Arabic Parser Evaluation and Improvements

تقييم وتحسين محلل اللغة العربية

by

Khaled Mohamed Khaled Ezzeldin

A dissertation submitted in fulfilment

of the requirements for the degree of

MSc Informatics (Knowledge & Data Management)

at

The British University in Dubai

Professor Dr. Khaled Shaalan
June-2016

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

_____

Signature of the student

COPYRIGHT AND INFORMATION TO USERS

Abstract

This thesis focuses on comparing between two famous Arabic parsers Stanford Parser and Bikel parser using Arabic Treebank (ATB) for model training and testing and for this purpose we created a software that enables us to convert the ATB format to grammar format, convert the Arabic Morphological tags to Penn tags, and evaluate the parsers output by calculating the Precision, Recall, F-Score, and Tag Accuracy.

We also modify Bikel Parser to use the Penn tags in training to improve the Precision, Recall, F-Score, and Tag Accuracy results from the parse output.

تركز هذه الآطروحة على المقارنة بين اثنين من أشهر برامج التحللي الآحصائي للجملة العربية وهما محلل ستانفورد و بيكل باستخدام التحليل الشجري للغة العربية. هذا الكم من التحليل الشجري يستخدم لآنشاء وتجريب النموذج الأحصائي , ولهذا الغرض قمنا بتطوير برنامج قادر على تحويل التحليل الشجري الى قواعد النحو و تحويل التمثيل الصرفي العربي إلي التمثيل المستخدم في التحليل الشجري للغة العربية. كما قمنا بتقييم مخرجات المحلل الآحصائي باستخدام المعادلات المعيارية المتعارف عليها و قمنا ايضا ببعض التغيرات على محلل بيكل عن طريق استخدام التمثيل المستخدم في التحليل الشجري بهدف زيادة دقة محلل الجملة الأحصائي.

Dedication

In memory of my Father and Mother: thank you for dreaming and envisioning a different future for your children. You were our greatest teachers. You are greatly missed.

To my beloved children – Mohammed, Mariam, and Ahmed: for inspiring me endlessly. My success is yours.

Lastly: to Allah the Almighty, whom, without Him, this paper would not have been completed.

Acknowledgement

My sincere thanks to The Almighty – Allah for blessing me with potential to investigate and accomplish good results in this study.

My earnest and sincere thanks to Prof. Khaled Shaalan, for directing this theory and offering his colossal backing all through, without which I wouldn't have concocted a novel way to deal with opinion examination. His gainful remarks and quick perspectives have made it simpler for me to surpass greatness in this exploration. Not to overlook his understanding in clarifying and controlling me all through this trip. He is a decent helper and in addition a very goof reflector. He suggested as well critically challenged to think and progress further. He has been my best mentor.

And here we come to people who have been my day to day supporters, my family, my kids and my parents. Without their blessings and support, this study was next to impossible for me.

Contents

List of Tables

List of Figures

Chapter One

Introduction

1.1 Natural Language Processing

Bird et al. (2009, p.1X) state that from era to era the natural dialect and language has progressed and is hard to advance express standards, considering the double side, that it is on one side it will be anything but difficult to discover frequencies in order to check word, however then again it is trying to comprehend the human articulations so as to give only great reaction, yet the innovations in technologies expand on NLP(Natural Language Processing) has been utilized broadly as a part of foreseeing content, penmanship acknowledgment in cell telephones/PCs, machine translation(for case recoup content in English for a given content in Arabic).

Natural language processing is a field of software engineering, counterfeit consciousness, and computational semantics worried with the collaborations amongst PCs and human (normal) dialects. All things considered, NLP is identified with the zone of human–computer collaboration. Numerous difficulties in NLP include: regular dialect understanding, empowering PCs to get significance from human or characteristic dialect info; and others include common dialect era[1]. Characteristic dialect preparing is critical for more extensive range including individuals from industry (business), PC semantics and humanities figuring from the educated community (they perceive NLP as computational phonetics) (Steven et al. 2009)

1.2 Applications of Natural Language Processing

Question Answering:

QA is about building an application that can process the human language and generate the answer using a knowledge database that contains all required documents related to a specific subject (Ray and Shaalan, 2016). QA involves document retrieval, passage retrieval, and Named Entity Recognition (NER) (Oudah, Shaalan, 2016a) (Oudah, Shaalan, 2016b). Question is analyzed and then mapped to the

[1] https://en.wikipedia.org/wiki/Natural_language_processing

retrieved paragraph using the Information Retrieval methods. (Abdelbaki et al, 2011), created a question Answering system that is using the NER after recognizable proof of Question Types.

Machine Translation

MT is the procedure of making an interpretation of from source dialect to target dialect. (Al-Onaizan and Knight 2002) showed the significance of named elements in Machine interpretation utilizing monolingual and bilingual assets. The procedure took after by the creators included acknowledgment of named substances, this includes People names, Organizations, and places, and after this the phrases are transliterated. Entity acknowledgment is prominent to be the important thing stride for in addition coping with in Machine Translation.

Information Retrieval:

Taking into account an info inquiry the method required to recover the right record from the data set is named as Information Retrieval (Benajiba et al (2009)). Named Entity information is utilized as a part of Information Retrieval in order to recover the named elements from the Query info and too NE acknowledgment is fundamental in the records in order to locate the important archives mapping the inquiries.

1.3 Goals and Challenges

Arabic is an exceedingly inflectional Semitic dialect. [Farghaly and Shaalan 2009] represented one of the key components of Arabic dialect incorporates absence of capitalization, which is one of the rich element of Arabic dialect. Morphologically, it is portrayed by the accompanying elements:

- The aggregate number of Arabic letters is 28;
- Most of the Arabic words are resultant from, and transformable once more into regular roots;
- Approximately 85% of Arabic words are gotten from tri-sidelong roots;
- Arabic's normal things and particles are not transformable;
- From Around 10,000 roots arrangement of words, the nouns and verbs of Arabic are brought into existence.

- Arabic nouns have implications and can be utilized as adjectives words (الولد الماهر The capable Kid) or adverbs (تحرك بهدوء move discreetly);
- Arabic perceives three genders orientations: female, manly and neuter
- words acquired from different dialects for instance Bank "بنك"
- Arabic gives three classes to tending to the quantity of articles: solitary (singular); double (dual), and plural.
- According to Shaalan et al. (2015), the verbs could be not perfect, perfect and imperative.

Arabic and English are not quite the same as the morphological and syntactic points of view (Maytham and Allan 2011). This represents a test to the Arabic dialect scientists who wish to exploit existing English dialect preparing innovations. In addition, Arabic verbs are stamped expressly for various structures demonstrating time of the activity, voice and individual. They are additionally set apart with inclination (demonstrative, basic and interrogative). For ostensible (things, descriptive words, appropriate names), Arabic imprints genitive, accusative and nominative cases, gender, number orientation and definiteness highlights. Arabic written work is likewise known for being underspecified for short vowels. At the point when the class is otherworldly or instructive, the Arabic content would be completely indicated to maintain a strategic distance from perplexity.

Arabic Natural Language Processing Goals:

- People with minimal English foundation think that it is hard to overcome the logical distributions. In order to determine human interpreters should be enlisted who may think that its unpleasant to decipher sufficient measure of information. Subsequently ANLP would results in less cost for deciphering, abridging and recovering data in less time.
- Coinage and lexical hole examination are different territories which requires consideration and also formal and exact sentence structure is expected to safeguard the legacy of Arabic which is of high esteem.
- With the ANLP set up in not so distant future Arabic individuals can cross over any barrier between different nations.

1.4 Problem Statement and Research Questions

1.4.1 Problem Statement:

We found few researches that compare between different Arabic parsers using the same training and testing data. There is a few reaches that study the effects of Diacritic and non-Diacritic Arabic texts on parsers performance. There are a Few efforts that tried to improve the Bikel parser when it works on an Arabic text. We Found lack of software that can convert ATB format to the Grammar format and automatically evaluate the parser performance and convert between Arabic Morphological tags used in ATB and Penn Treebank tags.

1.4.2 Research Question

RQ1: Can we create a program that can convert the ATB to Grammar format and to evaluate the parsing performance and convert between Arabic Morphological tags used in ATB and Penn Treebank tags?

RQ2: What is the effect of Diacritic and non-Diacritic Arabic texts on parsers performance?

RQ3: Which parser has higher performance Bikel or Stanford parser?

RQ4: If we modify the training data to have the same ATB tags that Bikel is using and modify the mapping file inside Bikel parser could we gain higher performance?

1.5 Motivation

In contrast with cutting edge semantics, the points and inspirations of conventional Arabic sentence structure varied in two regards. Firstly, worried by ungrammatical dialect and persuaded to safeguard the dialect of the Arabic, we were principally keen on portraying Arabic's etymological tenets. Furthermore, in a similar manner as adherents of Islam today, we considered the Arabic dialect to be great when it comes to Day to Day use and in Quran. Driven by these convictions, we deliver point by point examination of a wide assortment of semantic wonders, building up a complete hypothesis of linguistic use.

1.6 Contribution

Evaluated two parsers Bikel and Stanford on the training and testing data. Evaluated the effect of Diacritic and non-Diacritic Arabic texts on parsers performance. Created Java program that converts ATB format to grammar rules format and evaluate the parsers performance by comparing the output from the parser with the gold standard and also convert between Arabic Morphological tags used in ATB and Penn Treebank tags. We modified Bikel parser to improve the parsing performance.

1.7 Thesis Structure

The following sections in this thesis includes Section 2 contains literature review, Section 3 covers the parsers with, Section 4 covers the implementation part, Section 5 includes results and evaluation, and last part Section 6 is about the conclusion.

Chapter Two

Literature Review

2.1 Arabic Challenges

Arabic and English are completely different from the morphological and grammar views. This creates a challenge to the Arabic researchers who would like to require advantage of existing English latest technologies (Shaalan, 2014), (Ray and Shaalan, 2016). Also, Arabic verbs are marked expressly for multiple forms indicating time of the action, sound and person. They're additionally marked with mood. For nominal adjectives, proper names and nouns, the case from genitive, accusative and nominative is marked through Arabic language, with near reference and determination to gender and number. Arabic writing is additionally famed for being underspecified for short vowels. Once the genus is religious or academic, the Arabic text would be totally such to avoid confusedness.

2.1.1 Compound Words

In Arabic, a word can be authored from a morpheme and appendage. There is not really any distinction amongst mind boggling and compound words in Arabic (Farghaly and Shaalan 2009).

Case: وبفعلهم

Parts:

1- و implies and

2- بـ implies by

3- فعل implies act

4- هم implies their

2.1.2 Subject Embedding

Not at all like English, Arabic is considered, from the syntactic angle, a master drop dialect which encodes the verb subject within its morphology (Farghaly and Shaalan 2009). For instance, the announcement "He played football" can be communicated in Arabic as " لعب الكرة ". The subject "He" and the verb "played" are spoken to in

Arabic dialect by the single verb-structure " لعب ". That is; "He played" is deciphered as "لعب " and "football" is interpreted as "الكرة"

2.1.3 Loose Sentence Order

Arabic shows a bigger level of opportunity in the request of words inside a sentence. It permits change of the standard request of parts of a sentence—the Subject-Verb-Object (SVO) (Jaf and Ramsay 2013). As an illustration, the sentence "The kid play football" can be deciphered, word-by-word, to the Arabic SVO phrase ‘‘الولد يلعب الكرة ". The last might be permuted to the standard Arabic request of a sentence the VSO structure ‘‘ لعب الولد الكرة‘‘. Both structures safeguard the target of the sentence. Tragically, the word by word English interpretation of the same could be not exact.

2.1.4 Subject and Object Conflation

In a few circumstances, the qualification between the subject and question in Arabic can challenge (Farghaly and Shaalan 2009). For instance, the English Arabic sentence "يحترم الأب الإبن", is in the VSO structure. Its interpretation ought to be: Son 'الإبنُ ', regards "يحترم" The Father ‘‘الأبَ '. In the event that the thing "الأبَ" is changed to "الأبُ" and "الإبنُ" is changed to 'الإبنَ'by changing just the blemish on its furthest left letter furthermore , the structure still remains a VSO one. In any case, the proposed subject turns into the article and the other way around since the interpretation will then be: Father 'الأبُ ', regards 'يحترم ', child "الإبنَ" such a basic illustration illustrates, to the point that uncertainty is very plausible if no clarifiers are furnished to help with the refinement amongst subjects and protests.

2.1.5 Word Ambiguity

A case of a questionable Arabic word is "السائل" which can be meant "fluid", or "poor person" and Arabic word "قدر" which can be meant "amount", or "announcement" Due to the undiacritized, unvowelized composing framework (Farghaly and Shaalan 2009).

2.1.6 Letter vagueness

Vagueness is not constrained to Arabic words as it were. Some Arabic letters when attached to morphemes lead to uncertain compound words(Farghaly and Shaalan 2009). Case underneath shows how attaching the letter 'ب', which compares to b in

English to a nuclear word, will swing it to a compound one. Such is the situation on the grounds that, as a prefix, the letter "ب" tackles any of the accompanying faculties: through, in, by and utilizing.

Case:

"ببركة" signifies "Through gift"

"بالبيت" signifies "In The house"

"بالمال" signifies "By The cash"

"بالقلم" signifies "Utilizing The pen"

2.1.7 Arabic Diglossia

Diglossia is a wonder (Ferguson 1959 1996) whereby two or more assortments of the same dialect exist one next to the other in the same discourse group. Arabic, notwithstanding, displays a genuine diglossic circumstance where no less than three assortments of the same dialect are utilized inside a discourse group (Farghaly 2005) and in encompassed circumstances. Traditional Arabic is the dialect of religion and is utilized by Arabic speakers as a part of their day by day supplications while Modern Standard Arabic (MSA), a later assortment of Classical Arabic, is utilized by instructed individuals as a part of more formal settings, for example, in the media, classroom, and business. With family, companions, and in the group, individuals talk their own local lingo which differs impressively from district to area.

2.2 Arabic Morphological Analysis

As stated by Al-Hamalawee (2000), an Arabic word combines the element of a single and distinct lexeme which convey a particular supremacy, this when contrasted by other researchers like Saliba and Al-Dannan (1989) who traced an Arabic word to be orthographically formalised as a composition of two spaces. The words with strong, consistent that is other than normal words are the ones with a shape morphologically. Hamza as well germination is missing letters in flawed words (El-Dahdah 1988; Al-Khuli 1991; Metri and George 1990). Words are ordered by Arab etymologists as things, verbs, or particles (Al-Saeedi 1999).

Spencer (1991) stated that a morpheme which is solely important in a word is the root when compared to Al-Khuli (1991), for the most, when we talk about English, the word base is in rare case termed as root. According to Al-Atram (1990) in Arabic, be that as it may, the base or stem is unique in relation to the root. In Arabic the root is the first type of the word before any change procedure, and it assumes an imperative part in dialect thinks about (Metri and George 1990). The key points to be looked into are that the powerless roots are the ones with one or two or more long vowels.

According to Al-Khuli (1991), a morpheme or the morphemes which are put together through which an append is notified is termed to be as a Stem. Paice (1994) added to the stem role as a communication of thoughts.

A morpheme called as affix is one the type illustrated by (Al-Khuli 1991; Thalouth and Al-Dannan 1987) as a morpheme which could be added after or prior or within a stem or root postfix or infix or prefix. This is done to form words which are new. The predefined standards that is semantic where the linking takes place between components which seen in Arabic language. Ali (1988) states that the joins increase in number with the addition of components.

The prefixes clearance in English is typically hurtful on the grounds that it can turn around or generally adjust the importance or syntactic capacity of the word. But when we talk about Arabic, the clearance of prefixes is not having any impact on the words supremacy.

A semantic branch with appropriate placement in line with the internal structure of the words is Morphology. As depicted by (Krovetz 1993; Al-Khuli 1991; Hull and Grefenstette 1996), Morphology plays a vital role in the development of word which contains roots other morphology properties like stated by (Spencer 1991; Krovetz 1993; Aref 1997; Hull and Grefenstette 1996) Arabic Morphology is derivational or inflectional.

Inflectional morphology is linked to a given stem which doesn't influence the word's syntactic classification, for example, thing, verb, and so forth. Case, gender, number, strained, individual, state of mind, and voice are a few case of attributes that may be influenced by intonation. Wherein Derivational morphology is connected to

morphemes which can intrigue the syntactic classification of the word. Hence, the similarity and difference between inflectional and derivational is not straightforward.

A process which checks on the various structures of normally used words involves a morphological investigation system which usually combines components as well lexical differences. Here the structure is interior and includes root, stem, joins and samples. The components merge is an exceptionally helpful procedure in numerous common dialect applications (Krovetz 1993), for example, data recovery (El-Affendi 1998), content characterization, and content pressure. According to El-Affendi (1991), when Arabic in particular is taken into account, the fundamental reason for any morphological investigation method is to find the word foundation which is mostly correct but is not inline for the rest.

2.2.1 Arabic Morphological Analyzers

2.2.1.1 Buckwalter (BAMA)

The Buckwalter Arabic Morphological Analyzer (BAMA) is an unreservedly accessible standard analyser for morphological task, brought to existence so as to start of Penn Arabic Treebank labelling (Buckwalter 2002). BAMA's investigation calculation relies on upon its dictionary.

Arround 78,836 lexical sections with 40,218 lemmas are available in the analyser. Firslty, the above is placed in tables with prefixes passages, postfixes and stems and with the use similarity tables the fragments are blended together. The grammatical feature tagset utilized as a part of these lexicon documents is similar to Penn Arabic Treebank tagset. The morphological analyzer forms undiacritized Arabic content, giving back a few conceivable investigations for every word. Its examination calculation creates every conceivable division into stems, postfixes and prefixes. With the process of blending, the tables are examined to find out whether the formation is phonetically credible. The subsequent sifted examinations are yield with diacritization and morphological remarks which is moved up by vocabulary components.

2.2.1.2 MADA-ARZ

Habash et al. (2013) exhibited MADA-ARZ was a system put up for morphological analysis of ARZ which outperformed as MSA tagger (MADA) on ARZ content. This MADA-ARZ alongside does machine interpretation when compared to MADA. The extension of MADA-ARZ included the designing of the MADA-ARZ to deal with Dialect Arabic.

Habash (2007a) takes note of that Arabic morphological assets use distinctive, regularly contrary, with regards to morphological model. The use of Stemmers focussed on segregating the word stems with some analyzers removing roots as well. So as relate and resolve assets which are opposite, Habash (2007a) introduces lexemes in the ALMORGEANA framework which uses components and lexemes to provide morphological study in both directions. This bi-directional morphological assortment helps in preparing undertakings, for example, machine interpretation. The framework uses a vocabulary taking into account word reference information from BAMA, however applies an alternate calculation to accomplish morphological task. Like the Buckwalter analyzer, ALMORGEANA yields a few conceivable morphological investigations for every word in Arabic which is informative. Bu utilizing information from the PennTB converting lexeme -in addition to highlight representation, Habash and Rambow (2005) manufacture a measurable model to rank conceivable investigations utilizing bolster vector machines prepared to perceive individual morphological components. Habash and Rambow (2005) reported accuracy of 98% at all token level and 99% at word level when they experimented with PennTB for Morphological Analysis by using a diminished tagset. The BAMA vocabulary is ought to be the similarity between ALMORGEANA and (Habash, Rambow and Roth 2009b) framework. Habash, Rambow and Roth (2009b) introduces a toolbox with TOKAN and MADA analysers. BAMA is used by MADA morphological analyzer to find the nearest match using weighted mechanism, when compared to TOKAN which can be easily fitted giving morphological segregation of words as per possible plans within tokenization. The toolbox has been utilized for an assortment of work like Machine Translation by (Badr et al. 2008) and (Benajiba et al. 2008; Farber et al. 2008) named entity recognition.

When compared with Buckwalter Analyzer, this toolbox is appealing in light of the fact that it creates a solitary morphological investigation for every Arabic word.

2.2.1.3 MADAMIRA

MADA was worked for MSA; an Egyptian Arabic (EGY) form (MADA-ARZ) was later worked by connecting to the CALIMA EGY analyzer and retraining the models on EGY explanations (Habash et al. 2013). MADA utilizes a morphological analyzer to deliver, for every information word, a rundown of investigations determining each conceivable morphological translation with regards to that particular word, taking care of all possible elements of morphology like POS, lemma, diacritization, and inflectional and clitic. With the use SVM and N-gram, MADA then deliver an expectation, per word in context, for various morphological elements.

The AMIRA toolbox incorporates a tokenizer, chunker, a grammatical feature tagger with a key focus on learning which is directed learning when compared to MADA which is equivocal to morphology. In later forms of AMIRA, a morphological analyzer and MADAMIRA take after the same general configuration as MADA (see Figure 1), with some extra segments propelled by AMIRA.

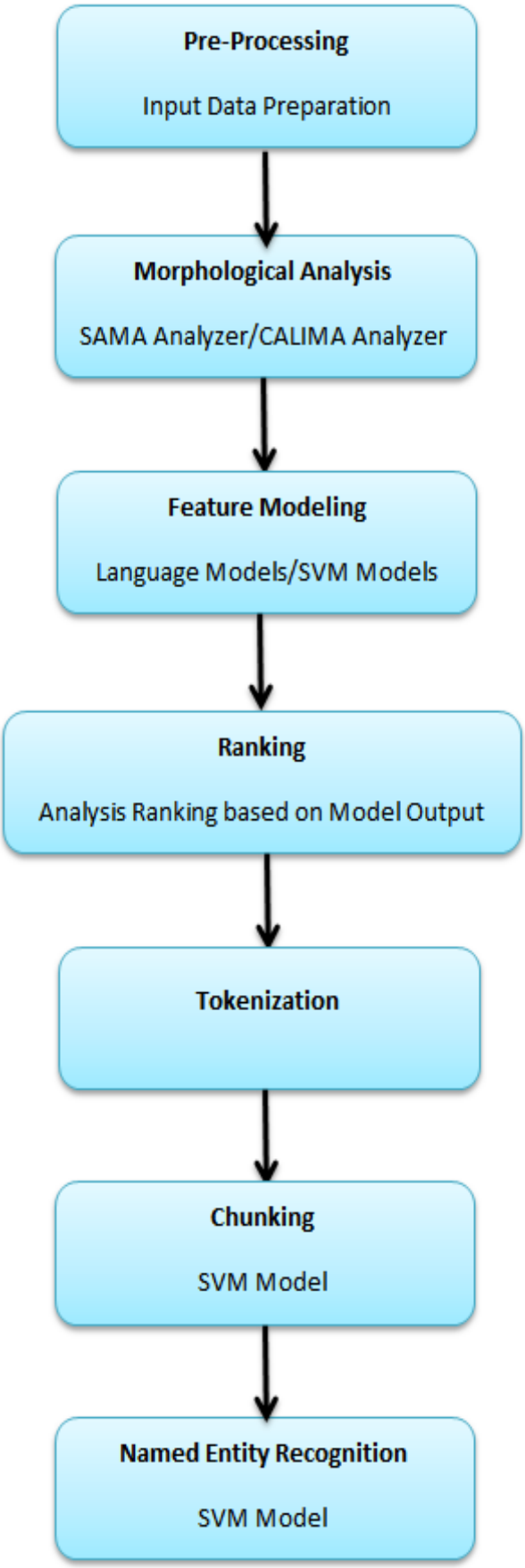


Figure (1) MADMIRA Architecture

Information content (either MSA or EGY) enters the Preprocessor, which cleans the content and changes over it to the Buckwalter representation utilized inside MADAMIRA. The content is then passed to the Morphological Analysis segment, which builds up a rundown of all conceivable examinations (free of connection) for every word. The content and investigations are then moved into a segment called as Feature Modeling wherein the morphological words are predicated. SVMs are utilized for shut class highlights, while dialect models anticipate openclass elements, for example, lemma and diacritic structures. An Analysis Ranking segment then scores every word's examination list in light of how well every investigation concurs with the model forecasts, and after those sorts the examinations in view of that score. The top-scoring investigation of every word can then be passed to the Tokenization segment to produce an altered tokenization (or a few) for the word, as indicated by the plans asked for by the client. The picked examinations along with tokens will be used by the chunker (Phase) to divide content into lumps (utilizing another SVM model). Hence, the named entities are identified with this named entity recognizer. When all the asked for parts have completed, the outcomes are come back to the client. Clients can ask for particularly what data they might want to get; notwithstanding tokenization, base expression lumps and named elements, the diacritic structures, lemmas, sparkles, morphological elements, parts-of-discourse, and stems are all straightforwardly gave by the picked examination.

Notwithstanding copying the ability of the past apparatuses, MADAMIRA was intended to be quick, extensible, and simple to utilize and keep up. MADAMIRA is executed in Java, which gives significantly more prominent rate than Perl and permits new elements to be immediately incorporated with the current code. The outsider dialect model and NLP SVM utilities utilized by MADA and AMIRA were disposed of and supplanted; notwithstanding enhancing execution and making the product less demanding to keep up, this expels the requirement for the client to introduce any extra thirdparty programming. MADAMIRA makes utilization of quick, straight SVMs fabricated utilizing LIBLINEAR (Fan et al. 2008; Waldvogel 2008)

2.2.1.4 SALMA

SALMA stands for Standard Arabic Language Morphological Analysis. SALMA uses all the more fine-grained morphological tagset in view of ideas from the convention of Arabic semantic (Sawalha, Atwell and Abushariah 2013, Sawalha and Atwell 2010).

2.7 million word-root sets are vowelized and 23 Arabic lexicons are put together in SALMA tagger. Arabic text is clarified utilizing an arrangement of 22 morphological elements that incorporate grammatical form, gender orientation, number, individual, case, mind-set, definiteness, voice, accentuation and roots. With the use of a database comprising of 2,700 plus verb examples and 980 things the SALMA tagger scanned for proper root-design sets. Following this, the Morphological items are illustrated.

SALWA tagger when experimented upon by Sawalha et al. (2013) achieved good results with around 2000 words. Similar attempts by Dror et al. (2004) for classical Arabic showed good deductions and by Al-Sulaiti and Atwell (2006) for Modern Arabic showed 98% accuracy for labelling Modern Arabic and for classical Arabic around 90% accuracy.

Hence deducing that programmed morphological analysis can be accomplished, but this varies by utilizing an option set of labels with morphological elements for classical Arabic.

2.2.1.5 OTHERS

Their methodology utilizes limited state registering utilizing FSMs. Al-Badrashiny(2014) exhibited a technique for changing over provincial Arabic (particularly, EGY) written in Arabizi to Arabic script taking after the CODA tradition for DA orthography. They accomplished a 17% blunder diminishment over execution of a formerly distributed work (Darwish, 2013) on a visually impaired test set. Later on, they plan to enhance a few parts of their models, especially FST character mapping, the morphological analyzer scope, and dialect models. They additionally plan to chip away at the issue of programmed distinguishing proof of non-Arabic words. They will be extending the framework to chip away at other Arabic vernaculars. We likewise plan to make the 3ARRIB framework openly accessible

2.3 Arabic POS Tagging

Given a sentence, decide the grammatical form for every word. Numerous words, particularly regular ones, can serve as different parts of discourse. For instance, "book" can be a thing ("the book on the table") or verb ("to book a flight"); "set" can be a thing, verb or descriptive word; and "out" can be any of no less than five distinctive parts of discourse. A few dialects have more such vagueness than others. Dialects with minimal inflectional morphology, for example, English are especially inclined to such equivocalness[2]. Chinese is inclined to such uncertainty since it is a tonal dialect amid verbalization. Such enunciation is not promptly passed on by means of the substances utilized inside the orthography to pass on proposed meaning.

2.3.1 Arabic POS Taggers

For Arabic dialect diverse taggers had been created by examines and organizations. Organizations like RDI, Sakhr and Xerox. These organizations created taggers for business purposes.

2.3.1.1 Half breed tagger

El-Kareh and Al-Ansary (2000) built up a half breed tagger that utilized factual strategy and morphological standards as HMMs. Their tagger performed tests for deciding the tag of the word then the client of the framework could acknowledge the present proposal or supplant it. It was called self-loader. El-Kareh and Al-Ansary tagger was gotten from conventional Arabic punctuation. It accomplished a precision of 90%.

2.3.1.2 Shereen Khoja tagger

Shereen Khoja (2001) built up the APT framework (Automatic Arabic POS-Tagger). This tagger is consolidated from two methodologies: measurable and principle based procedures. The APT is considered as the main tagger framework for Arabic dialect. The tagset which was utilized as a part of APT comprised of 131 labels got from the BNC English tagset. Khoja got her underlying tagset from the linguistic use of Arabic dialect. The APT accomplished a precision of 86 %.

[2] https://en.wikipedia.org/wiki/Natural_language_processing

2.3.1.3 Brill's POS-tagger

Freeman (2001) utilized a machine learning approach and executed Brill's POS-tagger for the Arabic dialect. His tagger depended on a labeled corpus. The corpus was developed physically and it contained more than 3,000 words. A tagset of 146 labels was utilized (Elhadj 2009).

2.3.1.4 POS-tagger utilizing principle based technique

Maamouri and Cieri (2002) built up a POS-tagger utilizing principle based technique. They construct their Arabic tagger in light of programmed explanation yield created by the morphological analyzer of Tim Buckwalter. The created tagger accomplished an exactness of 96%. Diab et al. (2004) built up a POS-tagger for Arabic dialect. This tagger utilized the bolster vector machine (SVM) strategy and LDC (Linguistic Data Consortium). It comprised of 24 tagset.

2.3.1.5 HMM tagger

Banko and Moore (2004) displayed a HMM tagger for Arabic dialect. This tagger accomplished a precision of 96%. Guiassa (2006) built up a tagger that utilized half and half strategy for standard based and a memory-based learning technique and it accomplished a precision of 86%. Couple of specialists were considered the structure of Arabic sentence like Shamsi and Guessoum (2006). They built up an Arabic POS-tagger for un-vocalized content with a precision of 97% utilizing the HMMs. Another POS-tagger considered the structure of Arabic sentence and consolidated morphological examination with Hidden Markov Models (HMMs) created by Elhadj (2014). The acknowledgment rate of this tagger achieved 96%.

2.3.1.6 AMT (Arabic Morphosyntactic Tagger)

Alqrainy (2008) built up a POS-tagger utilizing the rulebased approach. This tagger was called AMT (Arabic Morphosyntactic Tagger). The contribution for AMT was untagged crude incompletely vocalized Arabic corpus. The objective of the tagger was to dole out the right tag to every word in the corpus delivering a POS-labeled corpus without utilizing a physically labeled or untagged lexicon. The AMT comprised of two standard segments: design based principles and lexical and relevant guidelines. The AMT framework accomplished a normal precision of 91%.

2.3.1.7 Genetic algorithm Tagger

Ali and Jarray (2013) utilized the Genetic calculation to build up an Arabic grammatical form labeling. They utilized a decreased tagset as a part of their tagger.

Mohamed and Kubler (2010) created two techniques for Arabic–part of discourse labeling. The two techniques are: Whole word labeling and Segmentation-based labeling.

2.3.1.7 Other POS Taggers

Monirabbassi (2008) built up a grammatical feature tagger for Levantine Arabic (LA), utilizing MSA as the preparation information and testing on a Levantine corpus. The information they use are the MSA Treebank (ATB) from the Linguistic Data Consortium (LDC) and the Levantine Arabic Treebank (LATB), likewise from the LDC. The ATB comprises of 625,000 expressions of daily paper and newswire content, The LATB comprises of 33,000 expressions of translated phone discussions. Because of the lexical, linguistic, and inflectional contrasts amongst MSA and its vernacular, this technique created an exactness of 69.21%. They accomplished a crest precision of 73.28% got by utilizing a blend of the essential model and the two-layer Markov model.

A grammatical feature tagger was built by Monirabbassi (2008) who used Data Consorium and Levantine ATB which contained 33,000 sentences when compared to normal ATM with 625,000 sentences. Another POS tagger which was built by Rambow et al. (2009) for Levantine Arabic (LA), with 73% accuracy measure.

Egyptian Colloquial Arabic POS tagger was brought into existence by (Duh et al. 2002). They use the current assets and information for a few assortments of Arabic. The overall conclusion for this tagger illustrated that MSA prepared tagger performed better on ECA. However, the vital finish of these outcomes is that the True possibility behind MSA outperforming is due to many Arabic Vernaculars using MSA.

TBL approach was used by AlGahtani et al. (2009) for POS tagging by using ATB tagger. Section level tagging was ought to perform better than word-level. Croos-

validation was used to split the corpus into 90:10 ration with 90% training data and the rest 10% testing data achieving 96% accuracy.

2.4 Arabic Treebanks

Throughout the most recent quite a few years, the advancement and utilization of clarified corpora has developed to end up a noteworthy center of exploration for both etymology and computational regular dialect handling. Corpora give the observational confirmation that is utilized to progress different speculations of dialect (Sampson and McCarthy 2005) following the same lines (Kucera and Francis 1967; Hajič et al. 2003) stated the use of corpora as the most important part of electronic vocabularies and grammatical form taggers. Syntactic comment and morphological comment are embedded into Treebanks. The following sub-sections covers a close look at the Penn, Columbia and Prague Arabic treebanks.

2.4.1 The Penn Arabic Treebank

According to Marcus, Santorini and Marcinkiewicz (1993), the PATB was the most elaborative syntactic tagger for any language with a clear guidance towards parsing. For the previous 20 years, PATB is the benchmark for English and ought to be the first choice for Arabic. According to Maamouri et al. (2004) is one of a kind analyser which performs driven analysis of morphology and parsers (syntactic). With the use of Engligh tagset the Arabic annotation becomes easier (Maamouri et al. 2004).

Be that as it may, after the underlying arrival of the treebank a few voting public parsers beforehand produced in general for English language where later fitted to Arabic language. However Arabic Treebank is ought to be a major task with regards to parsing.

According to the explanation rules (Bies and Maamouri 2003), shapes are seen to be common for lonely Arabic verb with joined clitics.

```
(S
  (CONJ wa- وَ)
    (NP-TPC-1
      (PRT
        (NEG_PART -lA لا)
      )
      (PUNC ")
      (NO_FUNC dy$ ديشْ)
      (PUNC ")
    )
    (VP
      (IV3MS+IV+IVSUFF_MOOD:I ya+lotaqiT+u يَلْتَقِطُ)
      (NP-SBJ-1
        (-NONE- *T*)
      )
      (NP-OBJ
        (DET+ADJ+NSUFF_FEM_PL+CASE_DEF_ACC Al+faDA}iy~+At+i الفَضائِيّاتِ)
      )
    )
    (PUNC .)
)
```

Figure (2) Penn Treebank Example

Typically, two phases were involved with PENN ATB tagger. Phase 1 included BAMA vocabulary with which the morphological part was tackled, which resulted in lemmas and other elements of morphology. This further is intrigued by panel of specialist who modify the morphological annotations after labelling through programming, so as to select the most applicable one. During phase2, with the use of Biel's parser a syntactic tree is brought into existence for each sentence using phase1 annotations (Bikel 2004a). Hence the same examination as phase1 is followed wherein the trees are checked by the specialist and amended accordingly as per the requirement.

With the new and varied changes in the PENN ATB has brought about explanations to be more qualified to Arabic's phonetic developments, contrasted with the voting public representation utilized for the Penn Treebank.

2.4.2 Prague Arabic Treebank

Smrž and Hajič (2006) highlight the notes to numerous current semantic speculations and inherent to software engineering and rationale, their association with the investigation of the Arabic dialect and its significance is intriguing as well.

When talking on similar grounds, annotation strategy was used by Hajič et al. (2004) to bring a multi-organised Treebank. The first step as seen in PENN ATB was

morphological annotation using Czech tagger by (Hajič and Hladká 1998). Czech tagger with the utilization of Penn Arabic TB was made available for Arabic, where 11% mistake were reported with nearly 1% division errors for words in Arabic into morphemes. Step 2 was in contrast to PENN ATB where programmed annotations were resolved through morphological study and with syntactic observations. After accomplishing one area with the use of syntactic parser it was made on the available info so as to parse the left over corpus. Lately, these trees were the subsequent reliance trees were modified by the specialist based on the requirement. Figure (3) presents a Prague Arabic Treebank.

Figure (3) Prague Arabic Treebank Example

2.4.3 The Columbia Arabic Treebank (CATiB)

With the utilisation of an average syntactic structure, the Columbia ATiB was built which stood out when compared to PENN ATB AND Prague ATB. Success parameter involved with CATiB is the development of Treebank with strategies on

quick annotation with fewer labels, enabling specialist to cover a large content. Table (1) shows POS tags used in Columbia Arabic Treebank

POS	Meaning
VRB	Verbs
NOM	Nominals (adjective, adverbs, nouns, and pronouns)
PROP	Proper Nouns
PNX	Punctuation
VRB-PASS	Passive-voice verbs
PRT	Particles

Table (1) POS tags used in Columbia Arabic Treebank

Table (2) reveals the tag set (reliance) with regards to several levels wherein except for the modifier label (MOD), the reliance relations depend on understood customary syntactic parts. These labels are effectively justifiable by master annotators acquainted with customary Arabic language structure. The explanation conspires intentionally rejects extra relations utilized for profound labelling, for example, the utilitarian labels for time and accommodate in PENN TB.

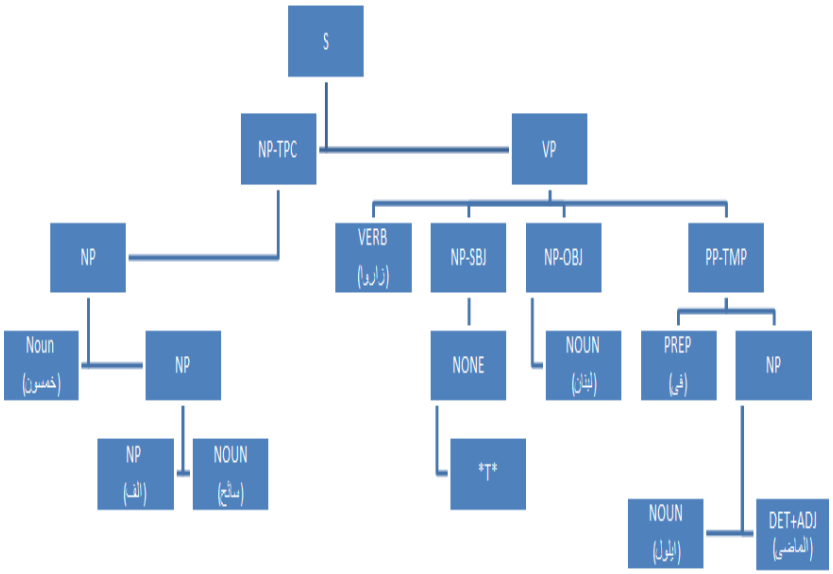
Dependency Tag	Meaning
OBJ	Object
SBJ	Subject
PRD	Predicate
TPC	Topic
TMZ	Specification

IDF	Possessive
MOD	Modifier

Table (2) Dependency tags in the Columbia Arabic Treebank

Plans with respect to ideas from the Arabic etymological custom improves the explanation procedure (Habash et al. 2009a). This looks at to the methodology used for Arabic Corpus for Quran, using tagset into account customary syntax, yet uses an all the more set of fine-grained labels.

Coulumbia ATiB utilizes a natural reliance representation and social marks propelled by Arabic punctuation, for example, tamyīz (particular) and iḍāfa (possessive development) notwithstanding general predicate-contention structure names, for example, subject, item and modifier. Figure 4 depicts the constituency (upper) tree and dependency (lower) tree.



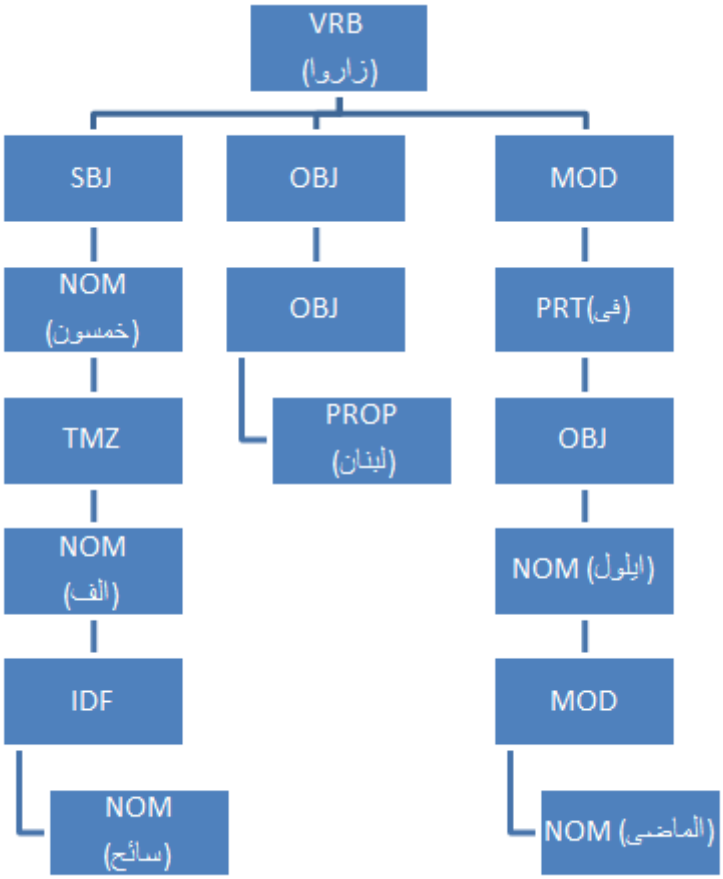


Figure (4) Constituency tree from Penn ATB (upper tree) and a dependency tree from Columbia ATB (lower tree)

As with past Treebank, the explanation technique continues in numerous stages. In the main stage, the content is grammatical feature labeled and morphologically sectioned utilizing the Habash and Rambow (2005) toolbox. The parser was prepared utilizing information from the Penn Arabic Treebank via naturally changing over voting public trees into reliance trees. The upper tree in Figure 4, uses Penn ATB and lower tree is a reliance tree belonging to Columbia Treebank.

Chapter Three

Parsing

3.1 Syntax

Language structure is the branch of phonetic that arrangements with the development of expressions, statements, and sentences. Regardless of the fact that the expressions of the info are presently enriched with a "label" (grammatical feature classification), despite everything they speak to as more than a basic, level series of words. To start understanding the "significance " carried on by proclamation, it is important to clarify first its" structure" i.e., to decide, e.g., the subject and the object of every verb, to comprehend what changing words adjust what different words, what words are of essential significance, and so forth. Doling out such structure to an information proclamation is called "syntactic analysis" or "parsing ".

3.1.1 Syntactic Structure

3.1.1.1 Constituency Structure (Phrase Structure)

In phrase structure words are sorted out into settled constituent

Example:

```
(S
   (NP
      (D This)
      (N Eample)
   )
   (VP
      (V is)
      (VP
         (V Shows)
         (NP
            (D the)
            (A phase)
            (N structure)
         )
      )
   )
)
```

Figure (5) Constituency Structure

Constituent acts as a unit that can show up in better places.

Example:

- ذهب محمد (إلى المدرسة) (بالسيارة)
- ذهب محمد (إلى المدرسة) (بالسيارة)
- ذهب محمد (بالسيارة) (إلى المدرسة)

- **Headed phrase structure**

Examples:

VP ➔ …VB… (Verb phrases usually contains Verb)

NP➔ …NN… (We call this rule noun phrase because it contains noun)

3.1.1.2 Dependency Structure

Dependency structure demonstrates dependency between words inside the sentence, arrows used to demonstrate this relation:

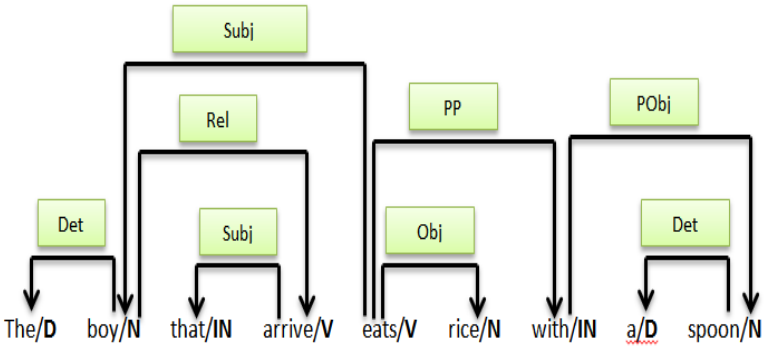


Figure (6) Dependency structure

3.2 Parsing Approaches

3.2.1 Rule Based Parsing

In such parsers, information about the syntactic structure of a dialect is composed as etymological guidelines, and these tenets are connected by the parser to info content portions keeping in mind the end goal to create the subsequent parse trees. Data

about individual words, for example, what POS they might be, is generally put away in an online word reference, or "vocabulary," which is gotten to by the parser for every word in the information content preceding applying the phonetic principles.

Despite the fact that administer based parsers are broadly utilized as a part of genuine, working NLP frameworks, they have the weakness that broad measures of (lexicon) information and work (to compose the guidelines) by exceedingly talented language specialists are required so as to make, enhaÿ@e, and look after them. This is particularly valid if the parser is required to have "wide scope", i.e., on the off chance that it is to have the capacity to parse NL content from a wide range of areas (what one may call "general" content).

3.2.2 Statistical Parsing

In the most recent couple of years, there has been expanding action in the computational etymology group concentrated on making utilization of measurable techniques to gain data from substantial corpora of NL content, and on utilizing that data as a part of factual NL parsers. Rather than being put away in the conventional type of word reference information and syntactic guidelines, semantic learning in these parsers is spoken to as factual parameters, or probabilities. These probabilities are ordinarily utilized together with less complex, less indicated, word reference information and/or rules, in this way assuming the position of a significant part of the data made by talented work in principle based frameworks.

Points of interest of the factual methodology that are asserted by its advocates incorporate a noteworthy diminishing in the measure of guideline coding required to make a parser that performs sufficiently, and the capacity to "tune" a parser to a specific kind of content essentially by separating measurable data from the same sort of content. Maybe the hugest weakness seems, by all accounts, to be the prerequisite for a lot of preparing information, regularly as huge NL content corpora that have been commented on with hand-coded labels determining parts-of-discourse, syntactic capacity, and so forth.

3.3 Statistical Parsing So Far

Unmistakable objectives for exploration on statistical parsing:

• Construct a parser that can predict the grammatical structures of a sentence with the most elevated exactness: the probabilistic parser will probably discover the syntactic parsing that boosts P ((1/4) |S), where (1/4) is a syntactic shape and S is the sentence being referred to.

. Construct a dialect representation for assignments for example, discourse acknowledgment,

a version that doles out chances to strings in a dialect, using P (S, (1/4)). it's far imperative to observe that P (S, (1/4)) is huger than P (S, (1/4)) given that we are able to determine the second from to begin with, yet no longer the other way round (charniak 2000). Likewise,

P(s, ¼) P(S, (1/4)) is adaptable because it we can utilize it to manufacture a dialect display or choose the parsing that have the most noteworthy likelihood. A lot of researches that have focused on constructing the factual parser with relaying on the dialect representations to catch the grammatical imperatives of the dialect. It was fascinating to research the existence of a relationship between the execution of a factual parser and the dialect model based on that parser. Past examination on factual parser could be grouped as managed and also non-supervised techniques: the previous assessment factors utilizing an arrangement of sentence/parsing sets like preparing information, the last utilize unprocessed content as preparing information.

In light of the technique that the parser creates a syntactic structure for the sentence, the recent factual parsers can likewise be delegated a probabilistic models and non-probabilistic models. The Statistical generative models depend on the possibility that a sentence can be produced by a statistical model. At the point if we characterizing the probabilistic model for a dialect, customarily this probabilistic model is fixing to the tenets of a syntax then it will ensure that the exclusive strings created by the sentence structure get likelihood which is more than zero.

3.3.1 Probabilistic Context-free Grammar Parsers

3.3.1.1 Context-Free Grammars

G = (T, N, S, R)

T is a set of terminal symbols

N is a set of nonterminal symbols

S is the start symbol (S ∈ N)

R is a set of rules/productions

A grammar G generates a language L

We have a classification which changes as a succession of different classifications. And after that inevitably this records to what are called terminal images which are words. Thus utilizing this language structure, we can create sentences. So we begin with the begin image S and afterward we can extend down utilizing any of the standards of the language structure.

Example:

S → NP VP

VP → V NP

VP → V NP PP

NP → NP NP

N → السيارة

N → ابراهيم

V → ذهب

P → مع

3.3.1.2 Probabilistic Context-Free Grammar

Every rule is attached with its probability

G = (T, N, S, R, P)

T is a set of terminal symbols

N is a set of nonterminal symbols

S is the start symbol (S ∈ N)

R is a set of rules/productions of the form X → γ

P is a probability function

$$\forall X \in N, \sum_{X \to \gamma \in R} P(X \to \gamma) = 1$$

P: R $\to$ [0,1]

A grammar G generates a language model L.

$$\sum_{\gamma \in T^*} P(\gamma) = 1$$

Example:

S $\to$ NP VP	(1.0)
VP $\to$ V NP	(0.6)
VP $\to$ V NP PP	(0.4)
NP $\to$ NP NP	(0.1)
N $\to$ السيارة	(1.0)
N $\to$ ابراهيم	(0.6)
V $\to$ ذهب	(0.3)
P $\to$ مع	(1.0)

Probabilistic connection free sentence structures abbreviated as PCFG is a characteristic begin for the statistical parsing techniques of common dialect. Pastry specialist et al. (1992; 1979), portray within outside calculation, The non-supervised methodology for getting the PCFG creation guideline probabilities. We can find a different vital explanation behind the prominence of constructing the PCFG parsers: the global accessibility of the CFG parsed tree bank corpus. this large parsed corpus is required for preparing administered statistical models and evaluating parsing precision for managed and non-supervised statistical models. A generally accessible and usually utilized CFG tree bank corpus is Penn Treebank (Marcus et al. 1993). They published three versions of Penn Treebank starting with the first version at 1991.

The second published version fixes the issues of the first version by increasing the accuracy and consistency of the tags and also adding extra explanations to the tags, those changes made the parsed sentences more understood. The third form includes

extra components, for example, disfluency explanations for the Switchboard areas (which depend on interpreted discourse gathered from phone conversations limited to around 70 unique themes).

Despite the fact that a PCFG is a well-known formalize grammar structure; some researches considered that PCFG as a poor model for human language of common dialect in a few regards. For example, in the non-supervised examples, it will fail if we trained PCFGs with inside-outside technique to produce the expected syntactic structure, also the statistical models results from parsing PCFG do not minimize the perplexity. In supervised examples also PCFG fails until they add extra context. A lot of work had been done to improve the PCFGs for example improve the non-supervised trained cases and add extra information to the PCFG.

Lexicalized PCFG Parsers

Like examined at start of the section, the researches objectives of a parser vary from those trying to create a dialect model. Be that as it may, there are some significant fundamental cooperation between the two errands, proposing that by concentrating just on one arrangement of objectives, we may free some synergistic advantages that would come about because of considering both arrangements of objectives together. Research on dialect displaying for discourse acknowledgment has found that in spite of the fact that n-grams models are a good example of a grammatical models, those models are extremely viable if we utilized them as a dialect models to the discourse acknowledgment undertakings. Then again, how to manufacture excellent dialect models from high precision parsers is still under scrutiny. Focused organized dialect models normally incorporate parameters for displaying either variably or expressively. PCFGs research suggest that a construct (parameterization) simply in light of auxiliary relations won't fill in and also one that likewise incorporates word character or matches of words with a specific basic connection. As stated by Hindle et al. (1991) about lexical conditions which are of high value because of prepositional expression before 1990, as Marcus (1990) revealed the strategies which uses lexical conditions to be pulled up together to accomodate full parsing models. Looking ahead from the time of Hindle et al. (1991) most of the investigations focused on directed approaches in parsing models. These models incorporate the lexicalized PCFGs of (Charniak 1992; 1997; Collins (1999); Roark (2001)).

3.3.1.2.1.1 Collins' Three Generative, Lexicalized PCFG Parsers

Three parsing models for lexicalized PCFG were introduced by (Collins 1999). Based on Model 1, PCFGs was elongated to lexicalize linguistic which utilizes word, POS tag and each non-terminal. The lexicalized PCFG revolves around name production in the first place, followed by creation of modifiers and lastly, modifiers are created to one side (head). Words separation relationship between head-modifier is put together in the model.

Model2 was generated based on Model1, wherein parameters based on likelihood with regards to headwords were added. The key constraint was the sub-categorization keeping in mind headword. The sub-categorization was keenly involved in the left/right modifiers.

Model3 outburst from Model1, wherein development using a master plan was used in GPSG (Generalized Phrase Structure Grammar). Gazdar et al. (1985), to empower the parser to use co-indexation on account of wh-development. Wh-development in Model3, was accomplished with the addition to non-terminals in parse tree. Every one of the three models utilized a graph parser to locate the most extreme likelihood tree for every info sentence. Model2 and Model3 performed better than Model1 when Collins models were tested on Wall Street Journal PTB corpus, wherein, Model 1 87% named accuracy was achieved when compared to 88% on Model2 and 88% on Model3.

3.3.1.2.1.2 Roark's Top-down PCFG Parser

Based on left-right parsing, Roark (2001) introduced a dialect model depended on the top-down parser (probabilistic). The key part of Roark (2001) PCFG parser was left-right parsing which enabled word probabilities outperforming with the inferences when compared to other models. The execution of both his parser and dialect model shows guarantee (Roark 2001). Top-down parsing calculation constructs an arrangement of established tree(parse) left to right. As well a point which came out in reverse direction includes a parser which is left-to-right and induction is missing then prefix string probabilistic structure is un-rooted. On the contrary a parser with inductions but not left to right, figure out probabilities of words from past.

Roark's parser uses a direction in top-down pattern to deduce using probabilistic model. The way the association of the settled prefix induction is seen, the majority of the molding occasions extricated starting from the top left context has been indicated; henceforth, a contingent probabilistic model utilizing these occasions won't add to the hunt multifaceted nature. When tested on Wall Street Journal PTB, 85% named accuracy was achieved by Roark's parser which was less when compared to Collins Parser.

3.3.1.2.1.3 Charniak's Maximum-Entropy Inspired PCFG Parser

A base up model for PCFG which is lexicalized by putting together head word data into the parser Charniak (1992; 1997). Here a non-terminal called as guardians plays a major role in overall performance with guardian's headword, guardian's non-terminal and the kind of its guardian's headword. 89% named accuracy was achieved by Charniak (1992; 1997).

3.3.1.2.1 Partially Supervised Training of PCFGs

Pereira and Schabes (1992) broadened within outside calculation Baker (1979) so that an incompletely sectioned corpus can be utilized as a part of a semi-managed way to train a parallel stretching PCFG. There are two extraordinary advantages of this technique contrasted with Baker's inside-outside calculation (Baker 1979). To begin with, the new calculation is more effective. On the off chance that the corpus is completely sectioned, utilizing the new calculation with time as O(n) when compared to the time of Baker's O(n3). The other major realization here is with the data which is accommodated in the fractional sectioning alongside the more definite structure inside the sectioning in an unsupervised way, in this manner the expansion offers more all-inclusive statement and adaptability than Baker's calculation. At the point when assessed on sectioning precision resulted in 90% precision which when compared with Baker's Calculation which is 37. %. The key underlying diminishing factor with Baker's calculation is the technique utilized by Baker, which is unsupervised resulted in very poor result.

Using a manually made structure for a particular area, Black et al. (1992) introduced a strategy with the use probabilities and calculations resulted in 75% accuracy.

Wherein Schabes et al. (1993), highlighted an accuracy of 90% with parallel elaborating trees which allow calculations for full parsers.

3.3.1.2.2 Strengthening Structural Sensitivity

The issue with standard PCFG parser and also with partner probabilities was tackled by Briscoe and Carroll (1993), with the introduction of LR parser in line with Alvey Natural Language Tools (ANLT) parser. The drawback with Stanford PCFG was the loss of structure in the parse inference because of which the parser can't recognize deductions in which the same tenet is connected numerous times in various ways along with this the connection reliance data is not very much demonstrated. With regards to the LR parser, the rules were easily deployed to the LR state within the table (LR Parse). LR parser dealt with the CF runs based on linguistic and estimated name-esteems. Here the key point that was of main focus was the generation of a LR parse table was developed in view of the CF runs.

With the use of nearly 200 plus sentences, Briscoe and Carroll (1993) and developed a framework which gave 76% accuracy when tested on 150 sentences being parsed. Bod (1993) display fundamentally develops auxiliary data of conventional PCFGs to incorporate bigger tree pieces. Despite the fact that Bod (1993) did not utilize word character, with the strategy stretches to deal with syntaxes which are lexicalized syntaxes. In any case, the computational multifaceted nature of the parsing calculation may make it hard proportional up to more mind boggling corpora or incorporate word personality data. Goodman (1996) endeavored to execute a more effective DOP calculation.

3.3.1.3 Parsers for More Lexicalized Grammars

Church (1990) construct absolutely in light of auxiliary relations, for example, PCFGs, ought to be less fruitful than depend on sets with very fundamental relations (Marcus 1990). In this part, the punctuations (lexicalized) with a comparison on parsers is presented.

3.3.1.3.1 Stochastic Tree-Adjoining Grammar Parsers

Schabes(1992) built up a TAG model similar to (Resnik 1992) based on Stocahstic parsing, wherein the EM calculation was elongated to perform

tags(Stochastic) (Baker 1979). Joshi and Srinivas (1994) built up the instrument of super tagging as an initial phase which revolved around developing Trees based on adjoining grammars.

3.3.1.3.2 Hybrid Parsers of CFGs and DGs

Capturing the best features of CFGs and DGs, experts have put forward distinguished parsers.

3.3.1.3.3 Probabilistic Link Grammar Parsers

The variance in probabilistic connection of sentence structure by Lafferty et al. (1992) presented a model which was based on top-down parsing calculation with the addition probabilistic analysis. However; in any case, the parsing execution has not been assessed.

A. Collins' Probabilistic Parser Based on Bigram Lexical Dependencies

The model utilizes a separation measure between ward words together with accentuation data as extra contingent components. Around 85% accuracy was achieved by parser when tested on the wall street journal PennTB. Collins estimated that a more tightly incorporation of labeling and modules being parsed will enhanve the accuracy of parsing as well as labelling.

B. Chelba's Statistical Structured Model

Chelba et al. (1999) built up probabilistic parser with calculation for EM based on re-estimation.

3.4 Arabic Statistical Parsing Models

3.4.1 Arabic Dependency Parsing

Based on free word appeal like in the case of Arabic, the latest parsing job has been focused on reliance linguistic use. CoNLL (2007) based on the task that test parsers (factual reliance) some dialects were tested (Nivre et al. 2007a). Cutting edge parsers for Modern Arabic were tried in the mutual undertaking utilizing information from the Prague Arabic Treebank (). The Arabic content was provided with best quality level morphological explanation, including grammatical form labels, division and components commented on from the treebank. The same methodology is utilized as a

part of this postulation, where highest quality level morphological explanation is likewise accepted as contribution for assessing another Classical Arabic parser.

3.4.2 Dual Dependency-Constituency Parsing

Inside distributed writing, past work that most nearly looks like the half and half reliance supporters parsing calculation presented in this study is based on the methodology by (Hall and Nivre 2008) and for Swedish (Hall, Nivre and Nilsson 2007b). Be that as it may, as opposed to the cross breed parser, their joined model yields two parse trees for an info sentence, giving unmistakable explanation to reliance and voting public representations. They additionally depict their methodology as cross breed parsing.

Chapter Four

Implementation: Parsers used in the Experiment

4.2 Previous Work

The key findings of Apple Pie, Collins/Bikel's, Charniak's, and the Stanford parser are presented under this chapter. A curiosity of this work is the assessment of the parsers along new measurements, for example, vigour and crosswise over sort, specifically account and explanatory. For the last part we built up a best quality level for assessing parsers with chose account and descriptive writings from the TASA corpus. No huge impact, not as of now caught by variety in the length of sentence is seen to have a significance. Other than this one of the major worry is the parser assessment concerning specific mistake sorts that are expected to be risky for further preparing of the subsequent parses. The aftereffects of this coordinated assessment affirmed the robotized parser assessment. However, the yield of both strategies does not relate by any stretch of the imagination, so that neither one of the methods makes the progress of the other nor both added to the general assessment of the parsers.

Charniak's parser proved to be the best among all tried parsers with regards to syntactic data when taken from legitimate source. It should be paid attention to, obviously, that parsers not assessed here may beat this competitor. Likewise noted ought to be that Charniak has as of late enhanced his parser further (Charniak and Johnson 2005).

(Green, D. Manning 2010) built up higher parsing baselines, we have demonstrated that Arabic parsing execution is not as poor as already thought, but rather stays much lower than English. We have portrayed sentence structure state parts that essentially enhance parsing execution, indexed parsing mistakes, and evaluated the impact of division blunders. With a human assessment we likewise demonstrated that ATB between annotator assertion stays low in respect to the WSJ corpus. Our outcomes propose that so as to benefit from the current models (parsing) requires consistency as well improvised syntactical annotation.

(Maamouri et al. 2008) worked on the Arabic Treebank (ATB), discharged by the Linguistic Data Consortium, which contains various comment documents for every source record, due to some degree to the part of diacritic incorporation in the tagging procedure. The information is made accessible in both "vocalized" and "un-vocalized" shapes, with and without the diacritic imprints, individually. Much parsing work with the ATB has utilized the un-vocalized structure, on the premise that it all the more nearly speaks to "this present reality" circumstance. We bring up a few issues with this use of the un-vocalized information and clarify why the un-vocalized structure does not truth be told speak to "certifiable" information. This is because of a few parts of the Treebank comment that as far as anyone is concerned have at no other time been distributed.

(Seth Kulick et al. 2006) found that the past work has exhibited that the execution of current parsers on Arabic is far beneath their execution on English or even Chinese, which thus hurts execution on NLP errands that utilization parsing as information. They investigate a portion of the issues required in this distinction, and they concentrate on the Collins parsing model as actualized in the Bikel parser. The corpus utilized for the trials is the Arabic Treebank. They bunch these issues in three ways. To begin with, it is vital when contrasting Arabic parsing execution with different dialects that the examination be a reasonable one; consequently, they first

talk about some issues around assessment and demonstrate that present Arabic parsing execution is not exactly as terrible as already thought. Second, we show a few alterations to the parser which give unassuming expansions in execution. At long last, they investigate further contrasts between the Arabic Treebank and the Penn Treebank and propel a few theories in the matter of why parsers experience issues with Arabic.

(Comelles, E., Arranz, V. and Castellon, I. 2010) have exhibited an assessment among a few body constituency and dependency parsers to utilize the best framework towards the advancement of a programmed Machine Translation assessment metric. Accordingly, they have played out a manual assessment with a specific end goal to distinguish the etymological mistakes made by the parsers and see whether such sort of data could be dependably connected to the assessment of Machine Translation yield, which will be a piece of their next analyses. They have given results on this assessment and they have concentrated on the most widely recognized sorts of etymological mistakes made by the parsers. After a nearby examination of the blunders, they presume that a standout amongst the most widely recognized and essential mistakes made by both voting public and reliance parsers are identified with the PoS and Phrasal classification task precision, and in addition the recognizable proof of the extent of an expression. It is additionally worth seeing that most parsers examine accurately complex structures, for example, relative statements, though a large portion of them come up short in dissecting basic structures, for example, SVOiOd sentences which contain straightforward expressions. This could be clarified by the diverse space and kind of syntactic structures in their corpus when contrasted with that used to prepare measurable parsers. Since their point is to proceed with their work on the advancement of the information based metric, they investigate encourage how the semantic issues reported here could influence the metric itself. Given that our assessment was extremely strict as far as not tolerating any sort of mistake and not considering any worth averaging, as some different assessments do.

4.2 Collins's (Bikel's) parser

Based on the probabilities among head-words within the parse tree is the CBP (Collins' factual parser) Collins (1996, 1997). The rules in the tree further help in

designating hub to a head -youngster. In the head off-spring, the lexical leader of hub turns into the lexical leader of the guardian hub. Connected with every hub is an arrangement of conditions inferred in the escorting manner. With this, there is a resultant youngster with non-head, wherein an addition in reliance to the set takes place with a triplet containing the guardian non-terminal, non-head-kid non-terminal and the head-kid non-terminal. Based on the programming style, the CYK calculation is dependant for parsing. (Bikel, 2004b) introduced an upgraded version of Collins‟ parser for reimplementation, a very much illustrative model for language structure (Collins 1999).

4.3 Stanford Parser

Klein and Manning (2003) uses a structure which is free of connection within sentence with parameters used are less in number and takes after CYK graph parser to generate parses by taking the most applicable parse tree for that sentence. The lexicalized and un-lexicalised rendition was seen to test.

Chapter Five

Evaluation and Results

5.1 Statistical Parsers Evaluation

PARSEVAL Measures

PARSEVAL measures are the standard method for assessing parser quality. There are three PARSEVAL measures: Precision, Recall and F-Score. The measure of right constituents in a parser is accuracy. A constituent is thought to be right on the off chance that it coordinates a constituent in the Gold Standard (number of right constituents (yield) in parser yield isolated by number of constituents in the parser yield). Review is the relative measure of right constituents contrasted with the Gold

Standard parse (number of constituents from the highest quality level (yield) that can be found in the parser yield isolated by the quantity of constituents in the best quality level). F-Score is the weighted accumulation of accuracy and review

Precision = # Correct Constituents / # Constituents in parser output

Recall = # Correct Constituents / # Constituents in gold standard

F-Score = 2 * Precision * Recall / Precision + Recall

5.2 Evaluation of Bikel and Stanford Parser

5.2.1 Pre-Processing:

(1) Arabic Tree Bank Part 3 is used for testing and training, this version of ATB contains 22524 parsed Trees

Example:

(S (CONJ wa-)(VP (PV+PVSUFF_SUBJ:3MS -kAn+a)(PP-PRD (PREP min)(ADJP (DET+ADJ+CASE_DEF_GEN Al+mumokin+i)(NOUN+CASE_INDEF_ACC jid~+AF)))(SBAR-SBJ (SUB_CONJ >an)(S (VP (IV3MS+IV+IVSUFF_MOOD:S ya+HoSul+a)(NP-SBJ (NOUN+CASE_INDEF_NOM hujuwm+N))(SBAR-ADV (SUB_CONJ law)(S (VP (PRT (NEG_PART lam))(NO_FUNC nbdA)(NP-SBJ (-NONE-))(PP-CLR (PREP bi-)(NP (NOUN+CASE_DEF_GEN -qaSof+i-)(POSS_PRON_3MP -him)))))))))(PUNC ")(PUNC .))*

(2) Java program is developed that can convert the above tree to grammar rules as shown:

S ---> CONJ VP PUNC PUNC ***(non-terminal Rule)***

CONJ ---> wa- ***(terminal Rule)***

VP ---> PV+PVSUFF_SUBJ:3MS PP-PRD SBAR-SBJ ***(non-terminal Rule)***

PV+PVSUFF_SUBJ:3MS ---> -kAn+a

PP-PRD ---> PREP ADJP

PREP ---> min

ADJP ---> DET+ADJ+CASE_DEF_GEN NOUN+CASE_INDEF_ACC

DET+ADJ+CASE_DEF_GEN ---> Al+mumokin+i

NOUN+CASE_INDEF_ACC ---> jid~+AF

SBAR-SBJ ---> SUB_CONJ S

SUB_CONJ ---> >an

S ---> VP

VP ---> IV3MS+IV+IVSUFF_MOOD:S NP-SBJ SBAR-ADV

IV3MS+IV+IVSUFF_MOOD:S ---> ya+HoSul+a

NP-SBJ ---> NOUN+CASE_INDEF_NOM

NOUN+CASE_INDEF_NOM ---> hujuwm+N

SBAR-ADV ---> SUB_CONJ S

SUB_CONJ ---> law

S ---> VP

VP ---> PRT NO_FUNC NP-SBJ PP-CLR

PRT ---> NEG_PART

NEG_PART ---> lam

NO_FUNC ---> nbdA

NP-SBJ ---> -NONE-

*-NONE- ---> **

PP-CLR ---> PREP NP

PREP ---> bi-

NP ---> NOUN+CASE_DEF_GEN POSS_PRON_3MP

NOUN+CASE_DEF_GEN ---> -qaSof+i-

POSS_PRON_3MP ---> -him

PUNC ---> "

PUNC ---> .

(3) After this I extract the terminal rules to separate the lexicon to convert the parsed trees to the original Arabic sentence (Buckwalter Arabic transliteration) and also the (+) sign inside the lexicon is removed to be like the following:

*wa- -kAna min Almumokini jid~AF >an yaHoSula hujuwmN law lam nbdA * bi- -qaSofi- -him " .*

(4) After that the ATB is divided into two parts 90% for training and 10% for testing using random sampling.

(5) Evaluation Program:

Java program is created to automatically calculate the Precision, Recall, F-Score, and tag Accuracy for each parsed tree and the overall performance average in the testing sample Example:

Gold Standard File (2 Trees) as an Input:

*(S (CC wa-)(VP (VBD ->aDAfat)(NP (-NONE- *))(PUNC ")(SBAR (IN <in~a)(S (NP (NNP stiyfin)(NNP kinot))(VP (VBP yanotamiy-LRB-null-RRB-)(NP (-NONE- *T*))(PP (IN <ilaY)(NP (NP (NN fi}apK))(SBAR (WHNP (-NONE- *0*))(S (VP (PRT (RP lA))(VBP tasotafiydu)(NP (-NONE- *T*))(RB (JJ kaviyrAF))(PP (IN min)(NP (NNS xadamAti-)(PRP$ -nA)))(SBAR (IN li>an~a-)(S (NP (PRP -hA))(VP (VBP taDum~u)(NP (-NONE- *T*))(NP (NP (NN >a$oxASAF))(SBAR (WHNP (-NONE- *0*))(S (VP (VBP yuwAjihuwna)(NP (-NONE- *T*))(NP (NP (NN ma$Akila))(JJ (JJ SaEobapF)(NN jid~AF)))(PP (IN li-)(NP (NN -tanoZiymi)(NP (NN HayAati-)(PRP$ -him))))))))))))))))))(PUNC ")(PUNC .))*

*(S (S (CC wa-)(VP (VBD -$ab~a)(NP (NN HariyqN))(PP (IN fiy)(NP (NN Aldab~Abapi)))))(S (RB (RB vum~a))(VP (VBN nuqila)(NP (NP (NN Aljunuwdu))(SBAR (WHNP (WP Al~a*iyna))(S (VP (VBD kAnuwA)(NP (-NONE- *T*))(PP (IN fiy)(NP (NN dAxili-)(PRP$ -hA)))))))(NP (-NONE- *))(PP (IN <ilaY)(NP (NN Almusota$ofaY)))(PP (IN li-)(NP (NN -AlEilAji)))))(PUNC .))*

Parser output trees (2 Trees) as an Input:

*(S (CC wa-) (VP (VBD ->aDAfat) (PUNC *) (PUNC ") (SBAR (IN <in~a) (S (NP (NNP stiyfin) (NNP kinot)) (VP (VBP yanotamiy-LRB-null-RRB-) (NP (NP (NN *T*)) (PP (IN <ilaY) (NP (NN fi}apK) (JJ *0*))) (SBAR (S (VP (PRT (RP*

*lA)) (VBP tasotafiydu) (NP (NN *T*)) (NP (NP (JJ kaviyrAF)) (PP (IN min) (NP (NNS xadamAti-) (PRP$ -nA)))) (SBAR (IN li>an~a-) (S (NP (PRP -hA)) (VP (VBP taDum~u) (NP (NP (CD *T*) (NN >a$oxASAF) (JJ *0*)) (SBAR (S (VP (VBP yuwAjihuwna) (NP (NN *T*) (NP (NP (NN ma$Akila)) (ADJP (JJ SaEobapF) (NN jid~AF)))) (PP (IN li-) (NP (NN -tanoZiymi) (NP (NN HayAati-) (PRP$ -him))))))))))))))))) (PUNC ") (PUNC .))*

*(S (CC wa-) (VP (VBD -$ab~a) (NP (NP (NN HariyqN)) (PP (IN fiy) (NP (NP (NN Aldab~Abapi)) (SBAR (S (ADVP (RB vum~a)) (VP (VBN nuqila) (NP (NP (NN Aljunuwdu)) (SBAR (WHNP (WP Al~a*iyna)) (S (VP (VBD kAnuwA) (VP (VBP *T*) (PP (IN fiy) (NP (NN dAxili-) (PRP$ -hA))) (PUNC *) (PP (IN <ilaY) (NP (NN Almusota$ofaY))) (PP (IN li-) (NP (NN -AlEilAji))))))))))))) (PUNC .))*

Evaluation File (2 Trees) output :

(1) Percision=0.685714285714286 Recall=0.63157894736842 F1=0.657534246575342 Tag Accuracy=0.806

(2) Percision=0.541666666666667 Recall=0.56521739130435 F1=0.553191489361702 Tag Accuracy=0.9

Over All Percision = 0.6136904761904762

Over All Recall = 0.5983981693363845

Over All F1 = 0.6053628679685223

Over All Tag Accuracy = 0.8527777777777779

Two sets of files are created with and without Diacritic because parses will be evaluated with and without Diacritic:

Set One (with Diacritic):

- Buckwalter Arabic transliteration of testing sample file (with Diacritic)

  Example :

*wa- -$ab~a HariyqN fiy Aldab~Abapi vum~a nuqila Aljunuwdu Al~a*iyna kAnuwA *T* fiy dAxili- -hA * <ilaY Almusota$ofaY li- -AlEilAji .*

This file will be used as an input to the parser to generate the parsed trees that will be compared to the gold standard trees to evaluate the parser.

- Gold Standard Trees File (with Diacritic) 10% of ATB

  Example:

  (S (S (CONJ wa-)(VP (PV+PVSUFF_SUBJ:3MS -$ab~a)(NP-SBJ (NOUN+CASE_INDEF_NOM HariyqN))(PP-LOC (PREP fiy)(NP (DET+NOUN+NSUFF_FEM_SG+CASE_DEF_GEN Aldab~Abapi)))))(S (ADVP-TMP (ADV vum~a))(VP (PV_PASS+PVSUFF_SUBJ:3MS nuqila)(NP-SBJ-2 (NP (DET+NOUN+CASE_DEF_NOM Aljunuwdu))(SBAR (WHNP-3 (REL_PRON Al~a*iyna))(S (VP (PV+PVSUFF_SUBJ:3MP kAnuwA)(NP-SBJ-3 (-NONE- *T*))(PP-LOC-PRD (PREP fiy)(NP (NOUN+CASE_DEF_GEN dAxili-)(POSS_PRON_3FS -hA)))))))(NP-OBJ-2 (-NONE- *))(PP-DIR (PREP <ilaY)(NP (DET+NOUN Almusota$ofaY)))(PP-PRP (PREP li-)(NP (DET+NOUN+CASE_DEF_GEN -AlEilAji)))))(PUNC .))

6) This file will be used to compare the parser output file against it.

- Training File (with Diacritic) 90% of the ATB

  This file contains the parsed trees that will be used to train Sanford and Bikel Parsers.

  Set two (without Diacritic):

- Buckwalter Arabic transliteration of testing sample file (without Diacritic)

  Example :

  *w $b Hryq fy AldbAbp vm nql Aljnwd Al*yn kAnwA *T* fy dAxl hA * <lY Almst$fY l AlElAj .*

- Gold Standard Trees File (without Diacritic) 10% of ATB

  Example:

  (S (S (CONJ w)(VP (PV+PVSUFF_SUBJ:3MS $b)(NP-SBJ (NOUN+CASE_INDEF_NOM Hryq))(PP-LOC (PREP fy)(NP (DET+NOUN+NSUFF_FEM_SG+CASE_DEF_GEN AldbAbp)))))(S (ADVP-TMP (ADV vm))(VP (PV_PASS+PVSUFF_SUBJ:3MS nql)(NP-SBJ-2 (NP (DET+NOUN+CASE_DEF_NOM Aljnwd))(SBAR (WHNP-3 (REL_PRON

*Al*yn))(S (VP (PV+PVSUFF_SUBJ:3MP kAnwA)(NP-SBJ-3 (-NONE- *T*))(PP-LOC-PRD (PREP fy)(NP (NOUN+CASE_DEF_GEN dAxl)(POSS_PRON_3FS hA)))))))(NP-OBJ-2 (-NONE- *))(PP-DIR (PREP <lY)(NP (DET+NOUN Almst$fY)))(PP-PRP (PREP l)(NP (DET+NOUN+CASE_DEF_GEN AlElAj))))))(PUNC .))*

- Training File (without Diacritic)

Java program is created to transfer Arabic Text from and to Buckwalter Arabic transliteration

Example (with Diacritic):

*wa- -$ab~a HariyqN fiy Aldab~Abapi vum~a nuqila Aljunuwdu Al~a*iyna kAnuwA *T* fiy dAxili- -hA * <ilaY Almusota$ofaY li- -AlEilAji .*

وَشَبَّ حَرِيقٌ فِي الدَبّابَةِ ثُمَّ نُقِلَ الجُنُودُ الَّذِينَ كانُوا فِي داخِلِ ها إِلَى المُسْتَشْفَى لِ العِلاجِ.

Example (without Diacritic) :

*w $b Hryq fy AldbAbp vm nql Aljnwd Al*yn kAnwA *T* fy dAxl hA * <lY Almst$fY l AlElAj .*

و شب حريق في الدبابة ثم نقل الجنود الذين كانوا في داخل ها إلى المستشفى ل العلاج .

5.2.2 Stanford Evaluation

Training

90% of ATB is used (20249 tree) in training to generate the Arabic Parsing Model. Two models are created one for the diacritic and non- diacritic Arabic texts using the following commands:

With Diacritic

java -Xmx6g edu.stanford.nlp.parser.lexparser.LexicalizedParser -tLPP edu.stanford.nlp.parser.lexparser.ArabicTreebankParserParams -arabicFactored -train atb_no_plus_training.txt -saveToSerializedFile arabic_3TB_buckwalter.ser.gz -saveToTextFile arabic_3TB_buckwalter.txt

Without Diacritic

*java -Xmx65g edu.stanford.nlp.parser.lexparser.LexicalizedParser -tLPP edu.stanford.nlp.parser.lexparser.ArabicTreebankParserParams -arabicFactored -train atb_no_plus_no_* diacritics *_training.txt -saveToSerializedFile arabic_3TB_buckwalter_no_* diacritics.*ser.gz -saveToTextFile arabic_3TB_buckwalter_no_* diacritics.*txt*

The two generated models are *arabic_3TB_buckwalter.ser.gz,* and *arabic_3TB_buckwalter_no_* diacritics.*ser.gz*

- Testing

The tested sample file has 1935 tree. Using the previously generated models, Stanford parser is tested against diacritic and not diacritic testing files using the following commands:

- With Diacritic

java -Xmx65g edu.stanford.nlp.parser.lexparser.LexicalizedParser -tLPP edu.stanford.nlp.parser.lexparser.ArabicTreebankParserParams -arabicFactored -tokenized -writeOutputFiles -outputFilesExtension out -outputFormat "oneline" -loadFromSerializedFile arabic_3TB_buckwalter.ser.gz Arabic_sentences_ diacritic _testing.txt

- Without Diacritic

java -Xmx65g edu.stanford.nlp.parser.lexparser.LexicalizedParser -tLPP edu.stanford.nlp.parser.lexparser.ArabicTreebankParserParams -arabicFactored -tokenized -writeOutputFiles -outputFilesExtension out -outputFormat "oneline" -loadFromSerializedFile arabic_3TB_buckwalter_no_vowls.ser.gz Arabic_sentences_no_ diacritic _testing.txt

The two generated files are Arabic_sentences_diacritic_testing.txt and Arabic_sentences_no_diacritic_testing.txt will be compared against the gold standard files.

Results

To calculate the results, tags in parser output file and gold standard is converted from Morphological Arabic Tags (used in ATB) to original Penn Treebank tag set:

Example:

(S (S (CONJ wa-)(VP (PV+PVSUFF_SUBJ:3MS -$ab~a)(NP-SBJ (NOUN+CASE_INDEF_NOM HariyqN))(PP-LOC (PREP fiy)(NP (DET+NOUN+NSUFF_FEM_SG+CASE_DEF_GEN Aldab~Abapi)))))(S (ADVP-TMP (ADV vum~a))(VP (PV_PASS+PVSUFF_SUBJ:3MS nuqila)(NP-SBJ-2 (NP (DET+NOUN+CASE_DEF_NOM Aljunuwdu))(SBAR (WHNP-3 (REL_PRON Al~a*iyna))(S (VP (PV+PVSUFF_SUBJ:3MP kAnuwA)(NP-SBJ-3 (-NONE- *T*))(PP-LOC-PRD (PREP fiy)(NP (NOUN+CASE_DEF_GEN dAxili-)(POSS_PRON_3FS -hA)))))))(NP-OBJ-2 (-NONE- *))(PP-DIR (PREP <ilaY)(NP (DET+NOUN Almusota$ofaY)))(PP-PRP (PREP li-)(NP (DET+NOUN+CASE_DEF_GEN -AlEilAji)))))(PUNC .))

Converted to:

*(S (S (CC wa-)(VP (VBD -$ab~a)(NP (NN HariyqN))(PP (IN fiy)(NP (NN Aldab~Abapi)))))(S (RB (RB vum~a))(VP (VBN nuqila)(NP (NP (NN Aljunuwdu))(SBAR (WHNP (WP Al~a*iyna))(S (VP (VBD kAnuwA)(NP (NONE *T*))(PP (IN fiy)(NP (NN dAxili-)(PRP$ -hA)))))))(NP (NONE *))(PP (IN <ilaY)(NP (NN Almusota$ofaY)))(PP (IN li-)(NP (NN -AlEilAji)))))(PUNC .))*

Using the evaluation program, the results were:

	Precision	**Recall**	**F1-Score**	**Tags Accuracy**
With Diacritic	**%63.75**	**%62.33**	**%62.86**	**%89.52**
Without Diacritic	**%63.24**	**%61.88**	**%62.38**	**%89.24**

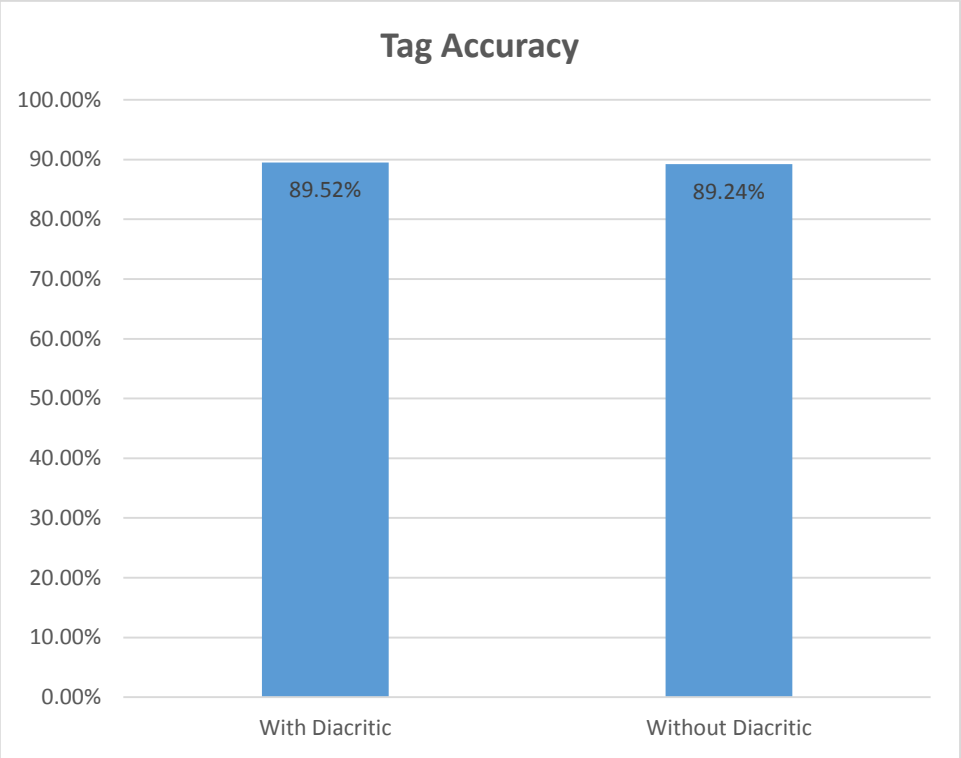


Figure (7) Stanford Parser evaluation with and without diacritic

5.2.3 Bikel Evaluation

- Training

  90% of ATB is used (20249 trees) in training to generate the Arabic Parsing Model. Two models are created one for the diacritic and non-diacritic Arabic texts using the following commands:

  - With Diacritic

  *java -Xms3000m -Xmx3000m -Dparser.settingsFile= arabic.properties danbikel.parser.Trainer -i atb*_diacritics

*_training.txt -o atb_*diacritics *_observed.txt -od atb_*diacritics *_derived*

- Without Diacritic

*java -Xms3000m -Xmx3000m -Dparser.settingsFile= arabic.properties danbikel.parser.Trainer -i atb_*no_diacritics*_training.txt -o atb_no_*diacritics*_observed.txt -od atb_no_*diacritics*_derived*

The two generated models are *atb_diacritics_derived*, and *atb_no_diacritics_derived*

- Testing

The tested sample file has 1935 tree. Using the previously generated models, Bikel parser is tested against diacritic and not diacritic testing files using the following commands:

- With Diacritic

java -Xms3000m -Xmx3000m -Dparser.settingsFile= arabic.properties danbikel.parser.Parser -is atb_diacritics_derived -sa Arabic_sentences_diacritic_testing.txt

- Without Diacritic

java -Xms3000m -Xmx3000m -Dparser.settingsFile= arabic.properties danbikel.parser.Parser -is atb_no_diacritics_derived -sa Arabic_sentences_no_diacritic_testing.txt

The two generated files are *Arabic_sentences_diacritic_testing.txt* and *Arabic_sentences_no_diacritic_testing.txt* will be compared against the gold standard files.

- Results

To calculate the results, tags in parser output file and gold standard is converted from Morphological Arabic Tags (used in ATB) to original Penn Treebank tag set:

- Using the evaluation program the results was:

	Precision	Recall	F1-Score	Tags Accuracy
With Diacritic	**%67.24**	**%63.21**	**%65.01**	**%87.15**
Without Diacritic	**%67.27**	**%63.51**	**%65.17**	**%87.63**

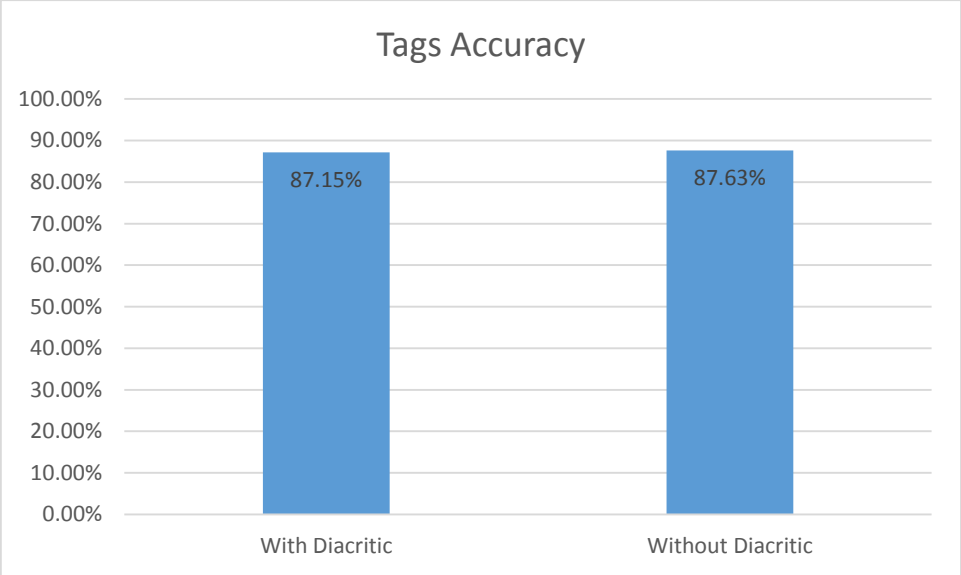


Figure (8) Bikel Parser evaluation with and without diacritic

5.3 Bikel Parser Modifications

5.3.1 Modifications

- Change in *training-metadata.lisp* which contains mapping between Morphological Arabic Tags (used in ATB) and original Penn Treebank tag set to enable the parser train trees that are tagged using Penn Treebank tags.

  Example:

  Old File:

  (NN NN)

  (ABBREV NN)

(DET+ABBREV NN)

(LATIN NN)

(DET+NOUN NN)

New File:

(ABBREV NN)

(DET+ABBREV NN)

(LATIN NN)

(DET+NOUN NN)

After the above changes the JAR file of Bikel Parser is regenerated. Training files (with/without diacritics) are changed to use Penn Treebank tags instead of Morphological Arabic Tags, noting that we didn't change the training file tags while we were comparing between Stanford and Bikel parses, we changed only the testing and the gold standard files.

Example:

Before Changes:

*(S (S (CONJ wa-)(VP (PRT (NEG_PART -lam))(IV3MS+IV+IVSUFF_MOOD:J ya+kun+o)(NP-SBJ (-NONE- *))(PP (PREP min)(NP (NP (DET+NOUN+CASE_DEF_GEN Al+sahol+i))(PP (PREP Ealay-)(NP (PRON_3MS -hi)))))(NP-PRD (NOUN+NSUFF_FEM_SG+CASE_DEF_NOM muwAjah+ap+u)(NP (NP (NOUN+NSUFF_FEM_PL+CASE_DEF_GEN kAmiyr+At+i)(NP (DET+NOUN+CASE_DEF_GEN Al+tilofizyuwn+i)))(CONJ wa-)(NP (NOUN+NSUFF_FEM_PL+CASE_DEF_GEN -Eadas+At+i)(NP (DET+NOUN+NSUFF_MASC_PL_GEN Al+muSaw~ir+iyna)))))))(CONJ wa-)(S (NP-TPC-1 (PRON_3MS -huwa))(VP (IV3MS+IV+IVSUFF_MOOD:I*

*ya+SoEad+u)(NP-SBJ-1 (-NONE- *T*))(NP-OBJ (DET+NOUN+CASE_DEF_ACC Al+bAS+a))))(PUNC .))*

After Changes:

*(S (S (CC wa-)(VP (PRT (RP -lam))(VBP yakuno)(NP (NONE *))(PP (IN min)(NP (NP (NN Alsaholi))(PP (IN Ealay-)(NP (PRP -hi)))))(NP (NN muwAjahapu)(NP (NP (NNS kAmiyrAti)(NP (NN Altilofizyuwni)))(CC wa-)(NP (NNS -EadasAti)(NP (NNS AlmuSaw~iriyna)))))))(CC wa-)(S (NP (PRP -huwa))(VP (VBP yaSoEadu)(NP (NONE *T*))(NP (NN AlbASa))))(PUNC .))*

- Training

  The new generated files are trained to generate (diacritics/non- diacritics) parsing models.

- Testing

  Using the previously generated models, Bikel parser is tested against diacritic and not diacritic testing files which contain 2267 line.

5.3.2 Evaluation

The gold standard data was utilized to compare the parser output. The results with the evaluation are as follows.

	Precision	**Recall**	**F1-Score**	**Tags Accuracy**
With Diacritic	**%73.11**	**%71.46**	**%72.13**	**%96.24**
Without Diacritic	**%73.10**	**%71.81**	**%72.29**	**%96.79**

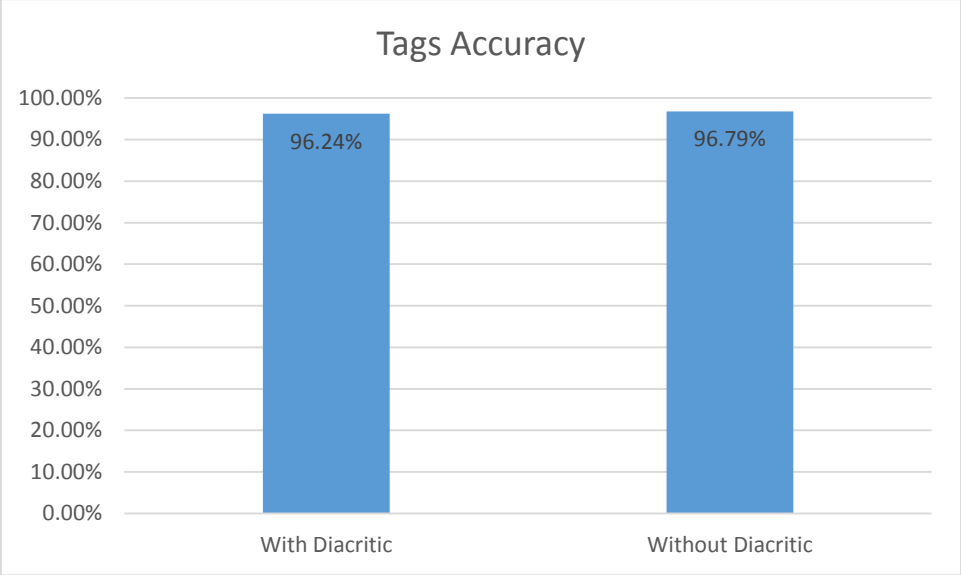


Figure (9) Modified Bikel Parser evaluation

5.4 Overall Evaluation

Comparison of Stanford, Bikel and Modified Bikel Figure(10)

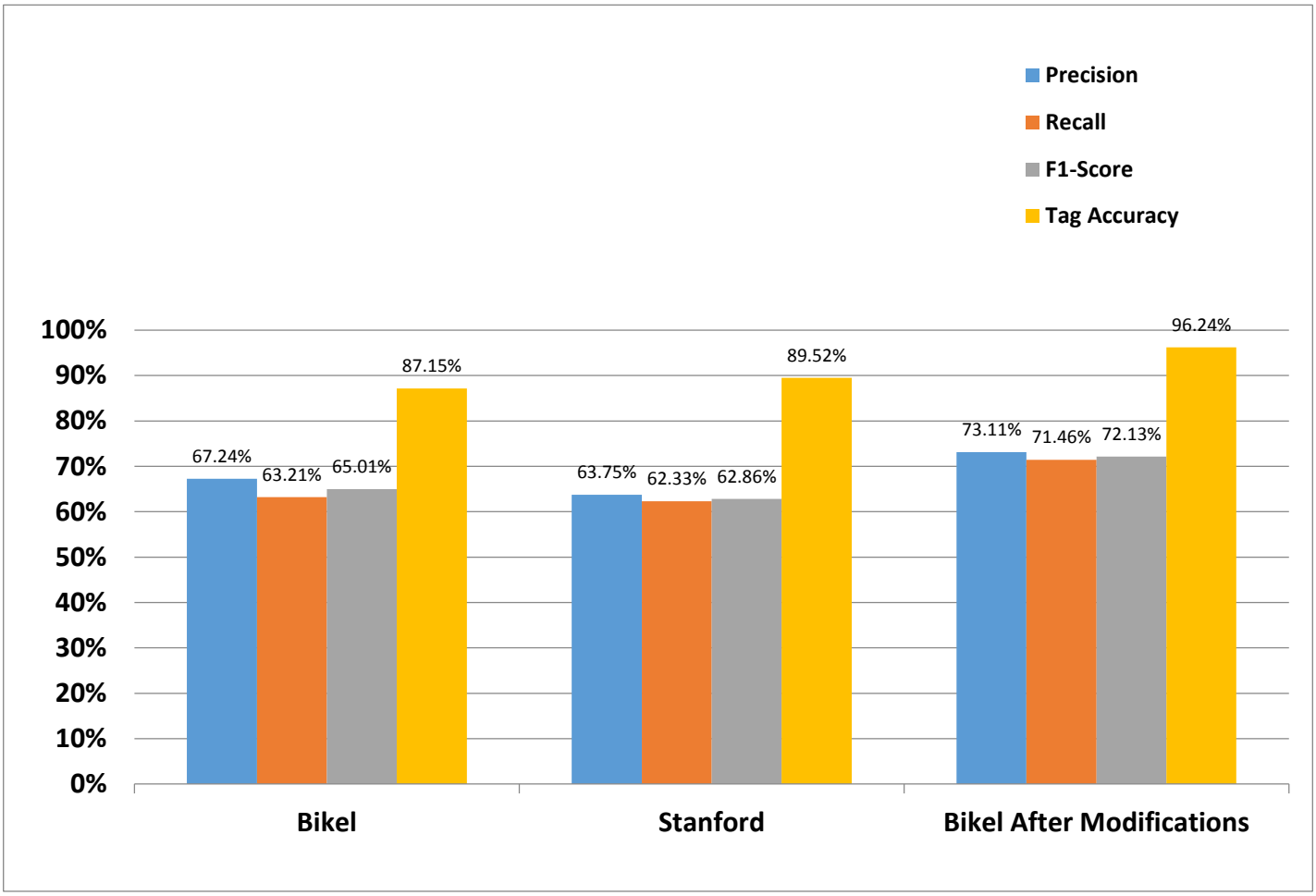


Figure (10) Overall Evaluation

Chapter Conclusion

In this experiment we created software that enables us to:

1- Convert Arabic morphological tags used in ATB to Penn tags.

2- Convert ATB annotation format to simple grammar rules.

3- Evaluate parsers output automatically.

We evaluated Bikel and Stanford parsers using diacritic and non-diacritic sample and we found that there is no remarkable difference between diacritic and non-diacritic samples, but we found that Bikel parser is performing slightly better than Stanford parser.

We changed the training file used for Bikel to be Penn tags instead of Arabic morphological tags, and change the Bikel source to accept this change, and after we

evaluate the modified Bikel parser we found remarkable improvement in the parser performance.

References:

1. Abdelbaki, H., Shaheen, M., & Badawy, O. (2011). ARQA high-performance arabic question answering system. In *Proceedings of Arabic Language Technology International Conference (ALTIC)*.
2. Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, 1996.
3. A. K. Joshi and B. Srinivas. Disambiguation of super parts of speech (or supertags): Almost parsing. In *Proceedings of the 1994 International Conference on Computational Linguistics*, 1994.
4. Al-Badrashiny, M., Eskander, R., Habash, N. and Rambow, O., 2014, June. Automatic Transliteration of Romanized Dialectal Arabic. In *CoNLL* (pp. 30-38).
5. Al-Atram, M.A. (1990). Effectiveness of natural language in indexing and retrieving arabic documents [in Arabic] (King Abdulaziz City for Science and Technology Project number AR-8-47). Riyadh, Saudi Arabia.
6. *AlGahtani, S., Black, W., McNaught, J., Arabic Part-Of-Speech Tagging using Transformation-Based Learning*, Proceedings of the Second International Conference on Arabic Language Resources and Tools(Year of Publication: 2009).
7. Al-Hamalawee, A. (2000). Shatha alarf fe fn alsarf. Dar Alkotob Alilmiyah [in Arabic]. Beirut, Lebanon.
8. Ali, B.B. and Jarray, F., 2013. Genetic approach for arabic part of speech tagging. *arXiv preprint arXiv:1307.3489*.
9. Al-Khuli, M. (1991). A dictionary of theoretical linguistics: English-Arabic with an Arabic-English glossary. Published by Library of Lebanon.
10. Al-Onaizan, Y. and Knight, K., 2002, July. Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting*

on Association for Computational Linguistics (pp. 400-408). Association for Computational Linguistics.

11. Al-Saeedi, M. (1999). Awdah Almasalik ila Alfiyat Ibn Malek. Published by Dar ihyaa al oloom [In Arabic]. Beruit, Saudi Arabia.
12. Al Shamsi, F. and A. Guessoum, 2006. A hidden markov Model-Based POS Tagger for Arabic.http://www.cavi.univparis3. fr/lexicometrica/jadt/jadt2006/PDF/004.pdf
13. Al-Sulaiti, L. and Atwell, E.S., 2006. The design of a corpus of contemporary Arabic. *International Journal of Corpus Linguistics*, *11*(2), pp.135-171.
14. Alqrainy, S., 2008. A morphological-syntactical analysis approach for Arabic textual tagging.
15. A. Monirabbassi, *Part of Speech Tagging of Levantine. Master Thesis, University of California, San Diego,2008*
16. Banko, M. and R.C. Moore, 2004. Part of speech tagging in context. Proceeding of the 20th international conference on Computational Linguistics, Aug. 23-27, Association for Computational Linguistics Morristown, New Jersey, USA., Article No. 556. http://portal.acm.org/citation.cfm?id=1220435
17. Benajiba, Y., Diab, M., & Rosso, P. (2009). Arabic named entity recognition: A feature-driven study. *Audio, Speech, and Language Processing, IEEE Transactions on*, *17*(5), 926-934.
18. Benajiba, Y., Diab, M. and Rosso, P., 2008, October. Arabic named entity recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 284-293). Association for Computational Linguistics.
19. Beesley, K.R. and Karttunen, L., 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.
20. Bird, S., Klein, E. and Loper, E., 2009. *Natural language processing with Python*. " O'Reilly Media, Inc."
21. Brill, E., *A simple rule-based part-of-speech tagger",* In Proce.ANLP-92, (Page: 152--155 Year of Publication: 1992) *[19]* Brill, E., *Some advances in transformation-based part of speech tagging*. In: Proce. Twelfth National Conference on Artificial Intelligence, (Page: 722-727 Year of Publication:1994).

22. B. Roark. Probabilistic top-down parsing and language modeling. C*omputational Linguistics*, 27(2):249–276, 2001.
23. Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0, Linguistic Data Consortium (LDC) catalog number LDC2002L49 and ISBN 1-58563-257-0.
24. C. Chelba and F. Jelinek. Recognition performance of a structured language model. In *Proc. Eur. Conf. Speech Communication and Technology (Eurospeech),* volume 4, pages 1567–1570, 1999.
25. Charniak, E. and Johnson,M. (2005) Coarse-to-fine n-best parsing andMaxEnt discriminative reranking. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 173–180. Ann Arbor, MI.
26. Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184.191.
27. Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL-EACL '97,* pages 16.23.
28. Collins, Michael John. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
29. Daniel M. Bikel. *On the Parameter Space of Lexicalized Statistical Parsing Models*. PhD thesis, Department of Computer and Information Sciences, University of Pennsylvania, 2004.
30. Daniel Bikel (2004a). On the Parameter Space of Lexicalized Statistical Parsing Models. PhD Thesis. University of Pennsylvania.
31. Daniel Bikel (2004b). Intricacies of Collins‟ Parsing Model. Computational Linguistics, 30:4 (479-511).
32. Diab, M., Hacioglu, K., & Jurafsky, D. (2004, May). Automatic tagging of Arabic text: From raw text to base phrase chunks. In Proceedings of HLT-NAACL 2004: Short papers (pp. 149- 152). Association for Computational Linguistics.
33. D. Sleator and D. Temperley. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University, Computer Science Dept, October,1991.

34. *Duh, K., Kirchhoff, K., Pos tagging of dialectal Arabic: A minimally supervised approach.* In Proceedings of the Association for Computational Linguistics (ACL). (Year Of Publication 2005)
35. E. Black, J. Lafferty, and S. Roukos. Development and evaluation of a broad coverage probabilistic grammar of english-language computer manuals. In *Proceedings of the 30th ACL*, pages 185–192, 1992.
36. E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997.
37. E. Charniak. A Maximum-Entropy-Inspired Parser. In *Proceedings of the First Annual Meeting of the North American Association for Computational Linguistics*, 2000.
38. El Hadj, Y., Al-Sughayeir, I., Al-Ansari, A., Arabic *Part- Of-Speech Tagging using the Sentence Structure,*Proceedings of the Second International Conference on Arabic Language Resources and Tools(Year of Publication:2009).
39. El-Dahdah, A. (1988). Dictionary of terms of declension and structure in universal Arabic grammar: Arabic-English, English-Arabic. Beirut, Lebanon: Librairie Du Liban.
40. El-Affendi, M.A. (1991). An algebraic algorithm for Arabic morphological analysis. The Arabian Journal for Science and Engineering, 16(4B), 605–611. Published by KFUPM, Dhahran, Saudi Arabia.
41. El-Affendi, M.A. (1992). Arabic dictionary compression using an invertible integer transformation and a bitmap representation. Journal of King Saud University, Engineering Sciences, 4(1), 105–125. Riyadh, Saudi Arabia.
42. El-Affendi, M.A. (1998). Performing Arabic morphological search on the Internet: A sliding window approximate matching (SWAM) algorithm and its performance. (Personal contact). King Saud University, Riyadh, Saudi Arabia.
43. El-Affendi, M.A. (1999). Building an Arabic distributed collaboration environment. The final report. Research project: AR-16-094, Supported by King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia.

44. El-Kareh, S. and Al-Ansary, S., 2000. An Arabic Interactive Multi-feature POS Tagger. In *Proceedings of the*.
45. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. Journal of Machine Learning Research, 9:1871–1874.
46. FARGHALY, A. 2005. A case for inter-Arabic Grammar. In Eligbali, A., Ed., *Investigating Arabic:Current Parameters in Analysis and Learning*. Brill, Boston.
47. Farghaly, A. and Shaalan, K., 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, *8*(4), p.14.
48. FERGUSON, C. 1959. Diglossia. *WORD, 15* 3, 325–340.
49. FERGUSON, C. 1996. Epilogue: Diglossia revisited. In *Contemporary Arabic Linguistics in Honor of El-Said Badawi*. The American University in Cairo.
50. F. Pereira and Y. Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th annual meeting of the association for computational linguistics*, pages 128–135, 1992.
51. Freeman, A., 2001. Brill's POS tagger and a morphology parser for Arabic. Proceeding of the ACL'01 Workshop on Arabic Language Processing.
52. G. Gazdar, E. H. Klein, G. K. Pullum, and I. A. Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.
53. Habash, N., Roth, R., Rambow, O., Eskander, R. and Tomeh, N., 2013, June. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Hlt-Naacl* (pp. 426-432).
54. Hull, D.A. (1996). Stemming algorithms: A case study for detailed evaluation. Journal of the American Society for Information Science, 47, 70–84.
55. Jaf, S.F. and Ramsay, A., 2013. Towards the Development of a Hybrid Parser for Natural Languages. In *OASIcs-OpenAccess Series in Informatics*(Vol. 35). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
56. Jan Hajič and Barbora Hladká (1998). Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In Proceedings of the

International Conference on Computational Linguistics (COLING) (483-490). Montreal.

57. Jan Hajič, Barbora Hladká and Petr Pajas (2001). The Prague Dependency Treebank: Annotation Structure and Support. In Proceedings of the IRCS Workshop on Linguistic Databases (105-114). University of Pennsylvania.
58. Jan Hajič, Jarmila Panevová, Zdenka Urešová, Alevtina Bémovám, et al. (2003). PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In Proceedings of the Workshop on Treebanks and Linguistic Theories (57-68). Växjö, Sweden.
59. Jan Hajič, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, et al. (2004). Prague Arabic Dependency Treebank: Development in Data and Tools. In Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools (110-117). Cairo, Egypt.
60. J. D. Lafferty, D. Sleator, and D. Temperley. Grammatical trigrams: A probabilistic model of link grammar. In *Proc. of AAAI Fall Symp. Probabilistic Approaches to Natural Language*, Cambridge, MA, Oct.1992. [116] D. Sleator and D. Temperley. Parsing english with a link grammar. Technical Report CMU-CS-91-196, Carnegie Mellon University, Computer Science Dept, October,1991.
61. J. Goodman. Efficient algorithms for parsing the dop model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 143–152, 1996.
62. J. K. Baker. Trainable grammars from speech recognition. In *Speech communication papers, 97th mtg. accoustic soc. of America (D. H. Klatt and J.J. Wolf, eds.*, pages 547–550, 1979.
63. J. Allen. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Menlo Park, CA, second edition, 1995.
64. J. M. Eisner. Bilexical grammars and a cubic-time probabilistic parser. In *Proceedings of the International Workshop on Parsing Technologies*, 1997.
65. J. M. Eisner. An empirical comparison of probability models for dependency grammar. Technical report, University of Pennsylvania, CIS Department, Philadelphia PA 19104-6389, 1996.

66. Johan Hall and Joakim Nivre (2008). A Dependency-driven Parser for German Dependency and Constituency Representations. In Proceedings of the ACL Workshop on Parsing German (PaGe08) (47-54). Ohio, USA.
67. Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryiğit, et al. (2007a). Single Malt or Blended? A Study in Multilingual Parser Optimization. In Proceedings of the Shared Task Session of EMNLP-CoNLL (933-939). Prague.
68. Johan Hall, Joakim Nivre and Jens Nilsson (2007b). A Hybrid Constituency-Dependency Parser for Swedish. In Proceedings of the Nordic Conference on Computational Linguistics (NODALIDA) (284-287). Tartu, Estonia.
69. Klein, D. and Manning, C.D., 2003, July. Accurate unlexicalized parsing. In*Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1* (pp. 423-430). Association for Computational Linguistics.
70. Khoja, S., 2001, June. APT: Arabic part-of-speech tagger. In *Proceedings of the Student Workshop at NAACL* (pp. 20-25).
71. Krovetz, R. (1993, July). Viewing morphology as an inference process. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 191–202). New York: ACM.
72. Maamouri, M. and C. Cieri, 2002. Resources for Arabic natural language processing at the LDC. Proceeding of the International Symposium on the Processing of Arabic, Tunisia, 2002, pp: 125-146.
73. Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR International Conference on Arabic Language Resources and Tools*,pages 102–109, 2004.
74. Maad Shatnawi and Boumediene Belkhouche (2012). Parse Trees of Arabic Sentences Using the Natural Language Toolkit. In Proceedings of the International Arab Conference
75. Mahmoud Shokrollahi-Far, Behrouz Minaei, Issa Barzegar, Hadi Hossein-Zadeh, et al. (2009). Bootstrapping Tagged Islamic Corpora. In Proceedings of the International Conference on Arabic Language Resources and Tools (48-53). Cairo, Egypt.

76. Mansour, S., Sima'an, K., Winter, Y., *Smoothing a lexicon-based pos tagger for Arabic and Hebrew*, In ACL07 Workshop on Computational Approaches to Semitic Languages, Prague, Czech. (Year of Publication: *2007)*
77. M. Marcus. Session 9 summary. In *Proceedings of the June 1990 DARPA Speech and Natural Language Workshop*, pages 249–250, 1990.
78. M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.
79. Maytham, Alabbas and Allan, Ramsay. Evaluation of combining data-driven dependency parsers for arabic. In *Proceedings of the 5th Language & Technology Conference Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 546– 550, 2011.
80. M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.
81. M. Chitrao and R. Grishman. Statistical parsing of messages. In *Proceedings of Speech and Natural Language Processing Workshop*, pages 263–266, 1990.
82. Metri, G., & George, H. (1990). Al Khaleel. A dictionary of Arabic syntax terms. Beirut: Library of Lebanon.
83. Michael Collins (1999). Head-Driven Statistical Models for Natural Language Parsing. PhD Thesis. University of Pennsylvania.
84. Michael Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29:589–637, 2003.
85. Mitchell Marcus (1980). A Theory of Syntactic Recognition for Natural Language. MIT Press.
86. Mitchell Marcus, Beatrice Santorini and Mary Marcinkiewicz (1993). Building a Large Annotated corpus of English: The Penn Treebank. Computational Linguistics, 19:2 (313-330).
87. M. J. Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 184–191, 1996.
88. M. M. Wood. *Categorial Grammars*. Rouledge, 1993.

89. Mohamed, E. and Kübler, S., 2010, May. Arabic Part of Speech Tagging. In*LREC*.

90. Mohamed Maamouri, Ann Bies and Seth Kulick (2008). Enhancing the Arabic Treebank: A Collaborative Effort toward New Annotation Guidelines. In Proceedings of the Language Resources and Evaluation Conference (LREC) (3192-3196). Marrakech, Morocco.
91. Mohamed Maamouri, Ann Bies, Timothy Buckwalter and Wigdan Mekki (2004). The Penn Arabic Treebank: Building a Large-scale Annotated Arabic Corpus. In Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools (102-109). Cairo, Egypt.
92. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E., 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, *13*(02), pp.95-135.
93. Nizar Habash and Owen Rambow (2005). Arabic Tokenization, Morphological Analysis and Part-of-Speech Tagging in One Fell Swoop. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)* (573-580). Michigan, USA.
94. Nizar Habash (2007a). Arabic Morphological Representations for Machine Translation. In Abdelhadi Soudi (Editor), *Arabic Computational Morphology: Knowledge-Based and Empirical Methods*. Springer. Dordrecht.
95. Nizar Habash, Ryan Gabbard, Owen Rambow, Seth Kulick, et al. (2007b). Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (1084-1092). Prague.
96. Nizar Habash, Reem Faraj and Ryan Roth (2009a). Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of the International Conference on Arabic Language Resources and Tools (MEDAR)* (125-132). Cairo, Egypt.
97. Nizar Habash, Owen Rambow and Ryan Roth (2009b). MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In *Proceedings of the International Conference on Arabic Language Resources and Tools (MEDAR)* (102-109). Cairo, Egypt.

98. Nizar Habash and Ryan Roth (2009c). CATiB: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP Conference Short Papers* (221-224). Suntec, Singapore.
99. Nizar Habash (2010). *Introduction to Arabic Natural Language Processing*. Morgan and Claypool.
100. Oudah, M., Shaalan, K. (2016a). Studying the impact of language-independent and language-specific features on hybrid Arabic Person name recognition, Language Resources & Evaluation, Springer. doi:10.1007/s10579-016-9376-1
101. Oudah, M., Shaalan, K. (2016b). NERA 2.0: Improving coverage and performance of rule-based named entity recognition for Arabic, Journal of Natural Language Engineering (JNLE), FirstView:1-32, Cambridge University Press, UK, May 2016. DOI: 10.1017/S1351324916000097
102. Paice, C.D. (1994). An evaluation method for stemming algorithms. In W.B. Croft & C.J. van Rijsbergen (Eds.), Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (pp. 42–50). London: Springer-Verlag.
103. Pasha, A., Al-Badrashiny, M., Diab, M.T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O. and Roth, R., 2014, May. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *LREC* (pp. 1094-1101).
104. Rambow, O., Chiang, D., Diab, M., Habash, N., Hwa, R., Lacey, V., Levy, R., Nicols, C., Shareef, S., Simaan, K., *Parsing Arabic Dialects.* Technical Report, The Johns- Hopkins University, (Year of Publication: 2009).
105. Ray, S., Shaalan, K. A Review and Future Perspectives of Arabic Question Answering Systems, IEEE Transactions on Knowledge and Data Engineering, IEEE, 2016, DOI: 10.1109/TKDE.2016.2607201.
106. R. Bod. Using an annotated corpus as a stochastic grammar. In *Proceedings of the Six Conference of the European ACL Conference*, pages 37–44, 1993.
107. R. Resnik. Probabilistic tree adjoining grammar as a framework for statistical natural language processing. In *Proceedings of the 1992 international conference on Computational Linguistics (COLING-1992)*, pages 418–424, 1992.

108. Roche, E. and Schabes, Y., 1997. *Finite-state language processing*. MIT press.
109. Saliba, B., & Al-Dannan, A. (1989, March). Automatic morphological analysis of Arabic: A study of content word analysis. In Proceedings of the 1st Kuwait Computer Conference, Kuwait.
110. Sawalha, M. and Atwell, E.S., 2010. Fine-grain morphological analyzer and part-of-speech tagger for Arabic text. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*(pp. 1258-1265). European Language Resources Association (ELRA).
111. Sawalha, M., Atwell, E. and Abushariah, M.A., 2013, February. SALMA: Standard Arabic Language Morphological Analysis. In *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on* (pp. 1-6). IEEE.
112. Y. Schabes, M. Roth, and R. Osborne. Parsing the Wall Street Journal with the inside-outside algorithm. In *Proceedings of the 1993 European ACL*, pages 341–347, 1993.
113. Seth Kulick, Ryan Gabbard and Mitchell Marcus (2006). Parsing the Arabic Treebank: Analysis and Improvements. In Proceedings of the Treebanks and Linguistic Theories Conference (31-42). Prague.
114. Shaalan, K. (2014). A Survey of Arabic Named Entity Recognition and Classification, Computational Linguistics, 40 (2): 469-510, MIT Press, USA.
115. Shaalan, Khaled, Magdy, Marwa, Fahmy, Aly. Analysis and Feedback of Erroneous Arabic Verbs, Journal of Natural Language Engineering (JNLE), 21(2):271-323, Cambridge University Press, UK, Sept. March 2015.
116. Slav Petrov (2009). Coarse-to-Fine Natural Language Processing. PhD Thesis. University of California, Berkeley.
117. Spencer, A. (1991). Morphological theory. Oxford: Basil Blackwell.
118. Spence Green and Christopher Manning (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis. In Proceedings of the International Conference on Computational Linguistics (COLING) (349-402). Beijing, China.
119. T. Briscoe and J. Carroll. Generalized LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25– 60, 1993.

120.T. L. Booth and R. A. Thompson. Applying probabilistic models to abstract language. *IEEE Transactions on Computers*, C-22(5):442–450, 1973.]

121.Thalouth, B., & Al-Dannan, A. (1987). A comprehensive Arabic morphological analyzer/generator. IBM Kuwait Scientific Center. Kuwait.

122.Tlili-Guiassa, Y., 2006. Hybrid method for tagging Arabic text. J. Comput. Sci., 2: 245-248. http://www.scipub.org/fulltext/jcs/jcs23245-248.pdf.

123.Waldvogel, B. (2008). Liblinear-java: http://liblinear.bwaldvogel.de.

124.Y. Schabes. Stochastic Lexicalized Tree Adjoining Grammars. In *Proceedingsof the International Conference on Computational Linguistics*, pages 75–80, 1992.

121.C. F. Hempelmann, V. Rus, A. C. Graesser, and D. S. McNamara. Evaluating state-of-the-art treebank-style parsers for coh-metrix and other learning technology environments. In Proceedings of the Second Workshop on Building Educational Applications Using Natural Language Processing and Computational Linguistics at the ACL conference, 2005.

122. Spence Green and Christopher D. Manning. 2010. Better arabic parsing:aselines, evaluations, and analysis. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 394–402, Beijing, China, August. Coling 2010 Organizing Committee.

123. Mohamed Maamouri, Seth Kulick and Ann Bies. 2008. Diacritic Annotation in the Arabic Treebank and its Impact on Parser Evaluation. In Proceedings of LREC 2008.

123. Seth Kulick, Ryan Gabbard, and Mitch Marcus. 2006. Parsing the arabic treebank: Analysis and improvements. In Proceedings of the Treebanks and Linguistic Theories Conference, pages 31–42, Prague, Czech Republic.

124. Comelles, E., Arranz, V. and Castellon, I. (2010). Constituency and Dependency Parsers Evaluation. SEPLN (ed.), Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural. Valencia:SEPLN, v. 45, p. 59-66. Valencia, Spain. ISSN: 1135-5948