod_{\langle i \quad j\rangle \in spans(S)} P(\alpha_{ij}, x_{ij} \mid B_{ij}) \\ &= \prod_{\langle i \quad j\rangle} P(\alpha_{ij} \mid B_{ij}) P(x_{ij} \mid B_{ij}) \end{aligned} \tag{2}$$

$$= \prod_{\langle i \quad j\rangle \in Tree(S)} P(\alpha_{ij} \mid true) P(x_{ij} \mid true) \prod_{\langle i \quad j\rangle \notin Tree(S)} P(\alpha_{ij} \mid false) P(x_{ij} \mid false),$$

where α_{ij} are spans, x_{ij} are contexts, $P(\alpha_{ij} | true)$ is the conditional probability of constituency of α_{ij} when B_{ij} is true, and $P(\alpha_{ij} \mid false)$ is the same probability when B_{ij} is false. In a similar manner, $P(x_{ij} | true)$ and $P(x_{ij} | false)$ can be defined for x_{ij}. The marginal probability of sentence S is:

$$P(S) = \sum_{All\ possile\ bracekting\ \mathbf{B}} P(B)P(S|B) \tag{3}$$

For inducing a grammar, CCM runs the EM algorithm on this model [4]. In the EM algorithm, brackets B are observed, and sentences S are hidden (unobserved) random variables.

3.1.2 The Induction Algorithm

As mentioned before, in CCM, only binary bracketings are valid, so all binary bracketings are equally likely. The EM algorithm includes the following two steps:

E-Step: Fix current θ and obtain the conditional bracketing likelihoods $P(B \mid S, \theta)$.

M-Step: Find θ' that maximizes $\sum_{B} P(B \mid S, \theta) \log P(S, B \mid \theta')$ with fixed θ.

In estimating parameters by the EM algorithm, the computational bottleneck is the E-step, where we must calculate posterior expectations of various tree configurations according to a fixed parameter vector θ. This problem can be fixed by a cubic dynamic program similar to the inside-outside algorithm [4].

3.1.3 Weakness in Free Word Order Languages

In addition to CCM weakness in dealing with long sentences, its performance further degrades when dealing with FWOLs.

In linguistic typology, the order in which words appear in sentences is called the word order. In FWOLs, the orders of some or all of the words in many sentences are not important, and they can freely appear in different places of the sentences.

As described in 3.1, CCM uses span type counts for validity discrimination of constituents and their contexts. Since in FWOLs, words appear in optional places, each span type is divided into a number of different span types. This property decreases the count of each span type. Consequently, there would be less information about pattern of constituents and their contexts available during parsing. CCM was applied to Persian, which is a rather FWOL, and it was shown that CCM's performance is reduced when dealing with such languages [21]. In the next section, we demonstrate how CCM performance can be improved by using parent information of constituents and contexts.

3.2 Parent-based CCM (PCCM)

In this section, we introduce a new model called parent-based constituent context model (PCCM), in which spans' parent information is employed to improve CCM performance especially in FWOLs.

3.2.1 Adding Parent Information

As described in section 3.1, in CCM, the context and constituent probabilities of each span are computed, and then used in grammar induction. PCCM takes advantage of two types of supplementary probabilities. The first type is the conditional constituent probability of every span, given span's parent. The second type is the conditional context probability of each span's context, given its parent's context.

For example, in the sentence of figure 1, we calculate the two following probabilities for the span *"IN NN2"*: constituent probability of span *"IN NN2"* by considering constituent probability of span *"IN NN2 VBD"*, as the parent span of *"IN NN2"*, ($P_{constituency}(IN\ \ NN2 \mid Parent(IN\ \ NN2\ \ VBD))$) and context probability of pair *"NN1-VBD"* by considering context probability of pair *"NN1-◊"*, as the parent context of *"NN1-VBD"*, ($P_{context}(NN1 - VBD \mid Parent(NN1 - \lozenge))$).

3.2.2 The Induction Algorithm

The employed induction algorithm is analogous to that of CCM. We combine extracted probabilities of original CCM with the new probabilities. The final probabilities will be used for grammar induction. Here like CCM, only binary bracketings are valid and we employ EM algorithm in a similar manner to CCM.

The difference between CCM and PCCM is in the definition and usage of θ parameters. In CCM, θ is only the context and constituent probability of each span of the sentences, but in PCCM, θ parameter is the context and constituent probability of each span of the sentences (original CCM) and the context and constituent probability of each span by considering the span's parent (PCCM).

In estimating parameters with EM algorithm, the computational bottleneck is the E-step, where we must calculate posterior expectations of various tree configurations according to a fixed parameter vector θ. This problem can be fixed by using dynamic programming. The only difficulty is that dynamic programming works in a bottom up manner, and we cannot get any knowledge about parents of spans. To tackle this problem, we used a memoization technique. Memoization works in a top down manner with an analogous order to that of dynamic programming. We also used relative frequency estimates for setting θ' of the M-step. It is worth noting that, due to the usage of parent information, the smoothing task in PCCM is even more important than in CCM.

It has been shown that the time complexity of CCM is $O(n^3)$, and its space requirement is $O(n^2)$ [4]. The time complexity of PCCM is the same as CCM [22]; however, its space requirement is $O(n^3)$, due to the need to store the extra information regarding parents of contexts and constituents.

4 EXPERIMENTAL RESULTS

In this section, we first give a brief description of Persian as an FWOL, and then describe the results of applying PCCM to two different corpuses of both English and Persian.

4.1 Persian Language

Persian is the native language of approximately one hundred million people, and is spoken in different countries such as Iran, Afghanistan, and Tajikistan. This language belongs to Indo branch of the Indo-European language family.

Normally the structure of declarative sentences in Persian is *"(S) (PP) (O) V"*. Parentheses in this structure represent optional components, i.e. subjects, prepositional phrases, and objects. This language has high potential to be categorized in the FWOLs, especially in the preposition adjunction and complements. For example, adverbs can occur anywhere in the sentences without any change in the meaning [23].

4.2 Experiments

We applied PCCM to both English and Persian. In English, we used WSJ-10 corpus with sentences of less than 11 words and ATIS corpus. Using the ten-fold cross validation method, the results were evaluated by measuring unlabeled precision (UP), unlabeled recall (UR), and F1 (Harmonic mean of UP and UR) of parsed trees against a number of gold trees (trees in the treebank).

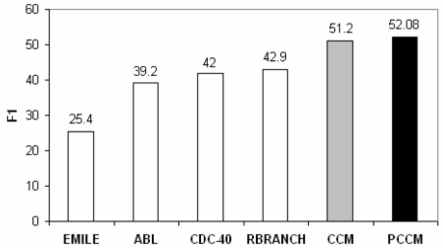


Figure 2. Parsing performance of PCCM method comparison with other unsupervised methods on ATIS corpus.

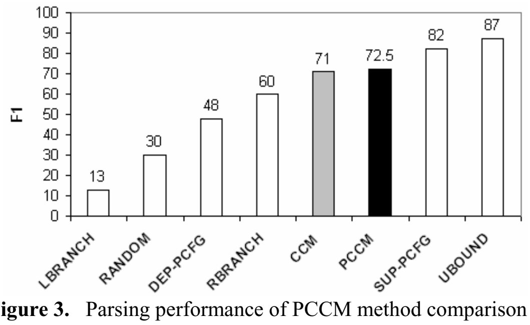


Figure 3. Parsing performance of PCCM method comparison with other unsupervised methods on WSJ-10 corpus.

Figure 3 shows the results of PCCM against that of a number of other techniques including DEP-PCFG [9] and SUP-PCFG [3]. The random, left- and right-branching approaches are also shown as the baselines. The random method chooses binary trees randomly. The left- and right-branching methods respectively choose left and right branching chains parsing. Since the structure of parse trees are binary, upper bound (UBOUND) of UP and F1 are less than 100 percent.

The results in both figure 2 and 3 show that PCCM on ATIS and WSJ-10 corpuses outperforms other unsupervised grammar induction methods.

In Persian, two different training corpuses were manually developed. The sentences of these corpuses contain less than 11 words, and have been extracted from a corpus named Peykareh [24, 25], which has been collected from formal newspapers in Persian. Peykareh has more than 32255 sentences and uses a tag set similar to the tag set used in [20, 26]. For extracting sentences, the punctuation and null elements were removed. The first corpus includes 3000 sentences, which have been manually changed in such a way that the structure of "S PP O V" is held. The common property of the sentences in this corpus is that the order of words are artificially fixed (i.e., they are not free in order). The second corpus comprises 2500 sentences of free word order. Some other features of these two corpuses are shown in table 1.

Table 1. Main features of first and second corpuses.

	Corpus 1	Corpus 2
Num. of sentences	3000	2500
Max. Len.	10	10
Min. Len.	2	2
Avg. Len	7	7
Num. of Pos Tags	18	18
Num. of Words	22153	18482

In Persian, we first ran CCM and PCCM on each of the above corpuses, separately. We also joined these corpuses to create a new mixed corpus, and repeated the experiments on this corpus, too. The results are shown in table 2. Three baselines for Peykareh are shown in table 3.

Table 2. Comparison of CCM and PCCM methods on Persian Corpuses.

Corpus	Method	UP	UR	F1
First Corpus	**CCM**	44.15	68.45	53.68
	PCCM	48.01	70.89	57.25
Second Corpus	**CCM**	26.67	51.3	35.17
	PCCM	31.21	54.23	39.62
Third Corpus	**CCM**	32.92	55.2	41.42
	PCCM	37.92	59.17	46.22

Table 3. Baselines for Peykareh treebank.

Method	UP	UR	F1
LEFT-BRANCHING	25.07	16.45	19.87
RIGHT-BRANCHING	17.63	11.57	13.97
UNBOUND	94.35	100	97.09

Table 3 shows that Persian, unlike English that is highly right-branching, is neither left- nor right-branching, which was also observed by [11]. However, high upper bound of F1 shows that Persian has a binary structure.

The results of table 2 show the effect of the free word orderness on the CCM's performance. The reduction in the performance of CCM on the second corpus in comparison to that of the first corpus is 18 percent in F1 score. The results of applying CCM to the combined corpus demonstrate that CCM shows little improvement.

Thus CCM method is weak in dealing with FWOLs. The reason for this weakness is that CCM works based on the repetition of constituents in their contexts. Since in FWOLs, some words can freely appear in different places of sentences, the mentioned repetition decreases substantially and, as a result, the performance of CCM worsens.

The experiments also show that PCCM outperforms CCM in both languages. However, the improvement achieved by using PCCM's parent information is more considerable in FWOLs.

An important implementation issue to note is the restriction of parent information usage in spans with maximum length of 5. In order to select an appropriate value for the maximum span's length, we applied PCCM to different maximum span's lengths, and to different corpuses. As it is shown in figure 4, the best performance is achieved when the spans are shorter than or equal to 5 words. One possible reason is that as spans get longer, the co-occurrence of spans and their parents will substantially decrease, and thus parsing will be less-informative.

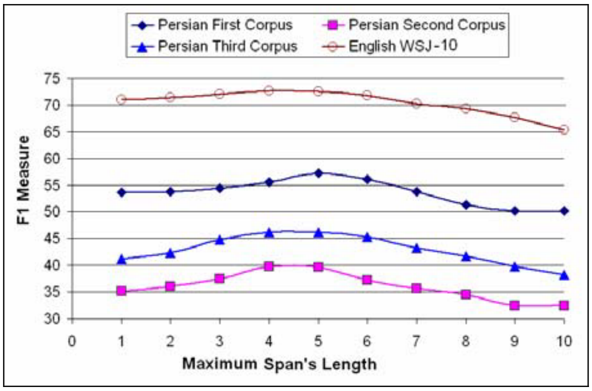


Figure 4. The effect of using parent information for different span's length in English WSJ-10 and Persian first, second and third corpuses.

5 CONCLUSION

Constituent Context Model (CCM) is currently the state of the art in unsupervised grammar induction. It combines distributional clustering methods with an EM parameter search. CCM works based upon sequences of words (spans) repetition. However, in free word order languages such as Persian, words can grammatically appear in different places of sentences and, as a result, the number of occurrences of each span type decreases. Consequently, CCM faces more divergent information. In this paper, we proposed a novel approach, called parent-based constituent context model (PCCM), by adding some history notion of context and constituent information of each span's parent. Considering parent information for constituents and contexts prevents from probability divergence and parsing will be more-informative. To evaluate the new method, we applied CCM and PCCM to both English and Persian (as a free word order language). The results of applying the new method to several corpuses with different degree of free word orderness show that using parent information improves CCM's performance, particularly when the degree of free word orderness is high.

REFERENCES

[1] T. Thanaruk and M. Okumaru, Grammar Acquisition and Statistical Parsing. Journal of Natural language Processing 2 (3), 1995.

[2] D. Klein and C. D. Manning, . Natural Language Grammar Induction Using a Constituent-Context Model, Advances in Neural Information Processing Systems 14 (NIPS 2001), vol.1. MIT Press, pp. 35–42, 2001.

[3] D. Klein and C. D. Manning, A Generative Constituent-Context Model for Improved Grammar Induction. In: ACL 40, pp. 128–135, 2002.

[4] D. Klein, The Unsupervised Learning of Natural Language Structure. Ph.D. Thesis, Department of Computer Science, Stanford University, (2005).

[5] A. Clark, Inducing syntactic categories by context distribution clustering, In the 4th Conference on Natural Language Learning, 2000.

[6] A. Clark, Unsupervised Language Acquisition: Theory and Practice. Ph.D. Thesis, University of Sussex, (2001).

[7] J. K. Baker, Trainable Grammars for Speech Recognition. In Klatt, D. H., & Wolf, J. J. (Eds.), Speech communication papers for the 97th meeting of the Acoustic Society of America, pp. 547–550, 1979.

[8] 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.

[9] G. Carroll and E. Charniak, Two Experiments on Learning Probabilistic Dependency Grammars from Corpora. Tech. rep. CS-92-16, Department of Computer Science, Brown University, 1992.

[10] B. Keller and R. Lutz, Evolving Stochastic Context-Free Grammars from Examples Using a Minimum Description Length Principle. In Workshop on Automatic Induction, Grammatical Inference and Language Acquisition, 1997.

[11] H. Feili, and G. R. Ghassem-Sani, Unsupervised Grammar Induction Using History Based Approach, In Computer Speech and Language 20 644–658, (2006).

[12] A. Stolcke and S. M. Omohundro, Inducing Probabilistic Grammars by Bayesian Model Merging. In Grammatical Inference and Applications, Proceedings of the Second International Colloquium on Grammatical Inference. Springer Verlag, (1994).

[13] S. Chen, Bayesian Grammar Induction for Language Modeling. In Proceedings of the 33rd Annual Meeting of the ACL, pp. 1995.

[14] S. M. Lamb, On the Mechanization of Syntactic Snalysis. In 1961 Conference on Machine Translation of Languages and Applied Language Analysis, Vol. 2 of National Physical Laboratory Symposium No. 13, pp. 674–685, 1961. Her Majesty's Stationery Office, London.

[15] E. Brill and M. Marcus, Automatically Acquiring Phrase Structure Using Distributional Analysis. In Proceedings of DARPA workshop on speech and natural language, 1992.

[16] S. Finch, N. Chater, and M. Redington, Acquiring Syntactic Information from Distributional Statistics. In Levy, J. P., Bairaktaris, D., Bullinaria, J. A., & Cairns, P. (Eds.), Connectionist Models of Memory and Language. UCL Press, (1995).

[17] M. Van Zaanen, ABL: Alignment-Based Learning, In Proceedings of the 18th International Conference on Computational Linguistics (COLING 18), 961–967, 2000.

[18] M. Van Zaanen and P. Adriaans, Comparing two unsupervised grammar induction systems: Alignment-based learning vs. Emile. Research report series 2001.05, School of Computing, University of Leeds, 2001.

[19] A. Radford, Transformational Grammar, Cambridge University Press, Cambridge, 1988.

[20] K. Megerdoomian, Persian Computational Morphology: A Unification-Based Approach, NMSU, CRL. Memoranda in Computer and Cognitive Science, MCCS-00-320 , 2000.

[21] S. A. Mirroshandel, G. R. Ghassem-Sani and M. A. Honrapisheh, Using of the Constituent Context Model to Induce a Grammar for a Free Word Order Language: Persian. Proceedings of the 3rd L&TC, pp. 443-447, Poland, October, 2007.

[22] S. A. Mirroshandel, Persian Unsupervised Grammar induction using Constituent Context Model. M.Sc. Thesis, Department of Computer Science, Sharif University of Technology, 2007 (in Persian).

[23] H. Feili and G. R. Ghassem-Sani, An Application of Lexicalized Grammars in English-Persian Translation. Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004). Universidad Politecnica de Valencia, Spain, pp. 596–600, 2004.

[24] M. Bijankhan. The feasibility study for Persian language modeling, The Journal of Literature 162–163 (50–51), 81–96, 2003 (in Persian).

[25] M. Bijankhan, The role of corpus in generating grammar: presenting a computational software and corpus, Iranian Linguistic Journal 2 (19), 48–67, 2005 (in Persian).

[26] J.W. Amtrup, H. R. Rad, K. Megerdoomian and R. Zajac, Persian-English Machine Translation: An Overview of the Shiraz Project, NMSU, CRL. MCCS-00-319, 2000.