A Statistical Approach for the Induction of a Grammar of Arabic

Essma Selab
Natural Language Processing and Machine Learning Research Group (TALAA), Laboratory for Research in Artificial Intelligence (LRIA)
Université des Sciences et de la Technologie Houari Boumediene (USTHB), BP 32, El Alia 16111 Bab Ezzouar, Algiers, Algeria
eselab@usthb.dz

Ahmed Guessoum
Natural Language Processing and Machine Learning Research Group (TALAA), Laboratory for Research in Artificial Intelligence (LRIA)
Université des Sciences et de la Technologie Houari Boumediene (USTHB), BP 32, El Alia 16111 Bab Ezzouar, Algiers, Algeria
aguessoum@usthb.dz

Abstract— Over the last decade, a lot of research has focused on Arabic Natural Language Processing (ANLP). Various approaches and techniques have been used to develop ANLP tools. Some of these are rule-based while others are statistical or machine-learning-based. However, the development of some ANLP tools depends on the availability of a good Arabic grammar which covers the entire language. It turns out that the Arabic grammar used by most of the developed approaches was hand-crafted and most often extracted from short sentences. This manual development process is painstaking and time consuming while the developed grammar cannot describe the entire Arabic language.

We present in this paper a novel approach to automatically inducing a grammar of Arabic. The proposed method is language-independent. It combines a statistical n-gram extraction process and a constraint satisfaction step followed by a substitution phase to automatically induce Arabic grammatical rules from TALAA, a voluminous Arabic corpus. The methodology used is presented along with the results of applying a new evaluation metric.

The evaluation of the method shows that the induced Arabic grammar largely covers the Arabic language. The induced rules can be used to improve the current accuracy of Arabic parsers specifically, and Arabic NLP tools more generally.

Keywords— Arabic Natural Language Processing; grammar rules; statistical frequency distribution; n-grams; parsing.

I. INTRODUCTION

Arabic is one of the six official languages of the United Nations[1]. It is spoken by about 420 million people and used by one billion six hundred million Muslims [27] in their daily worship (prayers, recitation of the Holy Qur'an, etc.).

Arabic belongs to the Semitic family. It is an agglutinative language, written and read from right to left. It has an alphabet set of 28 letters [25]. In Arabic, clitics modify nouns, verbs, and adjectives which they relate. The morpho-syntactic features of Arabic make its automatic analysis a fastidious process because they increase the rate of ambiguity during the various analysis steps such as tokenization, morphological analysis, parsing, etc. [34, 23].

Over the last ten to fifteen years, various methodologies have been used to develop several Arabic Natural Language Processing tools and, as a consequence, a number of systems have been developed. However, unlike the English language, except for a few cases, Arabic still lacks NLP tools that can cover the various applications with high quality [38]. On the other hand, the quality of NLP tools is nowadays largely based on the availability of resources such as corpora, grammatical rules, dictionaries, etc. These allow indeed the analysis of the language sentences in large quantities and sufficient variations. Unfortunately, most of the developed approaches use hand-crafted Arabic grammar rules that were extracted most often from short sentences. This manual process takes a lot of time and requires expertise. The resulting hand-crafted Arabic grammars cannot describe the entire language. [35, 5]. The automatic induction of an Arabic grammar is required in order to improve the current accuracy of Arabic parsers specifically and ANLP tools more generally.

We present in this paper the methodology used to automatically induce Arabic grammatical rules from a large corpus. The proposed approach is based on an n-gram extraction process and the exploitation of related statistics interleaved with a substitution step. The algorithm then applies a constraint satisfaction process before extracting the grammatical rules. The TALAA corpus [41] was used to test and evaluate the proposed method.

In Section 2, we present some of the research work related to grammar induction. Our approach to Arabic grammar induction is explained in Section 3. The evaluation methodology and results are presented in Section 4 and the conclusion is given in Section 5.

II. RELATED WORK

Grammar Induction (GI), also known as grammar inference, is the process of acquiring grammars from a set of training data (words, trees, strings, etc.). The resulting grammar which is learned from the training data is then used to construct a syntactic parser for the language, i.e. a tool that is meant to output the syntactic structure(s) (parse tree(s)) of any input sentence [22, 17]. Automatic grammar generation is a challenging NLP research area. It includes methods for building

[1] These are Arabic, Chinese, English, French, Russian, and Spanish.;

978-1-5090-0478-2/15/$31.00 ©2015 IEEE

language models [7, 15], child language acquisition, unsupervised parsing using treebanks, etc.

Several works have been developed to classify grammar induction methods and various classifications according to several features (algorithm, required data, etc.) have been proposed. Roberts [40] argues that GI classification methods depend on the learning algorithm and can be classified into Categorial Grammar-based models, memory-based learning models, evolutionary computing and string-pattern search models. Edelman [24] suggested a classification based on the required input data; he classified GI algorithms into three categories: Supervised methods use a fully parsed and tagged treebank [12, 13, 14, 19, 36]; Semi- supervised methods use less supervision information than the previous category [39]; and unsupervised ones need only POS-tagged sentences without any supervised information. The latter may use a probabilistic context free grammar, the Inside Outside approach, the expectation maximization (EM) algorithm, etc., to induce grammatical rules in Chomsky Normal Form [6, 29, 30, 31, 32]. Finally, Cramer [20] classified GI into tag-based and word-based methods.

The ADIOS (Automatic DIstillation Of Structure) algorithm is a statistical method proposed in [42]. It is an unsupervised grammatical inference approach that works in three phases: initialization, pattern distillation and generalization. ADIOS applies a greedy learning algorithm to the graph representing sentences for identifying significant patterns and selecting the best pattern to assign a new non-terminal. The algorithm has been evaluated on the ATIS treebank and a precision of 65.7%, a recall of 30.08% and an f1-score of 42% were reported. Emile is a supervised grammatical inference method proposed in 1992 [1] and successively updated until the latest version in 2002 [2]. The main theoretical concepts behind the EMILE algorithm are expressions and contexts. The main idea is the substitution of expressions of the same syntactic type and expression of the same context. that are clustered together. In 2008 [26] presented UnsuParse algorithm that learns syntactic structures from a corpus of untagged sentences. It applies statistical methods to evaluate the co-occurrences of words. UnsuParse reported an f1-score of 63.4% on the NEGRA10, and a 45.5% NEGRA40. In 2012 [9] present the result of the PASCAL Challenge on the Grammar Induction competition. Their challenge made use of 10 different treebanks annotated in a range of different linguistic formalisms and covering 9 languages. [43] Presented a novel approach to unsupervised learning of probabilistic natural language grammars named unambiguity regularization. [8] propose a EM-based induction algorithm for inducing Combinatory Categorial grammar.

Arabic still lacks approaches that can cover this domain with high quality; most of the work on induction of grammar was performed on English, Chinese or German. Since producing annotated data is costly and takes a lot of time, researchers have used unsupervised grammar induction methods.

The approach we propose here is based on unsupervised learning. It uses statistical information to extract Arabic grammatical rules that cover the Arabic language as widely as possible. This will be explained in the next section.

III. Design of the Approach

The hand-crafted development of a grammar, especially for complex languages like Arabic, is time consuming and so far has not produced Arabic parsers that cover the entire language. We have opted instead for the automatic induction of a grammar of Arabic using probabilistic analysis.

The intuition behind our approach is that the grammatical groupings and sub-groupings in a language structure (Noun phrases, Verb phrases, etc.) is probably due to the frequent co-occurrence of words of various grammatical categories. This must have guided the early grammarians like Al-Khalil Ibn Ahmad Al-Farahidy and Sibawayh to creating names for the groups and sub-groups and using these in what is today known as production rules. After reflection, we have concluded that if we want to induce grammar rules from examples, one way of doing it would be to emulate the statistics-based process by finding out the frequencies of co-occurrences of n-grams of POS-tags. Each time an n-gram of POS-tags is found to have the highest frequency, a new grammar rule is created for it and a non-terminal is coined for it and substituted in the corpus for each occurrence of this specific n-gram. The process is repeated iteratively until the entire grammar is produced.

Before the application of the proposed grammar induction algorithm, we first needed a data preparation process.

A. Data preparation

Arabic is a rich and complex language. Inducing Arabic grammar rules involves the use of a large and rich Arabic corpus. We have thus decided to use the TALAA corpus to test and evaluate the proposed algorithm.

The TALAA corpus developed by [41] is a voluminous and varied corpus built from daily Arabic newspaper websites. It is a collection of more than 14 million words (15,891,729 tokens) contained in 582,531 different articles. We have taken part of the TALAA corpus and POS-tagged it using the SAIE "Statistical Arabic Information Extraction" system [3] to construct an annotated Arabic collection of more than 13,218 tokens. The SAIE POS-tagger uses a set of 58 fine-grained tags and was reported to have an F-measure of 97%. The corpus we have annotated was manually checked by two human experts and structured into an XML file.

B. An Approach to the Induction of Arabic Rules

Arabic is a morphologically rich language where prepositions, conjunctions and pronouns can be attached to a word stem as prefixes or suffixes. For example, in the word سَيَعْلَمُونَهُ (sayaEolamuwnahu[2] / they will know it): the antefix سَـ (sa / will), the prefix يَـ (ya/ they), the suffix ونَـ (wna / they) and the postfix ـهُ (hu / it) are attached to the stem علم (Elm) to yield the agglutinative form [10]. Due to this agglutinative nature of

[2] Buckwalter Transliteration is used in this paper to represent Arabic words : http://languagelog.ldc.upenn.edu/myl/ldc/morph/buckwalter.html

Arabic, our proposed approach takes as input tokenized, stemmed sentences in order to distinguish between different parts that constitute a word (prefixes, suffixes, stems). This justifies our choice of the TALAA corpus produced using the SAIE Tagger which produces tags that reflect the stems and affixes present in a word (or text).

1) The process of N-gram extraction

The N-gram extraction process is an important step in the proposed approach. We have presented above the intuition behind our approach to grammar induction and explained the central role of n-grams in this understanding of grammar induction. Thus we needed to generate n-gram statistics for the Arabic language within our approach. However, these n-grams are based on POS-tags not tokens.

The n-gram extraction step, through its generation of frequency distributions, allows the identification of all the constituents of the sentences (unigram POS-tags) and the chunks that correspond to the sentence phrases (bi-gram, tri-gram, etc., POS-tag sequences). The algorithm used to extract Arabic grammatical rules from tagged sentences is given in Fig. 1 and all the details of the proposed approach are explained thereafter.

```
Input:  POS-tagged Arabic sentences (corpus)
Output: Induced Arabic grammar

1: Begin()
2:     Data_File = set of POS-tagged sentences
3: Repeat:
4:   a= N_Gram_extraction(Data_File)
5:   F= Frequency_distribution(a)
6:   R=N_gram[Max(F)]
7:   Substitution(Data_File, R, Nonterminal)
8:
9: until (no more rules can be extracted)
10:
11:
12:  End()
```

Fig. 1. Arabic grammar induction algorithm.

2) Details of the Approach

(1) INITIALIZATION: The input data of the algorithm is a set of POS-tagged Arabic sentences.

(2) N-GRAM EXTRACTION: The N-gram extraction step extracts all the n-grams that constitute the corpus from 2-grams to n-grams, where n is the length of the sentence being processed. The input of this step is the set of POS-tagged sentences (corpus) and the output is the collection of all extracted n-grams. An illustration of this step is presented in the following example where S_i: corresponds to sentence i, T_i: is the tokenization of sentence i and P_i: is the POS-Tagging of sentence i.

S_{20} السعادةُ في رضى الوالدين. (alsa~EAdapu fy riDY AlwAlidayn / Happiness lies in the satisfaction of (one's) parents)

Tokenizing S_{20} gives T_{20} with seven (07) tokens; T_{20}: ال سعادة في رضى ال والد ين .

T_{20} gets tagged as P_{20} (using the SAIE Tagger):

P_{20}: DEF NOUN PREP NOUN DEF NOUN SUFF_SUBJ_ALL PUNC

Fig. 2 summarizes the different n-grams (for n ϵ [2, 7]) that have been extracted from the 20th sentence of the corpus at the first iteration on Line 4 of the algorithm.

20 0 2	DEF NOUN	ال سعادة
20 0 3	DEF NOUN PREP	ال سعادة في
20 0 4	DEF NOUN PREP NOUN	ال سعادة في رضى
20 0 5	DEF NOUN PREP NOUN DEF	ال سعادة في رضى ال
20 0 6	DEF NOUN PREP NOUN DEF NOUN	ال سعادة في رضى ال والد
20 0 7	DEF NOUN PREP NOUN DEF NOUN SUFF_SUBJ_ALL	ال سعادة في رضى ال والد ين
20 1 3	NOUN PREP	سعادة في
20 1 4	NOUN PREP NOUN	سعادة في رضى
20 1 5	NOUN PREP NOUN DEF	سعادة في رضى ال
20 1 6	NOUN PREP NOUN DEF NOUN	سعادة في رضى ال والد
20 1 7	NOUN PREP NOUN DEF NOUN SUFF_SUBJ_ALL	سعادة في رضى ال والد ين
20 2 4	PREP NOUN	في رضى
20 2 5	PREP NOUN DEF	في رضى ال
20 2 6	PREP NOUN DEF NOUN	في رضى ال والد
20 2 7	PREP NOUN DEF NOUN SUFF_SUBJ_ALL	في رضى ال والد ين
20 3 5	NOUN DEF	رضى ال
20 3 6	NOUN DEF NOUN	رضى ال والد
20 3 7	NOUN DEF NOUN SUFF_SUBJ_ALL	رضى ال والد ين
20 4 6	DEF NOUN	ال والد
20 4 7	DEF NOUN SUFF_SUBJ_ALL	ال والد ين
20 5 7	NOUN SUFF_SUBJ_ALL	والد ين

Fig. 2. Sample of the n-grams extracted by the algorithm after the application of the first iteration of its Line 4 on the 20th sentence of the corpus

The meaning of the codes assigned to each extracted n-gram (see Fig. 2) is as follows:

Let us consider the following line (of an extracted n-gram):

20 3 7 NOUN DEF NOUN SUFF_SUBJ_ALL رضى ال والد ين

20 corresponds to the position i of the sentence in the corpus (i.e. 20th sentence in this case); 3 is the position in the sentence of the first gram of the extracted n-gram; and 7 is the position in the sentence of the last gram of the extracted n-gram.

NOUN DEF NOUN SUFF_SUBJ_ALL: is the extracted chunk from the 20th sentence S_{20} of the corpus, this 4-gram starts from the 3rd token of the sentence until the 7th one (indicated by the sequence 20 3 7). It is to be noted that at this stage the "chunk" may be grammatically valid or not.

رضى ال والد ين: This third part corresponds to the Arabic chunks of the sequence 20 3 7.

(3) FREQUENCY DISTRIBUTION: We calculate here the frequency or count of occurrence of each extracted n-gram. At this step, the frequency distribution is applied to n-grams representing sequences of POS-tags rather than words.

(4) RULE EXTRACTION: Based on the frequency distribution results and constraints satisfaction process (explaine hereafter). Arabic grammar rules are statistically induced one at a time (Line 6 of the algorithm). Each extracted rule left hand side is a nonterminal symbol, and its right hand side is the n-gram that has the maximum frequency distribution.

Rule: NT_i ➔ N_grams[Max(frequency distribution)] where $i \in \mathbb{N}$.

Constraint 1: Since Arabic is an agglutinative language, the SAIE tagger (used to preprocess our data) uses a tokenizer and a stemmer module to separate affixes from stems (see the example in Table1).

TABLE I. ARABIC WORD AFTER TOKENIZATION AND STEMMING STEP.

Input	SAIE output		
	Prefix	Lemma	suffix
	Def	Noun	Suff_M_P
المعلمون(teachers)	ال	معلم(teacher)	ون
AlmuEalimwno	Al	muEalim	Uwn

In this example, the trigram `DEF NOUN SUFF_M_P` represents the POS-tag sequence of the word المعلمون. Since a rule cannot start with suffixes or end with definite articles or prefixes such as the rules:

```
NT ==> SUFF_F_S PREP DEF
NT ==> SUFF_F_P ADJ
```

We have to satisfy some constraints related to the Arabic morphology. Thus Constraint 1 can be stated as follows:

A rule cannot start with a suffix nor end with a prefix or a definite article, since a suffix is related (attached) to the previous token and the prefix to the next one. A rule must satisfy the following general formulation.

NT_i ==> Prefix A_1 … A_l
NT_j ==> B_1 … B_m suffix
NT_k ==> DEF C_1 … C_n
where
$\{i, j, k\} \in \mathbb{N}$, $\{A1, \ldots, Al, B1, \ldots, Bm, C1, \ldots, Cn\} \in (N \cup \Sigma)$

Constraint 2: Suppose we have two n-grams "`PREP DEF NOUN SUFF_F_P`" and "`DEF NOUN SUFF_F_P`". Without any constraints, the algorithm will output the following rules:

NT_i ==> PREP DEF NOUN SUFF_F_P
NT_j ==> DEF NOUN SUFF_F_P
where *i* and *j* are integers.

But since the 4-gram `PREP DEF NOUN SUFF_F_P` includes the 3-gram `DEF NOUN SUFF_F_P` the rule containing the smallest gram has to be extracted first because the rule that produces the 4-gram has to include the nonterminal that corresponds to the 3-gram `DEF NOUN SUFF_F_P`. Hence the need to satisfy the proposed constraint to obtain rules with the following correct formulation:

NT_i ==> DEF NOUN SUFF_F_P
NT_j ==> PREP NT_i
Where *i* and *j* are integers.

This second constraint is related to the ordering of the extraction of rules. Since the extraction of the grammar rules is based on the frequency distribution values, then the n-gram with the highest frequency distribution is selected to build the extracted rule. The example above explains the importance of the order of the extracted Arabic rules, and why we have to satisfy this constraint before automatically inducing other grammar.

$\{X,Y\} \in (N \cup \Sigma)$, where X has got n-grams, and Y has m-grams and m<n.
If n_gram ⊆ m_gram (eq. X = a Y, $a \in (N \cup \Sigma)$), the rule containing the smallest gram eq. NT_i ==> Y has to be extracted first, to obtain the following sequence:
NT_i ==> Y
NT_j ==> a NT_i / $\{i, j\} \in \mathbb{N}$

At the first iteration of the algorithm, the bigram `DEF NOUN` corresponded to the most frequent n_gram and NT_1 ==> DEF NOUN was the first rule extracted by the algorithm.

(5) SUBSTITUTION: After the rule extraction process, we obtain rules of the form NT_i ==> N-grams. At this step we replace in the input corpus all the instances of n-gram occurrences of the extracted rule by the nonterminal occurring in its head.

At the first iteration of the algorithm all the bigrams `DEF NOUN` (corresponding to the first extracted rule) are substituted by their corresponding nonterminal NT_1 (Fig.3 shows the process of substitution).

DEF NOUN
DEF NOUN PREP
DEF NOUN PREP NOUN
DEF NOUN PREP NOUN DEF
DEF NOUN PREP NOUN **DEF NOUN**
DEF NOUN PREP NOUN **DEF NOUN** SUFF_SUBJ_ALL
NOUN PREP
NOUN PREP NOUN
NOUN PREP NOUN DEF
NOUN PREP NOUN **DEF NOUN**
NOUN PREP NOUN **DEF NOUN** SUFF_SUBJ_ALL
PREP NOUN
PREP NOUN DEF
PREP NOUN **DEF NOUN**
PREP NOUN **DEF NOUN** SUFF_SUBJ_ALL
NOUN DEF
NOUN **DEF NOUN**
NOUN **DEF NOUN** SUFF_SUBJ_ALL
DEF NOUN
DEF NOUN SUFF_SUBJ_ALL
NOUN SUFF_SUBJ_ALL

NT_1
NT_1 PREP
NT_1 PREP NOUN
NT_1 PREP NOUN DEF
NT_1 PREP NOUN **NT_1**
NT_1 PREP NOUN **NT_1** SUFF_SUBJ_ALL
NOUN PREP
NOUN PREP NOUN
NOUN PREP NOUN DEF
NOUN PREP NOUN **NT_1**
NOUN PREP NOUN **NT_1** SUFF_SUBJ_ALL
PREP NOUN
PREP NOUN DEF
PREP NOUN **NT_1**
PREP NOUN **NT_1** SUFF_SUBJ_ALL
NOUN DEF
NOUN **NT_1**
NOUN **NT_1** SUFF_SUBJ_ALL
NT_1
NT_1 SUFF_SUBJ_ALL
NOUN SUFF_SUBJ_ALL

Fig. 3. Sample of the substitution process.

(6) Steps (2) to (5), i.e. Lines 4 to 7 in Fig. 1, are repeated until no new rules can be extracted.

Fig.4 shows the different n-grams ($n \in [2, 7]$) extracted from the 20^{th} sentence of the corpus at the second iteration of the algorithm.

20 0 3	NT_1 PREP	ال سعادة في
20 0 4	NT_1 PREP NOUN	ال سعادة في رضى
20 0 6	NT_1 PREP NOUN NT_1	ال سعادة في رضى ال والد
20 0 7	NT_1 PREP NOUN NT1 SUFF_SUBJ_ALL	ال سعادة في رضى ال والد ين
20 2 4	PREP NOUN	في رضى
20 2 6	PREP NOUN NT_1	في رضى ال والد
20 2 7	PREP NOUN NT1 SUFF_SUBJ_ALL	في رضى ال والد ين
20 3 6	NOUN NT_1	رضى ال والد
20 3 7	NOUN NT_1 SUFF_SUBJ_ALL	رضى ال والد ين
20 4 7	NT1 SUFF_SUBJ_ALL	ال والد ين

Fig. 4. Sample of the output of the algorithm at the 2nd iteration after the n-gram extraction process.

IV. Experimental Results

The aim of this work was to automatically induce a set of grammar rules that describe the Arabic language as exhaustively as possible. The proposed approach was applied to a collection of 13,218 tokens, 1500 sentences having between 3 and 19 tokens.

The developed system has generated 172 different Arabic grammar rules. Some of them corresponding to noun phrases, (NT_1 ==> DEF NOUN, NT_{24} ==> PCALL[3] NOUN NT_{18}) others to verb phrases, (NT_8 ==> IV2 IVERB[45]) or prepositional phrases (NT_2 ==> PREP NT_1). The substitution step of the approach allows us to combine different non terminals to deduce new rules. The figure below shows a sample of the extracted rules.

```
NT1 ==> DEF NOUN
NT2 ==> PREP NT1
NT3 ==> PVERB SUFF_SUBJALL
NT4 ==> NOUN SUFF_S_INDEF
NT5 ==> PREP NOUN
NT6 ==> PVERB NT1
```

Fig. 5. A sample of the extracted rules.

Parsing with the induced grammar

In order to evaluate the induced grammar using the proposed method, we first modified the NLTK[6] parser by replacing the NLTK rules database by the Arabic grammar rules output by the proposed algorithm. Next, we randomly selected a set of 200 different sentences (long sentences, short ones, verbal, nominal, etc.), which we parsed using the NLTK parser in which we injected our induced grammar. We then computed the parsing performance.

The example below shows some parsed sentences:

Sentence 1: قرأت كتابا مفيدا (qara>tu kitAbF mufydF / I read an interesting book).

The rules of grammar G_1 cover this verbal phrase:

```
      S    ==>  NT10 NT19
      NT10 ==>  NT3 NT4
G1:   NT3  ==>  PVERB SUFF_SUBJALL
      NT4  ==>  NOUN SUFF_S_INDEF
      NT19 ==>  ADJ SUFF_S_INDEF
```

The parse tree produced using the modified NLTK parser is:

```
S (NT10 (NT3 (PVERB SUFF_SUBJALL) NT4 (NOUN SUFF_S_INDEF )) NT19 (ADJ SUFF_S_INDEF))
```

Sentence 2: ظاهرة الاحتجاجات ليست غريبة على المجتمع الجزائري (ZAhirapu Al<HtijAjAto laysato garybapo Eala AlmujotamaEo AljazA}iry / The phenomenon of protests is not peculiar to the Algerian society).

The set of induced rules G_2 which covers this nominal phrase is:

```
      S     ==> NT39 NT52
      NT39  ==> NT11 NT15
      NT15  ==> DEF NOUN SUFF_F_P
      NT52  ==> NT44 NT37
G2:   NT44  ==> NEGATION SUFF_SUBJ_3FS NT11
      NT11  ==> NOUN SUFF_F_S
      NT37  ==> NT2 NT16
      NT2   ==> PREP NT1
      NT1   ==> DEF NOUN
      NT16  ==> DEF ADJ
```

[3] حرف نداء (Harofu nidA'/ call symbols)
[4] Imperfect verb
[5] Details of the tag set in [3]
[6] Natural Language Tool Kit available at www.nltk.org

V. EVALUATION METRIC

In order to evaluate the quality of the proposed approach, we have defined a numerical metric which is largely based on the familiar precision metric. The basic unit of the proposed metric is a sentence. We calculate the weights of all the parsed test sentences and then multiply them by a factor before computing the precision.

First we calculate the weight w_i of each parsed test sentence S_i by estimating the probability **P** of the n-gram sequences **(e_1 e_2 … e_m)** that constitute it. The more often the n-grams are observed in the training corpus, the higher their probability is and hence the weight of the whole sentence.

$$P(s_i) = P(e_1)P(e_2|e_1)P(e_3|e_1e_2) \dots P(e_i|e_1e_2 \dots e_{i-1}) = \prod_{k=1}^{m} P(e_k|e_1 \dots e_{k-1}) \quad (1)$$

Jurafsky argues [28]: "The intuition of the N-gram model is that instead of computing the probability of a word given its entire history, we will approximate the history by just the last few words". Given the bigram model and the Markov assumption, the general equation (2) is N-gram approximation to the conditional probability of a word in a sequence.

$$P(s_i) = P(e_1e_2 \dots e_m) \approx \prod_{k=1}^{m} P(e_k|e_{k-1}) \quad (2)$$

Thus the weights w_i are calculated as follow:

$$w_i = \prod_{k=1}^{m} P(e_k|e_{k-1}) \quad (3)$$

Next let l be the length of the test sentence and p the length of the n-gram which is correctly parsed using the induced grammar. We compute the Reward Factor RF

$$RF = \frac{p}{l}$$

Note that in the special case where p=l, RF = 1, i.e. the whole sentence was correctly parsed. Table 2 below shows to result of the RF corresponding to the test set:

TABLE II. RESULT OF THE REWARD FACTOR.

RF	**1**	**[0.5,1[**	**< 0.5**
NUMBER OF SENTENCES	103	76	21
%	51.5%	38%	10.5%

The following formula reflects the idea that the more chunks are correctly parsed, the higher the score that will be considered for this sentence in the overall grammar precision.

$$Precision = \frac{\sum_{k=1}^{n} RF * p(s_k)}{\sum_{k=1}^{n} p(s_k)}$$

In the proposed formula, the weights of the test sentences are used so as to penalize even more the sentences that are similar to the training sentences but are worse parsed by the induced grammar than the unseen sentences.

The proposed induction module has achieved 87% correctness using the definition of precision given above.

The proposed algorithm extracts a large set of grammar rules which often are clusters of variations of one another (nominal, verbal, etc.) since an Arabic word can be a noun, a verb or a particle, each word taking one of two genders[7], one of three numbers[8], and three grammatical cases[9] [4]. Samples of such variations of rules are shown below:

Sample 1:

```
NT15   ==> DEF NOUN SUFF_F_P
NT26   ==> DEF NOUN SUFF_M_P
NT17   ==> DEF NOUN SUFF_F_S
NT23   ==> DEF NOUN SUFF_M_S
NT49   ==> DEF NOUN SUFF_F_D
NT40   ==> DEF NOUN SUFF_M_D
```

Sample 2:

```
NT20   ==> FUTURE IV3 IVERB
NT28   ==> FUTURE IV2 IVERB
NT32   ==> FUTURE IV1P IVERB
```

We are working on the problem of generalizing the process so as to be able to induce from a set of specific grammatical rules (like the ones in Samples 1 and 2) an equivalent small set of generalized rules that optimally describe the Arabic language. On the other hand we are trying to improve the accuracy of our system by enriching the training set with more sentences.

For the English language, some models have obtained a result higher than 87%, but these proposed methods are based on supervised information where the sentences are syntactically analyzed and bracketed in the training set rather than our proposed statistical method which is based on unsupervised data.

[7]feminine and masculine

[8] singular, dual, and plural

[9] Nominative , accusative and genitive

Language Independence

We would like to point out that the approach we have presented above, though applied to Arabic, is no way specific to it. Indeed, the same steps of corpus preparation (prefixes, stems, and suffices), n-gram generation and frequency distributions, and substitutions, are perfectly applicable to the largest majority of natural languages we can think of.

VI. Conclusion

We have presented in this paper an approach to inducing an Arabic grammar. The proposed algorithm takes as input a set of tagged Arabic corpus then applies an iterative process which is based on an n-gram extraction step, n-gram frequency distributions and a substitution step. Two constraints are used to force the algorithm to generate only meaningful rules. Though applied to Arabic, the proposed methodology can be applied to the largest majority of natural languages we can think of.

References

[1] Adriaans, P.W., 1992. Language learning from a categorial perspective. PhD thesis, University of Amsterdam, Amsterdam.

[2] Adriaans, P.W., Vervoort, M., 2002. The EMILE 4.1 grammar induction toolbox. In: Adriaans P, Fernau H, vanZaanen M (eds) Grammatical inference: algorithms and applications: 6th international colloquium: ICGI2002. Lecture notes in computer science, vol 2484. Springer, Heidelberg, pp 293–295

[3] Al Shamsi, F. N., Guessoum, A., 2006. A Hidden Markov Model - Based POS Tagger for Arabic, Proceedings of 8th International Conference on Textual Data Statistical Analysis.

[4] Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M.S., Al-Rajeh, A ,2008. Automatic Arabic Text Classification. JADT' 2008 : The 9th International Conference on the Statistical Analysis of Textual Data, France.

[5] Al-Taani, A., Msallam. M. and Wedian, S., 2012. A Top-Down Chart Parser for Analyzing Arabic Sentences. The Intrnational Arab Journal of Information Technology: vol. 9 , No 2.

[6] Amaya, F., Benedi, J.M. and Sanchez, J.A., 1999. Learning Of Stochastic Context-Free Grammars From Bracketed Corpora By Means Of Re-estimation Algorithms. In: The VIII Symposium on Pattern Recognition and Image Analysis, vol. 1, pp. 19–126, Bilbao.

[7] Baker, J.K., 1979. Trainable grammars for speech recognition. In: D.H. Klatt, J.J. Wolf (Eds.), Speech Communication Papers for the 97th Meeting of the Acoustical Society of America, pp. 547–550.

[8] Bisk, Y., Hockenmaier, J., 2012 . Induction of Linguistic Structure with Combinatory Categorial Grammars. NAACL-HLT Workshop on the Induction of Linguistic Structure. Montréal, Canada, pp. 90-95

[9] Blunsom, P., Graçaa, J.V., 2012 The PASCAL Challenge on Grammar Induction. NAACL-HLT Workshop on the Induction of Linguistic Structure, Montréal, Canada, pp. 64–80.

[10] Brahmi, A., Ech-Cherif, A. and Benyettou, A., 2013. An Arabic Lemma-Based Stemmer for Latent Topic Modeling. In: the international Arab Journal of Information Technology, Vol. 10, No.2.

[11] Buckwalter, T., 2002. "Buckwalter Arabic morphological analyzer version 1.0". LDC Catalog No: LDC2002L49. Linguistic Data Consortium, University of Pennsylvania.

[12] Charniak, E., 1997a. Statistical Parsing With A Context-Free Grammar And Word Statistics. In: Proceedings of the 14th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Menlo Park, pp. 598–603.

[13] Charniak, E., 1997b. Statistical Techniques For Natural Language Parsing. AI Magazine 18 Vol. 4, pp. 33–44.

[14] Charniak, E., 2000. A Maximum-Entropy-Inspired Parser. In: NAACL 1, pp. 132–139.

[15] Chen, S.F., 1995. Bayesian grammar induction for language modeling. In: Proceedings of the Association for Computational Linguistics, pp. 228–235.

[16] Chomsky, N., 1957. Syntactic Structures. The Hague Mouton.

[17] Clark, A. and Lappin, S., 2010. Unsupervised Learning and Grammar Induction. In The Handbook of Computational Linguistics and Natural Language Processing, Wiley-Blackwell, Oxford, UK.

[18] Collins, M., 1996. A New Statistical Parser Based On Bigram Lexical Dependencies. In: The Proceedings of the 34th Annual Meeting of the ACL, Santa Cruz.

[19] Collins, M.J., 1997. Three Generative, Lexicalized Models For Statistical Parsing. In: ACL 35/EACL 8, pp. 16–23.

[20] Cramer, B., 2007. Limitations of current grammar induction algorithms. In: Proceedings of the 45th annual meeting of the ACL: student research workshop, June 25–26, 2007, Prague, Czech Republic

[21] Dans, K. 2005. Online closure based learning of relational theories. In ILP'05: Inductive logic programming, 172-189, Bonn, Germany.

[22] De la Higuera, C., 2005. A bibliographical study of grammatical inference. Pattern Recognition, vol. 38, pp. 1332–1348.

[23] Diab , M., Hacioglu, K. and Jurafsky, D., 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In Proceedings of Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting, USA, pp. 149–152.

[24] Edelman, S., Solan, Z., Horn, D. and Ruppin, E., 2005. Learning syntactic constructions from raw corpora. In:29th Boston University conference on language development, Cascadilla Press

[25] Habash, N. y., 2010. Introduction to Arabic Natural Language Processing. A Publication in the Morgan and Claypool Publishers series, Columbia University.

[26] Hänig, C., Bordag, S., Quasthoff, U., 2008. UnsuParse: unsupervised parsing with unsupervised part of speech tagging. In: Proceedings of the sixth international language resources and evaluation (LREC 2008)

[27] Istizada, Arabic & Middle East Marketing Solutions, http://istizada.com/complete-list-of-arabic-speaking-countries-2014/ (Last visited June 2015).

[28] Jurafsky, D. and Martin, J.H., 2008. An introduction to speech recognition, computational linguistics and natural language processing. 2nd Edition Paperback. ISBN-10: 0131873210

[29] Klein, D. and Manning, C.D., 2001. Distributional Phrase Structure Induction. In: Proceedings of the Fifth Conference on Natural Language Learning (CoNLL 2001), pp. 113–120.

[30] Klein, D. and Manning, C.D., 2001b. Natural Language Grammar Induction Using A Constituent-Context Model. In: Dietterich, T.G., Becker,

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

[32] Klein, D. and Manning, C.D., 2004. Corpus-Based Induction Of Syntactic Structure: Models Of Dependency And Constituency. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 04).

[33] LDC, Linguistic Data Consortium-University of Pennsylvania, http://www.ldc.upenn.edu/. (Last visited October 2014).

[34] Maamouri, M., Bies, A. and Kulick , S. 2006. Diacritization: A Challenge to Arabic Treebank Annotation and Parsing. In Proceedings of The British Computer Society Arabic NLP/Mt Conference.

[35] Marton, Y., Habash, N. and Rambow, O., 2013. Dependency Parsing Of Modern Standard Arabic With Lexical And Inflectional Features. Computational Linguistics, 39(1).

[36] Magerman, D.M., 1995. Statistical Decision-Tree Models For Parsing. In: The Proceedings of ACL Conference.

[37] Muggleton S, 1999, Inductive Logic Programming: Issues, results and the challenge of Learning Language in Logic, Artificial Intelligence, Vol: 114, Pages: 283-296, ISSN: 0004-3702

[38] Othman, E., Shaalan, K. and Rafea, A., 2003. A ChartParser for Analyzing Modern Standard Arabic Sentence. In Proceedings of the MT Summit IX Workshop on Machine Translation for Semitic Languages: Issues and Approaches, USA.

[39] Pereira, F. and Schabes, Y., 1992. Inside-outside re-estimation from partially bracketed corpora. In: The Proceeding of 30th Annual Meeting of the ACL, pp. 128–135.

[40] Roberts, A. and Atwell, E. 2002 Unsupervised grammar inference systems for natural language. Research report number 2002.20. School of Computing, University of Leeds

[41] Selab, E. and Guessoum, A., 2015. Building TALAA, a Free General and Categorized Arabic Corpus. In: Proceedings of the International Conference on Agents and Artificial Intelligence vol. 1, pp. 284–291, Lisbon, Portugal.

[42] Solan, Z., Horn, D., Ruppin, E., Edelman, S., 2005. Unsupervised learning of natural languages. Proc Natl AcadSci USA 102(33):11629–11634

[43] Tu, K., Honavar, V., 2012. Unambiguity Regularization for Unsupervised Learning of Probabilistic Grammars. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju Island, Korea.pp, 1324–1334.