Journal of Computer Science 7 (10): 1505-1514, 2011
ISSN 1549-3636
© 2011 Science Publications

# Rule Based Shallow Parser for Arabic Language

[1]Mona Ali Mohammed and [2]Nazlia Omar
[1]Department of Computer Science, Faculty of Information Science and Technology,
University Kebangsaan Malaysia 43600 Bangi, Selangor, Malaysia
[2]Department of Computer Science, Faculty of Science,
Omer Al Mukhtar University, Al Bayda, Libya

**Abstract: Problem statement:** One of language processing approaches that compute a basic analysis of sentence structure rather than attempting full syntactic analysis is shallow syntactic parsing. It is an analysis of a sentence which identifies the constituents (noun groups, verb groups, prepositional groups), but does not specify their internal structure, nor their role in the main sentence. The only technique used for Arabic shallow parser is Support Vector Machine (SVM) based approach. The problem faced by shallow parser developers is the boundary identification which is applied to ensure the generation of high accuracy system performance. **Approach:** The specific objective of the research was to identify the entire Noun Phrases (NPs), Verb Phrases (VPs) and Prepositional Phrases (PPs) boundaries in the Arabic language. This study discussed various idiosyncrasies of Arabic sentences to derive more accurate rules to detect start and the end boundaries of each clause in an Arabic sentence. New rules were proposed to the shallow parser features up to the generation of two levels from full parse-tree. We described an implementation and evaluate the rule-based shallow parser that handles chunking of Arabic sentences. This research was based on a critical analysis of the Arabic sentences architecture. It discussed various idiosyncrasies of Arabic sentences to derive more accurate rules to detect the start and the end boundaries of each clause in an Arabic sentence. **Results:** The system was tested manually on 70 Arabic sentences which composed of 1776 words, with the length of the sentences between 4-50 words. The result obtained was significantly better than state of the art Arabic published results, which achieved F-scores of 97%. **Conclusion:** The main achievement includes the development of Arabic shallow parser based on rule-based approaches. Chunking which constitutes the main contribution is achieved on two successive stages that include grouped sequences of adjacent words on the basis of linguistic properties.

**Key words:** Arabic shallow parsing, rule based approaches, text chunking, Arabic language processing, Arabic language phrases, Natural Language Processing (NLP), hand shallow, Part Of Speech (POS)

## INTRODUCTION

Natural language is a very important and ubiquitous part of human intelligence and society. Natural language processing (NLP) is the one of the various types of artificial intelligence sciences, which is overlapping in information and interferes significantly with the progress of linguistics with regard to the linguistic profile required for computers. Through the science of the software industry, we are able to analyze and simulate the understanding of natural language. NLP is the automated approach to analyze text that is based on a set of theories and a set of technologies together. In fact, NLP has recently received attention in terms of research and development. Syntactic analysis identifies certain patterns of words in a sentence as forming phrases of different types, such as noun phrases, verb phrases and adjectival phrases (Abney 1991; Ramshaw and Marcus 1995). Syntactic analysis categories: Full parsing and shallow parsing. In full parsing, a grammar and search strategy is used to assign a complete analysis for each sentence. On the other

**Corresponding Author:** Nazlia Omar, School of Computer Science, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia Tel: 60-3-89216733 Fax: 603-8921 6732

hand shallow for natural languages is often separated into two major parsing of sentence parts is analysis without building a complete typical parser tree (Pierce 2003). This study is to increase the performance of the Arabic shallow parser using the rule-based approach.

**Shallow parser task:** Shallow parsers represent the task of recovering only a partial amount of syntactic information to identify phrases from natural language sentences. Shallow parsing is the process of grouping consecutive words together to form phrases by a chunker, also called chunks (Patrick 2009). Chunking does not provide information on how the phrases attach to each other. The structures generally specified by shallow parsers include phrasal heads and their immediate and unambiguous dependents and these structures are usually non-recursive (Mokhtar *et al*., 2002). On the other hand a full parser tree defines completely by specifying the syntactic relationships between all constituents. A shallow parsing method can be more robust than a full parsing method in cases of low quality input or spoken language, because sometimes in the input there exists noise, mistakes and missing words, (Li and Roth, 2001). Full parsing is expensive, is not very robust, much slower and it gives more information than needed. On the other hand, partial parsing can be much faster, more robust and be sufficient for many natural language processing applications (Pierce 2003).

**Parts of speech tags of the Arabic language:** Part of speech (POS) tags are widely used NLP tools and applications development. The input of the shallow parser task takes the form of POS tags most of the time. These tags are different for different languages. Arabic linguistic is usually unclear and the parts of speech are difficult to define. For Arabic, several tag sets had been proposed. Classified words into three main classes. Verbs are sub classified into 3 subclasses, nouns into 46 subclasses and particles into 23 subclasses. Khoja (2001) described more detailed tagsets. Her tagset contains 177 tags, 57 verbs, 103 nouns, 9 particles, 7 residuals and 1 punctuation. In addition, there is the Arabic Treebank tagset, which is used in the Arabic Treebank. These POS tags are extensively used in this work. Table 1 shows the reduced Arabic Treebank tagset.

Dukes *et al*. (2010) propose an Arabic tagset based on traditional Arabic grammar which is to tag the Arabic Quranic corpus. Their tag set contains 35 tags, 3 nouns a verb and 31 particles.

**Methods of shallow parser:** Several researchers applied different techniques to deal with chunking in several languages. These techniques are rule-based, corpus-based and the hybrid approach for shallow parser. After well over a decade from the control of the statistical paradigm in NLP applications, we seem to be witnessing a renewed interest in rulebased approaches to solve common problems such as partial syntactic parsing (Grover and Tobin 2006). The advantages of rule-based chunkers are that rules can be hand-written and easily comprehended. On the other hand, the disadvantages are that the rules are language and corpus specific and it takes a large amount of work and needs lots of linguistic knowledge (Shaalan 2010). In the literature, we can find several learning methods which have been applied to perform shallow parsing: memory-based learning, transformation-based learning, hidden markov models, maximum entropy, support vector machines. The greatest advantage of using machine learning techniques is ease of classification by styles, domains. However a major disadvantage is heavy reliance on the quality and size of training corpora and also, for the best result the training and testing data must be under the same domain. Machine learning approach usually gives good results when the training set and the testing data are similar (Shaalan 2010). However, the accuracy of the shallow parser based on machine learning techniques is affected by the following parameters: the language, the domain, the training data and the size of the training data.

**Related work:** The first proposed text shallow parser by Abney (1991) proposed a new approach to parsing by starting with identifying related chunks of words. He divided the parsing task into chunkers and attachments to them. He mentioned that when we read, we read chunk by chunk. This work introduced this natural phenomenon into the machine world. He deduced that the task of the chunker was to convert sentences into non-overlapping phrases and the attacher was to combine these chunks in such a way that it would be possible to get complete parses of the sentences. After Abney, much of the work that has been done on chunking was applied to different techniques for implementation of chunking tasks in different languages.

Compared to what has been done in English and other languages, there is only one approach that has been investigated for Arabic shallow parsing Diab *et al*. (2004; 2007; 2009). Diab *et al*. (2004) performed tokenization, POS tagging and used an SVM-based approach for Arabic text chunking.

Table 1: The reduced arabic treebank tagset

| Pos tag | Label | Pos tag | Label |
|---|---|---|---|
| Conjunction | CC | Possessive pronoun | POSS_PRON |
| Number | CD | Imperfective verb | VBP |
| Adverb | ADV | Non inflected verb | NIV |
| Particle | PART | Relative pronoun | REL_PRON |
| Imperative verb | IV | Interjection | INTERJ |
| Foreign word | FOREIGN | Interrogative particle | INTER_PART |
| Perfect verb | PV | Interrogative adverb | INTER_ADV |
| Passive verb | PSSV | Demonstrative pronoun | DEM_ PRON |
| Preposition | PREP | Punctuation | PUNC |
| Adjective | ADJ | Proper noun | NOUN_PROP |
| Singular noun | SN | Personal pronoun | PRON |
| Plural noun | SPN | | |

They adopted an existing SVM tool Allwein *et al*. (2000). They trained this tool using the Arabic Treebank. Their chunks of data were derived from the LDC Arabic Tree Bank using the same program that extracted the chunks for the shared task. They used the same features and achieved over-all chunking performance of 92.06% precision, 92.09% recall and 92.08 F-measure. There had been no previous studies done under this research before. Also Diab (2007a) used the same SVM tool for Arabic tokenization, POS and chunking. They reported a chunking performance of 93.04%. Diab *et al*. (2007b) and Diab (2009) also used the same SVM tool and also trained using the Arabic Treebank. They reported chunking performances of 96.06% and 96.33%, respectively. However, there are a lot of approaches which have been successfully applied to many languages. It would be interesting to adopt and test them on Arabic languages which have a radically different morphology and syntax.

**Phrases in Arabic language:** Arabic words classified as noun "إسم", verb "فعل", or particles "حَرف", intended for items which are neither noun nor verb. The clear difference between the three parts is the declension "الأعراب "or syntactic parsing. The major three phrases for the Arabic language.

**Noun phrase:** A noun phrase is one which starts with a noun or a pronoun. It represents the entity of person, place, animal, etc.) about which the phrase is talking. The nominal sentence is composed of "starting" "المبتدأ", which is followed by "information" "الخبر". Information is the part of the phrase to complete the information about starting.

**Verb phrase:** A verbal phrase is one which starts with a verb in any of the three forms (present verb, past verb and order verb). The verbal phrase is considered stronger than a noun phrase composition-wise. The verbal sentence is composed of verb "الفعل" which is followed by "subject" "الفاعل". This means that the verb did not need more than the subject to fulfill the meaning. In this situation the verb call "الفعل اللازم", is an "intransitive verb". Another case is that of a verbal sentence composed of a verb "الفعل" followed by subject "الفاعل" followed by the object "المفعول به", which is who or what received the action of that verb. In this situation the verb call "الفعل المتعدى", is a "transitive verb".

**Prepositional phrases: A prepositional phrases" in:** Arabic are used just like in English. It is in the sequence of a preposition followed by a word or phrase. There are 20 particles "حرف الجر" in the Arabic language, some prepositions that are one-letter, two-letter and three-letter word groups.

## MATERIALS AND METHODS

**System architecture:** The flow chart of our system is shown in Fig. 1. The input of this system are sequences of linguistic objects (ranging from raw text to POS-tagged tokens) from which it produces a sequence of constituent structures such as NP, PP, VP.

However, the shallow parser system takes as input a sequence of disambiguated words, where each word has a single lexical reading. Therefore, our system is able to handle an input sequence generated by any Arabic morphological analyzer or Arabic POS tagger. Also, its input may consist of text which has been processed by a POS tagger or manually by POS annotated corpora.

**Pre-processing modules:** The system includes three optional pre-processing modules that can be used before the shallow parser. The utilization of these modules depends on the nature of the input. Actually these modules are used in the case when the input is raw text. However these modules will not be used when the input is an annotated corpus. The modules are normalization, tokenizer and POS disambiguation.

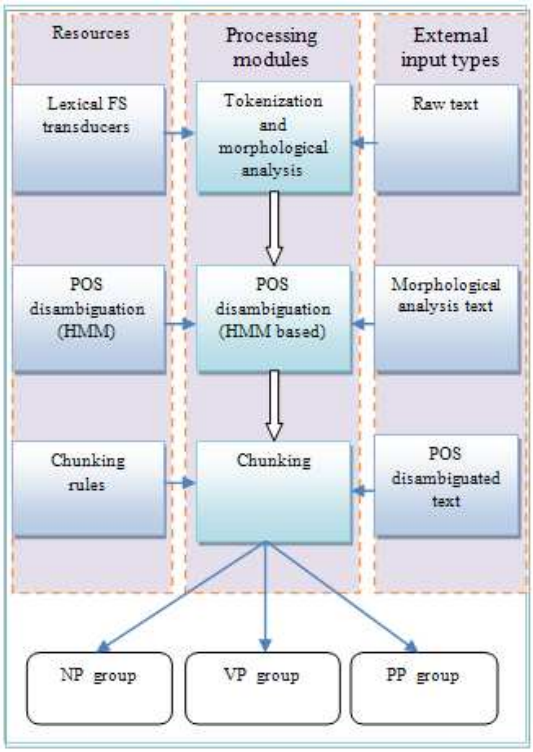


Fig. 1: The global architecture of the system

**The normalization module:** Normalization is a preliminary step to Arabic tokenization to ensure that the text is steady and predictable. It is a basic task that researchers in Arabic NLP always apply with a common goal in mind: reducing noise and sparsely in the data (El Kholy and Habash 2010). Normalization is a challenging process for researchers of Arabic NLP as a result of irregularity of using diacritic marks and certain letters in Arabic writing (Farghaly and Shaalan 2009).

**The Tokenizer module:** Arabic is a rich language with reference to inflection and derivation. The lexical items are created by attaching affixes to roots which require a very large corpus for good coverage of general Arabic. In Arabic language, one word may correspond to a single entity or many entities. A token is the minimal syntactic unit. The tokenization module is responsible for identifying a word, a part of a word (or a clitic), a multiword expression, or a punctuation mark (Attia 2007). Actually Arabic is a clitic language (Attia 2008) i.e. an Arabic token may consist of several lexical items which have their own meaning and POS. For example; according to Attia (2007) an Arabic verb can comprise up to four clitics: conjunction, tense, stems with affixes and object pronouns as shown in the following example.

Arabic verb: وَسَيَكلمكُم
Structure: و-س-يكلم-كم
POS:conj+fut+iv3ms+verb.imperfect+ivsuff.do:2mp
English translation: and + will + he/it + tell + you

**POS disambiguation:** POS disambiguation is the process of identifying the right grammatical category (POS) of each word in a text (corpus) based on both its definition, as well as its context. The POS disambiguation module is basically used when the input is in the form of a sequence of ambiguously tagged words or a raw text. This module consists of a stochastic POS tagger (based on HMMs).

**Corpora and domain:** The domain and genre can have a main rule on grammar, word senses and other linguistic properties. We utilize the Arabic Statistical POS Tagger (ASPOST) as reported Albared *et al.* (2010; 2011). ASPOST is a statistical part of speech tagger, trainable on different Arabic corpora. The system incorporates several methods of smoothing and of handling unknown words. ASPOST is optimized for Arabic language. The tagger is an implementation of the Viterbi algorithm for second-order Markov models. The transition probabilities are smoothed using linear interpolation of unigram, bigram and trigram maximum likelihood estimates in order to estimate the trigram transition probability: Unknown words are handled by using the linear interpolation of the word suffix probability and the prefix word probability. For the current work, a POS annotated corpus of 350 Arabic sentences which is composed of 9339 words extracted from different Arabic domains (economics, medical, sciences and etc) is utilized. Data on 280 Arabic sentences comprising 7563 words is prepared for analysis during the rule deriving and the remaining 70 Arabic sentences which comprised 1776 words are kept for testing the model.

**Implementation of rules:** A rule-based constituent for the shallow parser is used when the input is a sequence of lexical trees with no constituent structure. The input data is prepared in a specific format and each line contains only a POS tag matching with the word in the sentence. The rule formalism has been designed specifically for group sequences of categories into structures(chunks) to facilitate the dependency analysis. These rules are structured in layers that are applied on to the input sequences of sequential categories and they deal with syntactic structure and typical Arabic linguistic grammars to recognize several major categories of words in Arabic language.

Table 2: The single regular expression

| Sample | Meaning |
|---|---|
| **{ }** | Indicate the single chunk |
| **( )** | indicate the scope of the operators |
| < > | Determine part-of-speech tags |
| **?** | Zero or one of the previous item |
| * | Zero or more of previous item |
| \| | match a single item with others |

This rules are incrementally built and applied using the developing corpus. However the design contains 150 rules. The shallow parser then checks whether the first word of the input can belong to the category. The first 110 rules are implemented in the first stage to generate the first level from a shallow parser. The second rule set which contains 40 rules is run as a post-processing element in the second stage to generate the second level from a shallow parser. Initially, the hand crafted rules for the first level are derived, based on the experience through manual tagging for NP, VP and PP chunking.

Accordingly, a simple grammar chunk with a single regular expression rule defined is demonstrated in Table 2. The rules will describe sequences of tagged words to identify the three types of chunks which are covered by our rules.

**First phase:** The first aspect to be considered are the rules that aim at identifying NP, VP, PP chunk boundaries.

- Identification of Boundaries for Noun Phrases (NP): The first types of chunks are NP-chunks that, in our case, the essential point of interest are identifying noun phrase chunking. A NP is a group of words that work together, beginning with a noun or pronoun and optionally accompanied by a set of modifiers. Due to the typical Arabic grammatical structure for NP, we have the rules to build a grammatically correct noun phrase. The following are 7 general rules from which 70 rules for building a grammatically correct NP are derived as the following:

Rule1 NP:{ (<SN>│ <SPN >)* < POSS_PRON >? <ADJ >*}

**This rule has some exceptions:** In a case when applying the deriving rule (NP: { < SN> *}) to connect sequence of (SN), only one of them should be definite ( start with "ال"). Therefore, if the phrase contains more than one definite (SN), this rule cannot be applied and other rules should be applied depending on the sentence structure which will generate an understandable phrase. For example in the sentence below another rule will be applied which connects the first (SN) with the previous word (PREP) to identify (PP).

**In a case when applying the deriving rule (NP:** {< SPN> * < SN> *}) to connect a sequence of (SPN) followed by one or more (SN), the phrase should be contains indefinite Plural Nouns (not starting with "ال" ) followed by definite Singular Noun (start with "ال"). Therefore, if the phrase contains a definite (start with "ال") Plural Noun (SPN) followed by an indefinite ( not starting with "ال" ) (SN), then each one belongs to different phrase and this rule fails to be applied.

Rule2 NP: { <SN> < POSS_PRON >? <ADJ>* <NOUN_PROP >*}

Rule3 NP : { (<SPN>│<SN>)* <ADJ >* <SN>? <CD>?< NOUN_PROP >? <ADJ >?}

Rule4 NP: { < SN >* <DEM_PRON> (< SN >│< SPN >)? <ADJ>*}

Rule5 NP: {<ADJ> (<SPN>│<SN> ) * (<POSS_ PRON>│<ADJ >*)? }

Rule6 NP: {<SN> < POSS_PRON>? <CD>? (<SN>│<SPN>│<NOUN_PROP>)? <ADJ>*}

Rule7 NP: { < X : POS(X) $\in$ NP components> (<CC> < Y: POS(Y)= POS(X) >)* }

- **Identification of boundaries for Verb phrases (VP):** The second types of chunks are VP-chunks that, in our case, cover the compound verbal tenses and moods. A VP is a group of words that work together whose beginning commences with a noun, to express a unified meaning to provide information about the subject of the sentence. As mentioned above, due to the typical Arabic grammar for verb phrases, we have the rules to build a grammatically correct VP. The following are 2 general rules from which 30 rules for building a grammatically correct VP are derived.

Rule1 VP:{ (X: POS (X) $\in$ {<PAS>, <PRV > , <IV >}) <PPRON> }

Rule2 VP :{ (W: POS (X) $\in$ {<PAS>, < PASSV >,<PRV > , <IV >})( Y : POS(Y) $\in$ NP components> and Y is the last word)}

**In previous rule if Y is not last word we have two cases:** If the next word (after Y) is one of the following ( (CC), (REL_PRON), (ADV), (PAS), (PRV ), (PASSV ) or (IV) ) then the same rule is applied. We should note that, if the next word (after Y) (CC) it should not be (و, أو) (and, or) which connects two or more NP components of the same type. So, if CC is (و, أو) (and, or) or "أو" we don't apply this rule and the NP rules should be applied first.

2. If the next word (after Y) is not one of the following ((CC), (REL_PRON), (ADV), (PAS), (PRV ), (PASSV ) or (IV), then the NP rules should be applied first. If the NP rules fails then the same rule is applied.

. I**dentification of boundaries for preposition phrases:** The third type of chunks are PP and they could be defined as a combination of a preposition and a word or phrase, in our case. As mentioned above, due to the typical Arabic grammar requirement for PP, we have the rules to build a grammatically correct preposition phrase. The following are 2 general rules from which 10 rules for building a grammatically correct PP are derived:

Rule1 PP :{ <PREP > <PPRON>}

Rule2 PP: {<PREP> <Y: POS (Y) is one of NP components and Y is the last word >}

In the previous rule if Y is not last word we have two cases:

**If the next word (after Y) is one of the following:** ((CC), (REL_PRON), (ADV), (PAS), (PRV), (PASSV) or Imperative Verb (IV) ) then the same rule is applied. We should note that, if the next word (after Y) is (CC) it should not be (و, أو) (and, or) which connects two or more NP components of the same type. So, if CC is (و, أو) (and, or) or "أو" we don't apply this rule and the NP rules should be applied first.

**If the next word (after Y) is not one of the following:** ( (CC), (REL_PRON), (ADV), (PAS), (PRV ), (PASSV ) or (IV) ), then the NP rules should be applied first. If the NP rules fails then the same rule is applied.

**Second Phase:** Having developed the first level based on the above NP, VP, PP rules and some other linguistic grammatical requisites, the next step is to the second level. The following are 5 general rules generating for second level, to build grammatically correct phrases boundaries:

Rule1 NP <NP> <CC>? <NP>

In the previous rule if the (NP) which comes after CC is not followed by (PP), then we do not apply this rule. The next rule, which will connect (NP) with (PP ) should be applied first:

Rule2 NP <NP> <PP>

If the noun phrase (NP) followed by (PP), followed by noun phrase (NP) and (PP) contains only (PREP), the rule will be identified as {NP} {PP(<PREP> <NP>)}:

Rule3 NP <NP> <VP>

This rule cannot apply if the next phrase is NP, because in this case it should be a (VP) to connect with the next phrase (NP) to identify the (VP):

Rule4 VP <VP> <CC>? <VP>

Rule5 VP <VP> <NP>

In this rule should be put into consideration the requirement that the next two words must be scanned as following cases:

- If the (VP) is followed by (NP), followed by (PP) and then in turn followed by (PP) , another rule will be applied which connects the {NP PP}as a (NP)
- If the (VP) is followed by (NP), followed by a (CC) (أو,و) and then by a (NP), it will chunk as (VP) followed by two noun phrases connected together, because connecting the phrases by (أو,و) have a priority in the second level

Rule6 VP <VP > <PP> in this situation, it should be put into consideration the fact that the next two words must be scanned as following cases:

- If the (VP) is followed by (PP) contains (PREP and PPRON), followed in turn by (NP) and then (NP). In this case {VP ,PP, NP} will connected as a (VP
- If the (VP) is followed by (PP) and then followed by (PP) and the last (PP) contains only (PREP) followed by (NP), the rule will be identified as {VP} {PP} {PP(<PREP> <NP>)}:

Rule7 PP <PP> <NP>

In this situation, it should be put into consideration the fact that the next two words must be scanned as following cases:

- 1.If the next phrase is PP, because in this case it should be NP connected with the next phrase (PP) to identify NP
- 2. If the (VP) is followed by (NP), followed by a connector (CC) (أو,و) and then by a (NP), it will chunk as (VP) followed by two (NP) connected together, because connecting the phrases by (أو,و) have a priority in the second level

## RESULTS

The system is evaluated by conducting a series of experiments which depend on the length of the full sentence as well as each phrase separately. The generic rule set was applied to all experiments to identify (NP, VP and PP) and then compare the phrase output of the system with a human chunking standard set of the input text. However, the careful choosing of the Part Of Speech (POS) tagset has a directly impact on higher level syntactic processing (Diab 2007b). The accuracy of the chunker heavily depends in turn on the accuracies of the POS tagging (Siddiq 2009).

Turning now to the present shallow parser which has been tested on the Arabic Statistical POS Tagger (ASPOST) (Albared *et al*., 2010; 2011) which is trainable on different Arabic corpora. Evaluation of the system as such will not reflect the accuracy of the shallow parser algorithm. So, to estimate the accuracy of the shallow parser algorithm rather than the system as a whole, any errors found in APOST have been corrected manually to the level of a Gold Standard Corpus (GSC). 70 sentences of various lengths and which in total consists of 1776 words, were used for each experiment. The shortest sentence is of four words while the longest one is of 50 words. For different ranges of one sentence length to the another, the range is divided to five periods, each period including 14 sentences available for evaluation.

Generally, the test suite included various possible word orders (VSO, SVO and VOS), copula-less constructions, transitive and intransitive verb constructions, sentential and nominal modifications, questions, negations, demonstrative and relative clauses, complementary phrases, compounding and sentences with multiword expressions.

**Experimental results:** Three experiments were preformed on the same sentence. These experiments are divided into two phases. Firstly, the experiments are conducted for the first level of shallow parser in the first phase and then the experiments proceeded with the second level of shallow parser, in the second phase. The results obtained after executing all the experiments are as follows:

**First phase of experiments:** The first phase of experimentation is basically implementation of the first level of the Arabic shallow parser. The experiments for the first level will be discussed one by one.

**Experiment1: Evaluation of f-scores as being inversely proportional to the length of the sentence:** The first type of experiment is conducted as a detailed manual analysis for the first level F-scores with the length of sentence rate to see how the length of sentence has an effect on the first level accuracy. Table 3 shows the first level F-scores against the length of sentence to extract the relationship between the first level F-scores and the sentence length.

From the data in Table 3, we can draw a correlation graph to present F-scores against the length of sentence for each phrase, as shown in Fig. 2.

To explore the relationship between F-scores of the shallow parser and the length of sentence, we use the statistical measure of correlation. Correlation measures the degree to which two variables are related together. There is a negative correlation of -0.79 between the number of words per sentence and the F-scores of the shallow parser, which indicates that the length of sentences and F-scores go in opposite directions. Although correlation does not necessarily imply causation, it gives an idea of how the length of the sentence and the F-scores can be related.

**Experiment 2: Evaluation each phrase being inverse to the length of sentence:** The second type of experiments is conducted based on a detailed manual analysis for F-scores for each phrase (NP, VP, PP) with the length of sentence rate to see how the length of sentence has an effect on the each phrase F-scores of the first level of shallow parser. Table 4 shows F-scores for each phrase against the length of sentence to extract the relationship between F-scores for each phrase and the sentence length.

From the data in Table 4, we can draw a graph to present F-scores against the length of sentence for each phrase, as shown in Fig. 3.

The graph clearly presents and compares a first level shallow parser performance analysis against the length of sentence for each phrase. It is not surprising that the highest percentage is in the sentences with 0-10 words for all phrases.

Table 3: Evaluation F-scores inverse the length of sentence in the first level

| Length of sentence | F-scores (%) |
|---|---|
| 0-10 | 100% |
| 10-20 | 100% |
| 20-30 | 97.19% |
| 30-40 | 97.69% |
| 40-50 | 97.68% |

Table 4: F-scores for each phrase and the sentence length in the first level

| | F-scores (%) | | |
|---|---|---|---|
| Length of sentence Phrase | Noun phrase | Verb phrase | Preposition |
| 0-10 | 100% | 100% | 100% |
| 10-20 | 100% | 100% | 100% |
| 20-30 | 97.5% | 96.77% | 97.2% |
| 30-40 | 98% | 97.14% | 97.77% |
| 40-50 | 97.77% | 97.5% | 97.77% |

Table 5: The overall F-scores for the first level

| | Overall F-scores |
|---|---|
| NP Chunks | 98.35(%) |
| VP Chunks | 97.94(%) |
| PP Chunks | 98.21(%) |
| Average | 98.18(%) |

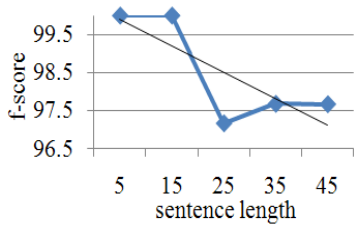


Fig. 2: Length of sentence effect on shallow parsing in the first level

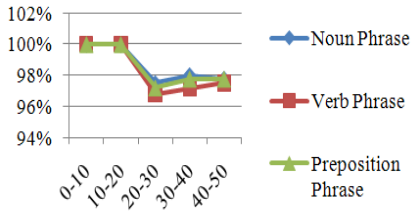


Fig. 3: F-scores against the length of sentence for each phrase in the first level

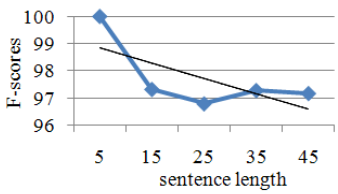


Fig. 4: Length of sentence effect on shallow parsing in the second level

There was no change in the percentage supplied by the sentences with 10-20 words for all phrases which remained at the highest F-scores. This may due to the fact that the sentences with lengths of less than 20 words are more fortunately of less complexity. However, in the sentences with lengths of 20-30 words F-scores showed a decrease for all phrases especially amongst Verb Phrases. Moreover F-scores showed an increase for all phrases in the sentences with lengths of 30-40 words and then there was a slight decrease in the sentences with lengths of 40-50 words amongst noun and verb phrases. In contrast there was no change preposition phrases which remained at 97.77%.

**Experiment 3: (Overall F-scores for The First Level):** The third type of experiments summarized the first level performance of the Arabic shallow parser. The final results of the systems show that the system set far outperforms for the first level, as shown in Table 5.

**Second phase of experiments:** The second phase of experimentation is basically implementation of the second level from the shallow parser. The experiments for the second level will be discussed one by one.

**Experiment1: Evaluation f-scores being inversely proportional to the length of sentence:** The first type of experiments is conducted involving a detailed manual analysis for the second level F-scores against the length of sentence rate to see how the length of sentence has an effect on F-scores. Table 6 shows the F-scores against the length of sentence to extract the relationship between F-scores and the sentence lengths.

From the data in Table 6, we can draw a correlation graph to present F-scores against the length of sentence, as shown in Fig. 4.

We study the relationship between F-scores and the lengths of sentence using the statistical measure of correlation. As expected, there is a negative correlation of -0.69 between the number of words per sentence and the F-scores of the shallow parser, which indicates that the length of sentences and F-scores go in opposite directions.

**Experiment 2: Evaluation of Each Phrase Being Inversely to The Length of Sentence**: The second type of experiments is conducted based on a detailed manual analysis for F-scores for each phrase (NP, VP, PP) against the length of sentence rate to see how the length of sentence has an effect on the each phrase F-scores of the second level of shallow parser. Table 7 shows F-scores for each phrase against the length of sentence to extract the relationship between F-scores for each phrase and the sentence length.

Table 6: Evaluation F-scores inverse the length of sentence in the second level

| Length of sentence | F-scores (%) |
|---|---|
| 0-10 | 100 |
| 10-20 | 97.33 |
| 20-30 | 96.8 |
| 30-40 | 97.29 |
| 40-50 | 96.18 |

Table7: F-scores for each phrase and the sentence length in the second level

| Length of sentence | F-scores (%) | | |
|---|---|---|---|
| | Noun phrase | Verb phrase | Preposition phrase |
| 0-10 | 100% | 100% | 100% |
| 10-20 | 95.45% | 100% | 96.66% |
| 20-30 | 96.29% | 96.77% | 97.22% |
| 30-40 | 97.43% | 96.87% | 97.5% |
| 40-50 | 93.87% | 97.5% | 97.61% |

Table 8: The overall f-scores for the second level

| | Overall F-scores |
|---|---|
| NP Chunks | 95.94% |
| VP Chunks | 97.82% |
| PP Chunks | 97.5% |
| Average | 97.08% |

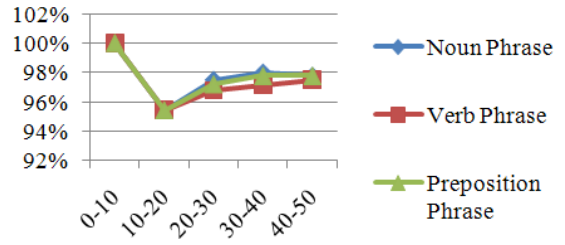


Fig. 5: F-scores against the length of sentence for each phrase in the second level

From the data in Table 7 we can draw a graph for F-scores against the length of sentence for each phrase, as shown in Fig. 5.

The graph clearly presents and compares a second level shallow parser performance analysis against the length of sentence for each phrase. It is not surprising that the highest percentage occurs in the sentences with 0-10 words for all phrases. Then, in the sentences with lengths of 10-20 words F-scores showed a decrease for noun and preposition phrases. Over the same length, verb phrases remained at the highest F-scores. However, in the sentences with lengths of 20-30 words F-scores showed an increase for Noun and preposition phrases with relatively less F-scores for Verb Phrase. Moreover the F-score showed an increase for all phrases in the sentences with lengths of 30-40 words especially for Noun Phrases. In those sentences with lengths of 40 to 50 words, there was a slight an increase in F-scores of verb and preposition phrases, with a drop in noun phrase F-scores.

**Experiment 3 (Overall F-scores for The Second Level):** The third type of experiments summarized the second level performance of Arabic shallow parser. The final results of the systems show that the system set far outperforms for the second level, as shown in Table 8.

## DISCUSSION

After comparison of all the experimental results using the same test corpus, the most important point to be made is that the chunking of unlabelled dependency relations is reasonably straightforward. The greater variance in sentence structure and grammatical functions in Arabic which leads to disambiguation of the boundary phrases of chunking, has several negative effects in the system. As has been noted, length of sentences has no effect on overall F-scores. Addition to that, we can see that, there is decrease in F-scores for all phrases in the second level. The major decrease was in noun phrases on the other hand, while the verb phrase F-scores showed a negligible decrease. The study confirmed that, the system performs with F-scores that are 97% better than state of the art published results on Arabic shallow parser

## CONCLUSION

This study describes the development of an Arabic shallow parser based on rule-based approach. The chunking which constitutes the main contribution are achieved on two successive stages that include grouped sequences of adjacent words on the basis of linguistic properties to identify each of NP, VP and PP. Based on the fact that the aim of the research is to generate results at two levels, the final results adopted were based on the second level results. Overall, the experiments has shown that the satisfactory results had been achieved. The system generates the first two levels from the Arabic full parser tree and the issues arise of extending this work to enable the building up of a full parser tree. In addition, other methods should be applied which have been successfully utilized for many languages for shallow parsers. Lastly, we would wish to see it integrated with statistical chunking to be implemented for better accuracy.

## REFERENCES

Abney, S., 1991. Principle-based parsing: computation and psycholinguistics.1st Edn., Springer, Dordrecht, ISBN: 0792311736, pp: 408.

Albared, M., N. Omar, M. Ab Aziz and M. Ahmad Nazri, 2010. Automatic part of speech tagging for arabic: An experiment using bigram hidden markov model. Rough Set Knowledge Technol., 6401: 361-370. DOI: 10.1007/978-3-642-16248-0_52

Albared, M., N. Omar, M. Ab Aziz and M. Ahmad Nazri, 2011. Developing a Competitive HMM Arabic POS Tagger Using Small Training Corpora. Intell. Inf. Database Syst., 6591: 288-296. DOI: 10.1007/978-3-642-20039-7_29

Allwein, E., R. Schapire and Y. Singer, 2001. reducing multiclass to binary: A unifying approach for margin classifiers. J. Mach. Res., 1: 113-141. DOI: 10.1162/15324430152733133

Attia, M., 2007. Arabic Tokenization System. Proceeding of ACL-Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Association for Computational Linguistics, Stroudsburg, PA, USA, pp: 65-72.

Attia, M., 2008. Handling Arabic Morphological and Syntactic Ambiguities Within The LFG Framework with a View to Machine Translation. 1st Edn., University of Manchester, Manchester, pp: 279.

Diab, M., 2007b. Improved arabic base phrase chunking with a new enriched POS tag set. Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, (Semitic '07), Association for Computational Linguistics Stroudsburg, PA, USA, pp: 89-96.

Diab, M., 2009. Second Generation AMIRA tools for arabic processing: fast and robust tokenization, pos tagging and base phrase chunking. Columbia University.

Diab, M., Hacioglu, K. and Jurafsky, D. 2007a. Automated methods for processing arabic text: From tokenization to base phrase chunking. Arabic Computat. Morphol., 38: 159-179. DOI: 10.1007/978-1-4020-6046-5_9

Diab, M., K. Hacioglu and D. Jurafsky, 2004. Automatic tagging of arabic text: From raw text to base phrase chunks. Proceedings of North American, Association for Computational Linguistics Stroudsburg, PA, USA, pp: 149-152.

Dukes, K., E. Atwell and A. Sharaf, 2010. Syntactic Annotation Guidelines for The Quranic Arabic Treebank. University of Leeds.

El Kholy, A. and N. Habash, 2010. Orthographic and morphological processing for english-arabic statistical machine translation. Columbia University.

Farghaly, A. and K. Shaalan, 2009. Arabic Natural Language Processing: Challenges and Solutions. ACM Transactions on Asian Language Information Processing. DOI: 10.1145/1644879.1644881

Grover, C. and R. Tobin, 2006. Rule-based chunking and reusability. University of Edinburgh.

Kanaan, G., A. Hammouri, Al-Shalabi, R. and M.Swalha, 2009. A new question answering system for the arabic language. Am. J. Applied Sci., 6: 797-805. DOI: 10.3844/AJAS.2009.797.805

Khoja, S., 2001. APT: Arabic Part-of-Speech Tagger. Lancaster University.

Li, X. and D. Roth, 2001 Exploring Evidence for Shallow Parsing. Proc. Workshop Comput. Nat. Language Learn., DOI: 10.3115/1117822.1117826

Mokhtar, S., J. Chanod and C. Roux, 2002. Robustness beyond shallowness: Incremental deep parsing. J. Natural Language 8: 121-144. DOI: 10.1017/S1351324902002887

Patrick, Y., 2009. Natural language understanding in controlled virtual environments. University Library.

Pierce, D., R, 2003. Cost-Effective Machine Learning Strategies for Shallow Parsing. 1st Edn., Cornell University, USA., pp: 183.

Ramshaw, L.A. and M.P. Marcus, 1995. Text chunking using transformation-based learning. Proceeding of the 3rd ACL Workshop on Very Large Corpora, May 23-23, Computation and Language, Lance Ramshaw pp: 82-94.

Shaalan, K., 2010. Rule-Based Approach in Arabic Natural Language Processing. Int. J. Inf. Commun. Technol., 3: 11-19.