A Comparative Evaluation of POS Tagging and N-Gram Measures in Arabic Corpus Resources and Tools

Sultan Almujaiwel *salmujaiwel@ksu.edu.sa*
College of Arts/Arabic Language Department
King Saud University
Riyadh, Saudi Arabia

Abstract

The purpose of this evaluation is twofold: an overview of the extent to which the functioning of the large-scale Arabic corpus resources examined serves the criteria of parts-of-speech tagging in the corpus design of linguistic data and to evaluate Arabic corpus analysis tools in terms of natural language processing statistics. The confusion matrix statistical method shows that some Arabic monitor corpora need further development, and the International Corpus of Arabic scores high levels on confusion matrices. There are nine Arabic corpus analysis tools under evaluation, and the attested reliable statistical outcomes are retrieved in terms of statistical algorithms for association measures. This is done by relying on one million empirically designated clean Arabic data to evaluate the association measures among the nine Arabic corpus analysis tools. The results presented at the end of this article indicate that the limitations could be tackled by evaluating the Arabic monitor Corpus resources rather than trusting them, and by implementing the new forms of programming rather than depending on the already-built natural Arabic language resources and tools.

Keywords: Arabic Corpus Resources, Arabic Corpus Analysis Tools, Corpus Linguistics, Confusion Matrices, Association Algorithms.

1. INTRODUCTION

Discussing all existing and available built-in Arabic corpus resources (ACRs) and Arabic corpus analysis tools (ACATs) is a significant challenge despite being uncommon compared to their counterparts for English. The purpose of this paper is to examine the functioning of the large-scale Arabic corpus resources that serve the criteria of parts-of-speech tagging in the corpus design of linguistic data and to evaluate Arabic corpus analysis tools in terms of natural language processing statistics.

Five general ACRs—KACSTAC [1], Tunisian Arabic Corpus TAC [2], [3], [4], Leeds' Querying Internet Corpora QIC [5], arabiCorpus [6], and International Corpus of Arabic ICA [7]—are evaluated. Three of them adopt the monitor corpus approach (representing different real texts of Arabic from different genres, domains, and times) [1], [6], [7]. The corpus design criteria and the search queries of the wild cards of a given Arabic root in these monitor corpora are reviewed and evaluated according to the literature of corpus linguistics and computational corpus linguistics. The evaluation of part-of-speech POS tagging systems is also given by measuring the performance of the main POS tagging—that is, the Confusion Matrix (CM).

The functionality of built-in tools and techniques for searching texts varies and needs to be addressed regarding which pre-processing functions and extended interfaces and windows show the results of any search and analysis processes selected by the user. These tools are known as ACPTs [8], aConCorde [9], AntConc [10], [11], WordSmith Tools [12], [13], Sketch Engine [14], [15], [16], MonoConc [17], KWIC [18], GraphColl [19], and LancsBox [19].

Six ACATs—Khawas (later referred to as ACPTs), aConCorde, AntConc, WordSmith Tools, Sketch Engine, and IntelliText Corpus Queries—are reviewed in terms of the following criteria: reading Arabic UTF-8 files, reading Arabic UTF-16 files, displaying Arabic diacritics, Arabic text in the right-to-left direction, normalizing hamza, Arabic interface, Arabic personal corpus (later do-it-yourself [DIY] corpus) [20]. These criteria are related to the pre-processing of Arabic corpus. The need for evaluating all existing ACATs in terms of their built-in corpus tools is essential. Thus, the elements that will be evaluated in this paper for ACATs are as follows: interface, window size, collocation window, file size, processing speed, functionality, and pre-processing options, and statistical packages.

2. ARABIC CORPUS RESOURCES (ACRs)

It is observed that using a monitor/large-scale corpus for the purpose of research is a matter of judgement [21]. If, and only if, the use of a corpus is a priority, then there are criteria that should be taken into account. These criteria are size, balance and representativeness. Table 1 shows these criteria for the five general ACRs. Among the genres that are common to several ACRs, the most frequent is newspapers. The linguistic data retrievals from this genre, apart from TAC and QIC, were made from web-based archived texts, thus leading to the conclusion that any sorts of texts from the web are the easiest medium to be obtained. The resources in TAC and QIC are typically from the web, retrieving Tunisian conversations and dialects. The issue of representativeness among the ACRs is somewhat difficult. Those that have included various genres are KACSTAC, ICA and arabiCorpus. Two corpora, namely TAC and QIC, cannot be analysed as the details of this information are not provided. However, a careful look at the proportions of the genres in KACSTAC, ICA and arabiCorpus reveals that none of them are better than any other in terms of whole search queries of words. Suffice to say that searching for a word from the web in QIC seems to be more reliable than KACSTAC, as the former constitutes 100% of the total 317m words while the latter makes up 1.6% of the total 1.2bn. In addition, textual Arabic resources in KACSTAC from Saudi Arabia makes up 34%, and 27% comes from another three countries: Syria, Iraq and Kuwait.

Name of Corpus	Medium	Size	Representativeness	Balance
KACSTAC	Written	Around 1.2bn	Newspapers, magazines, books, school textbooks, theses, periodicals, official documents, news agencies, web, edited (ancient) manuscripts	0.38, 0.12, 0.14, 0.01, 0.03, 0.02, 0.006, 0.008, 0.01, 0.25
TAC	Written, spoken	881,967	Web resources: blogs, phone conversations, folktales, internet forums, jokes, literature, opinion, plays, poem, political speeches, proverbs, radio, recipes, religion, SMS, Facebook, YouTube comments, forum posting, songs, sports, television dramas, web	1
QIC	Written	Around 317m	Wikipedia, web	1
ICA	Written	100m	Books, newspapers, electronic articles, theses	0.43, 0.29, 0.20, 0.08
arabiCorpus	Written, spoken	173,044,678	Newspapers, premodern, modern literature, Egyptian colloquial, non-fiction	0.780, 0.050, 0.060, 0.009, 0.101

TABLE 1: Size, Balance and Representativeness of The ACRs.

As far as the language resources of Arabic corpora are concerned, the research questions should determine which monitor corpus should be used that best represents the case being empirically examined where users cannot build their own linguistic corpora. The built-in word search queries are shown in Table 2. If the research focus is on Arabic dialects, the monitor corpora that should be looked at are TAC and the Egyptian colloquial sub-corpus of arabiCorpus. However, if the research focus is on a specific behavioural profile of a word used across the Arabic newspapers, the use of all of the newspaper genres in KACSTAC and arabiCorpus is necessary, as KACSTAC represents only four Arab countries, while arabiCorpus shows results from newspapers from Egypt, Morocco and Jordan.

Name of Corpus	Built-in word search queries
KACSTAC	- Interface (Arabic) - Countries - Time - Wildcard - Search in texts - Search in texts' titles - Word search distributions of genres and time - Concordancing (-15 n-gram +15) - Results save - Statistics (MI, MI3, t-score, dice, X^2, L (θ), z-score, log Dice)
TAC	- Word search - Categories - Search type (exact, stem or regular expressions)
QIC	- Interface (English) - Tag(s) search/ word search/ string search
ICA	- Interface (Arabic and English) - Arabic characteristics search - Wildcard - POS search (nouns, verbs, adjectives, pronouns, particles, conjunctions) - Parts of POS (exact, lemma, root, stem) - Morphological metrics - Numbers (singular, dual and plural) - Indefinite and definite articles - Masculine or feminine - Countries/ Topics - Concordancing (open n-grams)
arabiCorpus	- Interface (English) - Latin characteristics search - Arabic characteristics search - POS search (noun, adjective, adverb, verb and string) - Result summary - Citations (concordancing) - Subsections - Word forms (wildcards) - Collocations (1 *before*/*after* n-gram) - Citations save

TABLE 2: Word Search Queries in ACRs.

Where researchers need to search for Arabic newspapers which are issued in countries other than Arabic countries and need programming skills, one stands out: Spiderling [22]. The problem can also be solved by researchers who have no skills in programming languages, such as Spiderling, Python, Perl, etc., by employing the corpus-building tool of Sketch Engine's WebBootCat [14]. This technique allows the users of Sketch Engine to build a web corpus in many languages, including Arabic, by inserting particular seed words, URLs or websites. The amount of data retrieved by this function depends on the number of words the user requires at the time the licence subscription is paid. I built a web corpus using archived newspapers linked to the country of Algeria (https://www.djazairess.com) having subscribed to a 31 million word licence, and the process of building this corpus has reached that number. Looking at the size of classic Arabic texts that approximately date from the sixth century CE to the eighteenth century CE, they are not satisfactory despite so many books being freely available on electronic platforms and in TXT or DOC formats. For instance, al-Maktabah al-Shāmilah (http://shamela.ws/) offers 6,111 books, each of which can be exported as a DOC file, and they include words. A dependence on ACRs will not tackle all the potential social, scientific and humanitarian questions. Technical alternatives and applications are discussed further in this article. In addition, some vague genre classifications are found in KACSTAC: it is not clear if the periodicals are meant to be scientific journals or general journals and magazines. This has not been clarified in KACSTAC [1]. Other genres, such as books and edited manuscripts, are not as clearly classified as periodicals; in Arabic *al-kutub al-muḥaqqaqah* refers to those books that were written by hand in ancient times and were edited, examined and reissued in print in the modern period.

In terms of the importance of tagging and annotating the ACRs to make search query results more precise, TAC and QIC texts have not been tagged. KACSTAC is said to be tagged automatically for nouns and verbs. Such processing is good for Arabic roots that are amalgamated by inflectional affixes. For example, after ticking the option *verb* in the search query engine of KACSTAC, the root *qdr* and its frequencies cannot be determined in results as a verb. arabiCorpus provides five classes: noun, adjective, adverb, verb and string. Although advanced annotations [23] are helpful in Systematic Functional Linguistics, where the functions of text and discourse grammars, e.g., anaphoric and cataphoric (either the inner and outer pronominal referrers in texts) are annotated [24], none of the intended ACRs have been processed in detail. Hence, I will need to examine the word search queries in the three part-of-speech (POS) tagged ACRs by extracting all the results of the nominal and verbal forms of the root /q/-/d/-/r/ and measuring the positive and negative predictive actual/predicted values (Confusion matrix or CM) in order to come up with the error rate, meaning the relative difference between the true or false forms of nouns and verbs. The CMs of the actual and predicted classes of the nouns and the verbs of the selected root are provided in Table 3.

Corpus			**Predicted Frequency**		
KACSTAC			Nouns	Verbs	Recall
	Actual Frequency	Nouns	465,354	61,894	0.88
		Verbs	20,504	241,727	0.92
		Precision	0.96	0.80	
ICA					Recall
		Nouns	80,874	0	1.00
		Verbs	0	20,206	1.00
		Precision	1.00	1.00	
arabiCorpus					Recall
		Nouns	48,284	12,938	0.79
		Verbs	20,504	94,442	0.82
		Precision	0.70	0.88	

TABLE 3: CMs of the frequency average of the whole combined frequencies.

The analogues of what is truly or falsely nominal or what is truly or falsely verbal between the tested ACRs varied, despite the significant quantity of the absolute frequencies of each. By using the *caret* package in *R*, the train dataset has given the absolute frequency of what has been tagged as a noun or a verb for all the actually and naturally occurring morphosyntactic words with *qdr*, and the CM that makes the distinction between the predicted frequencies in which correct and incorrect predictions, located by the test dataset, form the following error rates: 11% in KACSTAC, 0% in ICA and 20% in arabiCorpus (Table 4).

ACRs	Accuracy	Precision		Sensitivity (Recall)		Error rate
		Nouns	Verbs	Nouns	Verbs	
KACSTAC	0.89	0.96	0.80	0.88	0.92	0.11
ICA	1.00	1.00	1.00	1.00	1.00	0.00
arabiCorpus	0.80	0.70	0.88	0.79	0.82	0.20

TABLE 4: The CM of actual and prediction results of all nominal and verbal *qdr* forms.

3. ARABIC CORPUS ANALYSIS TOOLS (ACATs)

Nine Arabic corpus tools are under evaluation, based on a plain untagged text format. This is because the tagged/annotated corpus gives the same results from each ACAT, as the statistical measures in it calculate the cases of tags rather than tokens. The challenge in evaluating these tools is the lack of precision in some functions such as the number of tokens. As the main purpose of these tools is to help users who are interested in analysing electronic texts, the common keys of search and analysis are based on the frequencies and concordances. This is not the pivot around which corpus studies revolve, as further advanced functioning is important. These functions, being the criteria of evaluation of the selected ACATs, are discussed in terms of a two-sided evaluation: the search where the file type, speed, word frequency and collocation window are designed, and the analysis regarding where the functionality and statistics are built in.

The experimental corpus I have designed and adopted contains exactly one million Arabic words, excluding numbers, punctuation marks and any symbols other than Arabic characters. The texts were gathered randomly from al-Maktabat al-Shāmila (http://shamela.ws/). Texts in this corpus are authentic, representing different eras of classic Arabic to test the ACATs and their functions in terms of frequency, concordance and statistics. Some corpus analysis tools are stated but with little information about their functionality, namely Sketch Engine, LinguaStream, AntConc, Upery, WordSmith Tools, Wmatric, Linguistic Inquiry and Word Count LIWC, BNCweb and WordStat [25]. Each tool produces a frequency profile of words and collocational behaviour, but LinguaStream provides a “visual assembly of modules…at various levels, from the morphological to the discursive” [25], and LIWC helps users analyse specific words conveying different sentiments and styles. New functionality concepts that enhance the features of the corpus analysis toolkits are set out [25], resulting in reengineering a toolkit they called superCAT in which the lexical POS are not processed as they tend to be infinite, as opposed to the sentence type in which the constructions of types and tokens in a sequence of POS tags are determined, but still have a tendency toward infinity. It can be argued that lexical diversity depends on the corpus representativeness by which lexical diversity fluctuates while the sequences of grammatical constructions come to a closure. However, only Sketch Engine, AntConc and WordSmith tools are suitable for processing Arabic texts.

4. CRITERIA OF EVALUATION

What a word is and how it is counted are defined in [26]. The designer of AntConc understands the way that processing tools process files in terms of the calculation of token frequency in the corpus, which is computationally determined by a single keyboard space button-press either side of each slot. This issue cannot be solved 100% in all ACATs considered in this present paper. This is because of the noise of Arabic morphosyntactic units that the ACATs try to detect. It is basically a matter of machine learning in which Arabic words are calculated by the concept of the space-button. It is impossible for Arabic types (the roots and stems of the different tokenised language characters) to be accurate in all ACATs. The reason is simply because of how a type (the origin character of its recurrently tokenised units) is processed in frequency number by the algorithms used in building each tool of the ACATs. In terms of solving the issue of corpus size, the designer of AntConc discussed four generations of software tools, outlining that the first and second generations achieved the capability of processing the ASCII characters in small-scale linguistic data but, in the second generation, such a process was possible using a personal computer [26]. These two generations were in use from the 1960s to the 1980s. The third generation extended into the 1990s, encountering obstacles in processing large-scale linguistic data that exceeded 100 million words, and with concerns about allowing data to be open to users. The fourth generation was ready-designed corpora uploaded on servers with their own corpus tools [27]. However, this has now changed. The corpora designed and hosted on a large server, such as corpus.byu.edu [28], CQPweb [29], SketchEngine [14], WMatrix [30], British National Corpus's Xara [31] and Bank of English [32], are limited for a trial use, but they are open for varying levels of subscription charge.

Since the review of the English corpus linguistic software tools in [26], ACATs (the Arabic linguistic software tools under evaluation in the present paper) have been enhanced further and can accept any DIY corpora (Table 5). All these ACATs can be used for Arabic corpora. Each ACAT is evaluated according to six criteria: file types, speed of processing, word frequency, collocation window, functionality, and statistical corpus linguistics. This is to come up with the limitations that need to be resolved in the corpus linguistic research. No matter whether the aim of processing the corpus for linguistic research is driven from the corpus, i.e., looking at the corpus itself and hypothesising for conducting a linguistic study, or is based on the corpus, i.e., testing pre-existing hypotheses and theories in linguistics [27], the objective of evaluating the ACATs according to the six criteria (Table 5) is to encourage corpus linguists to familiarise themselves with the programming by which the ACATs' limitations can be overcome.

4.1 File Type

The TXT file type is the easiest format, although it requires careful attention on the best way to prepare it for linguistic research processing. A format such as XML is recommended for the purpose of further encoding and advanced tagging and annotations for the texts. Other formats are provided for the annotated corpus, such as SGML and HTML in WordSmith Tools and SGML, COCOA and Helsinki in KWIC. However, the latter is not supported in most of the ACATs discussed in the present paper. Other formats, such as CSV, either the extension of Excel Sheet saved as a comma separated value or a TXT file are of importance in processing variables independently processed and separated given for the advanced statistical corpus processing. Some ACATs provide their own file types, such as WordSmith Tools, where the user can save the results of concordances, wordlists and keywords as a special WS file on the PC. However, the only ACAT that allows the user to upload many different formats of corpus files is Sketch Engine. Those ACATs that allow only TXT files are MonoConc, aConCorde and AntConc and ACPTs.

4.2 Speed

In terms of the PC RAM speed, it is known that the RAM of most PCs is 4GB and the price of the PC increases with greater RAM. However, the recommended minimum RAM capacity, according to my experience, should be at least 16GB. This facilitates processing corpus file(s) whose token numbers exceed 100m. Processing a 20m word corpus file, for example, in AntConc or ACPTs is not easy. The time it takes is very long, causing the user to become frustrated and increasing the likelihood of the ACAT application crashing.

My PC has a RAM of 16GB, and the speed for processing a one million TXT file varies from one ACAT to another. The tools that process the file instantly, as given in Table 5, are KWIC, MonoConc, aConCorde and LancsBox. Despite the latter being developed from GraphColl, the processing time was 18 seconds. As for the remaining tools, AntConc, ACPTs and WordSmith Tools run the TXT file in 9, 10 and from 20 seconds to 1 minute, respectively. Sketch Engine takes about 1.5 minutes, depending on the internet speed and the size of the corpus.

Tools	File type	Speed	Word frequency	Collocation window/span	Functionality	Statistics
KWIC	XML TXT	Instant	1,000,012	L5 n R5	- Concordance - Collocate - Stop word	Not built-in
MonoConc	TXT	Instant	1,000,016	L5 n R5	- Concordance - Collocate - Stop word - Networkable	Not built-in
aConCorde	TXT (UTF-8/ UTF-16/ MacArabic)	Instant	1m (82,963 types)	Not built-in	- Frequency list - Concordance	Not built-in
AntConc	TXT (UTF-8)	9 seconds	1,000,001 (82,961 types)	L20 n R20	- Concordance - Concordance plot - File view - Clusters/ N-Grams (rank, frequency, range, transitional probability between first and other words - Clusters) - Line breaks (space, \, %) - Collocates - Word list - Keyword list	Collocates: (MI, t-score) Keywords: log-likelihood and X^2
GraphColl	TXT (UTF-8)	18 seconds	1,000,016 (82,963 types)	Graph (L20 n R20)	- Frequency - Graph	Mu.groovy, MI, MI2, MI3, LogLik, Z, Dice, Log Dice, T, LogRatio, Minsens, DeltaP, Cohen
LancsBox		Instant	10,000,016	LOpen n ROpen	Frequency KWIC Whelk GraphColl Words Text	Mu.groovy, MI, MI2, MI3, LogLik, Z, Dice, Log Dice, T, LogRatio, Minsens, DeltaP
ACPTs	TXT. UTF-8	PC Ram 16g: 1m file takes 10 seconds	1m	5n-grams Concordance (L15 n R15)	Primary and reference corpus, preprocessing (remove diacritics, shadah, mad, numbers, symbols, Latin and normalise hamzah and taa marbutah), stop list, include list, document relative, frequency, word relative frequency, document frequency, word frequency	Chi square, weirdness coefficient, mutual information, log likelihood, z-score, t-score, dice coefficient, log dice

WordSmith Tools	Multiple formats (Unicode)	Between 20 sec to one minute	1m	L15 n R15	Concordance Keywords Word list	Type-token ratios and dispersion, MI
Sketch Engine	Multiple formats	Depends on Internet speed and size of subscription	998,000	L6 n R6	Word sketch Word sketch difference Thesaurus Concordance Wordlist N-grams Keywords One-click dictionary	T-score, MI, MI3, log likelihood, Minsens, log Dice, MI.log_f

TABLE 5: ACATs evaluated in the present study according to six criteria.

4.3 Word Frequency

When it comes to the word frequency, the resulting number depends on the design of the corpus. It is axiomatic that function words are more frequent than content words and will be more frequent in a 10 million corpus than a 5 million corpus. However, the question of having different numbers of words in a corpus, especially as these numbers change slightly from one ACAT to another, is problematic. Corpus design and word frequency are intertwined in terms of the reliability of the word count [33]. They raised awareness of what is called internal (linguistic) representativeness compared to external (situational) representativeness. The former is looking at linguistic behaviour in terms of morphological and syntactic information, as opposed to the latter being referred to by registers and the behaviour of their lexical items saturated for fluid situations. As in Table 5, why does the sampled corpus not remain the same in number? It increases slightly in one ACAT and decreases slightly in another, despite the sampled corpus (one million) designed by me being cleansed of punctuation marks, clitics, doubled spaces and symbols. However, by checking the first 100 words and the last 100 words in each ACAT, I found that the noticeably modest increase is due to the lines. This led me to consider the operation of line breaks in order to make the whole corpus into only one line, but the results for the word frequency remain constant. Another surprising finding is that the sampled corpus in a DOCX file shows that the word count is exactly the same as that given by Sketch Engine.

4.4 Collocation Window and Functionality

In terms of n-grams, they vary between 1 and 5 n-grams; I found that some corpora provided only a 5 n-gram span while other ACATs offered a span of collocates either side of *n* so the user can look through concordances without limitation. The reason for such different choices is because of the conventional corpus linguistic statistics by which the association measures are calculated. However, the need to have a concordance of L15 n R15 is conventional for pragmatic and discourse studies. The ACATs in which the collocation window extends for 5 n-grams either side of the node (the word under search) are KWIC and MonoConc. aConCorde does not provide such a function in its processing options. Those that provide the function of extending the collocation window span (concordancing) to L15 and R15 are ACPTs and WordSmith Tools. AntConc and GraphColl enable the user to have an L20 n R20. The LancsBox that is developed from GraphColl enables the user to have an open number of collocates associated with the node. Sketch Engine in its new version allows the user to have only six collocates either side of the node. In terms of the functionality, the ACATs that provide many options of corpus query functioning are AntConc, ACPTs and Sketch Engine, compared to the fewer provided functions found in KWIC and aConCorde (Table 5).

4.5 Statistics

Corpus linguistics is primarily based on quantitative approaches to language use. Some research in the field exploits the corpus for observations about using cases of phonetic variations or for informing a teacher how phrases behave. However, the basic research adopted from corpus linguistic perspectives should consider the corpus linguistic statistics [34], [35], [36] that go beyond the statistical packages built into most of the ACATs such as multi-level models and mixed-effects models [34]. Some corpus studies focus on basic statistics to make their studies readable and understandable to a broad readership. This can be found, for example, in [37]. Their study was based on an analysis of 12 British newspapers to sketch 19 salient Muslim-related events quantitatively and reproduce them qualitatively and in a diachronic frame. This is well-regarded, as such studies are clear to a wide range of target readers rather than the technical wording of only quantitatively advanced data analytics.

KWIC, MonoConc and aConCorde have no built-in corpus linguistic statistical packages. These statistics are entirely focal and essential in the corpus linguistics—namely chi-square (X^2), log-likelihood, weirdness coefficient, mutual information measures (MI, MI2, MI3), z-score (Z), t-score (T), log Dice, dice, mu_groovy, minimal sensitivity (Minsens), LogRatio, DeltaP, and Cohen. Some ACATs have been built with these statistical packages (Table 6) that calculate the significance of the association (node + collocate) in the linguistic data (Section 6).

5. ASSOCIATION MEASURES

In Table 6, all statistical formulae (given in [38]) are used to test the collocation relationship. Chi-square (goodness of fit test) and log-likelihood are contingency tabulated variables between different models (corpora or cases from different data). The former is used to measure the distributions of words in the corpus in a way that the associations of the words are significant or non-significant by chance, knowingly between two corpora or more, but it can be used for one corpus if it is used as two or more sub-corpora. The result of this measure is fit by p-values by which the confidence of the sample is assessed between observed values (absolute values) and expected values. Thus, it simply signifies whether the sample used is suitable for testing the hypotheses posed upon the selected corpus. The p-values extend from 0 to 1. The smaller the p-value, the more significant the expected data. Usually a p-value of .05 or less is accepted as significant. As for log-likelihood, the values are given in p-values, and it tests the best model of the sampled corpora under analysis. The weirdness coefficient is recommended to be used for comparing the association of two words between two corpora. Put simply, the results are between 1 and infinity. The former indicates that the collocation occurs only in the first corpus, but infinity indicates the opposite [39]. This algorithm is built in only Ghawwas. Mutual information Z and T are used to test the strong and weak relationships between two words occurring dependently and independently at the same time in the data.

Association measures	**Formula**
Chi-Square (X^2)	$\Sigma \frac{(O-E)^2}{E}$
Log-likelihood	$2 \times (O_{11} \times log \frac{O_{11}}{E_{11}} + O_{21} \times \ log \frac{O_{21}}{E_{21}} + O_{12} \times log \frac{O_{12}}{E_{12}} + O_{22} \times log \frac{O_{22}}{E_{22}})$
Weirdness coefficient	$\frac{O_{11}}{n_1} / \frac{O_{12}}{n_2}$
Mutual information (MI)	$log_2 \frac{O_{11}}{E_{11}}$
MI2	$log_2 \frac{{O_{11}}^2}{E_{11}}$
MI3	$log_2 \frac{{O_{11}}^3}{E_{11}}$
Z-score	$\frac{O_{11} - E_{11}}{\sqrt{E_{11}}}$
T-score	$\frac{O_{11} - E_{11}}{\sqrt{O_{11}}}$
Log Dice	$14 + log_2 \frac{2 \times 0_{11}}{R_1 + C_1}$
Dice	$\frac{2 \times O_{11}}{R_1 + C_1}$
Mu_groovy	$\frac{O_{11}}{E_{11}}$
Minimal sensitivity	$min\left(\frac{O_{11}}{C_1}, \frac{O_{11}}{R_1}\right)$
LogRatio	$log_2 \frac{O_{11} \times R_2}{O_{21} \times R_1}$
DeltaP	$\frac{O_{11}}{R_1} - \frac{O_{21}}{R_2}; \frac{O_{11}}{C_1} - \frac{O_{12}}{C_2}$
Cohen	$\frac{Mean_{in\ window} - Mean_{outside\ window}}{pooled\ SD}$

TABLE 6: Formulae of ACATs Statistics.

The values of Log Dice are from 1 upwards, when the value reaches 14 or more the relationship between the collocations is stronger. Dice is the opposite, where the strength of the collocation relationship is intertwined with the number of zeros in the decimals, i.e., the more zeros in the decimal the more significant/stronger the relationship.

Mu_groovy gives the divisional ratio between the observed frequency and the theoretically expected frequency of the collocation. LogRatio is an effect size that gives the different distribution of the collocation frequencies between data, and the proportional values become more significant as the uncommon collocation relationships become greater. The values of DeltaP appear in decimals of 1, showing the relative proportion of the collocation in the data, while Cohen is merely a weighted SD between data in order to come up with the value that shows the extent of the size effect.

6. DATA AND THE ASSOCIATION MEASURES

A one-million-word corpus sample was created for the purpose of investigating the frequency of the node, that of the node_collocate, and that of the collocate occurring independently. In addition, the experiment is based on evaluating the statistical association measures, which apparently vary from one ACAT to another, but some common measures are built in between some ACATs. However, not all ACATs are compiled with corpus linguistic statistics. Those that will be evaluated are AntConc, GraphColl, LancsBox, ACPTs, WordSmith Tools and Sketch Engine. The technique adopted for such an evaluation is based on choosing the node *sāḥil* (coast) as a node (*x*), occurring 175 times, and the collocate (*y*) *al-baḥr* (sea) occurring 974 times, and both (*xy*) as a collocation occurring 102 times in the one million sampled corpus (*n*) to be analysed. Then the statistical values of the association measures between the ACATs were assessed to provide the processing of the statistical corpus linguistic packages. I needed first to calculate the contingency table of co-occurrence frequencies of the bigram *sāḥil al-baḥr* (sea coast) as in Tables 7, 8 and 9.

a (O11) = $f(xy)$	b (O12) = $f(x\bar{y})$	$f(x*)$	R_1
c (O21) = $f(\bar{x}y)$	d (O22) = $f(\bar{x}\bar{y})$	$f(\bar{x}*)$	R_2
$f(*y)$	$f(*\bar{y})$	n	
C_1	C_2		

TABLE 7: Contingency table of a bigram *xy.*

a = 102	b = 175	277	R_1
c = 974	d = 998,749	999,723	R_2
1076	998,924	1m	
C_1	C_2		

TABLE 8: Contingency table of the bigram of the observed (*O*) frequencies.

a = 0.298052	b = 276.701948	277	R_1
c = 1075.70195	d = 998,647.298	999,723	R_2
1076	998,924	1m	
C_1	C_2		

TABLE 9: Contingency table of the bigram of the expected (E) frequencies.

The association measures calculated in the ACATs providing such a corpus of statistical measures are given in Tables 10, 11, and 12, showing the difference between the size of corpus and the respective numbers of nodes and collocates independently or adjunctly. Some measures are common between the tools that provide the statistical processing results. The values of the measures shared in common between AntConc, GraphColl, LancsBox, Ghawwas, WordSmith Tools, and Sketch Engine are MI and t-score (T) only. These values came up according to the number of nodes and of collocates, each of which occurred independently, and the number of the node + collocate occurring together. The node occurred 175 times and the collocate occurred 974 times in all of the tools, but the node occurred 196 times and the collocate occurred 1,022 times in Sketch Engine. The number of the node + collocate associated adjunctly occurred 102 times in AntConc, GraphColl, LancsBox, and Ghawwas and 115 times in WordSmith Tools. Technically it is hard to evaluate the system of WordSmith Tools in which the number of the node + collocate is higher while the number of times the node and the collocate occurred independently remained constant. On the contrary, the number of times both occurred independently is much higher in Sketch Engine. However, such a problem in word frequency might be tackled by using the standardised frequency measure [40] or simply reporting the normalised frequency.

Node	Collocate	Node+Collocate	MI	T-score
175	974	102	9.22501	10.08263

TABLE 10: Frequencies and association measures in AntConc.

GraphColl		LancsBox	
Node	175	Node	175
Collocate	974	Collocate	974
Node+Collocate	102	Node+Collocate	102
Mu_groovy	598.425533	Mu_groovy	598.320803
MI/MI2/MI3	9.22503/ 15.89745/ 22.56988	MI/MI2/MI3	9.22478/ 15.89720/ 22.56963
Log-likelihood	1111.818291	Log-likelihood	1187.942727
Z	246.648687	Z	246.627031
Dice	0.177546	Dice	0.177545
Log Dice	11.506262	Log Dice	11.506262
T	10.082628	T	10.461154
LogRatio	Error	LogRatio	9.384116
Minsens	102.000000	Minsens	0.104722
DeltaP	0.5819850	DeltaP	0.581984
Cohen	110.000000	-	-

TABLE 11: Frequencies and association measures in GraphColl and LancsBox.

ACPTs		WordSmith		Sketch Engine	
Node	175	Node	175	Node	196
Collocate	974	Collocate	974	Collocate	1,022
Node+Collocate	102	Node+Collocate	115	Node+Collocate	104
Chi-square	467.9485	-	-	-	-
Weirdness coefficient	0.9639	-	-	-	-
Log-likelihood	311.8543	Log-Likelihood	799.11	Log-likelihood	1229.613
-	-	LogRatio	-2.48	-	-
MI	2.5027	MI	9.45	MI	9.417
-	-	MI3	23.24	MI3	22.817
T-score	8.3175	T-score	10.89	T-score	10.183
Z-score	19.8007	Z	89.88	-	-
log Dice	1.7407	-	-	log Dice	11.450
Dice	2.0E-4	Dice	0.21	-	-
-	-	-	-	MI.log_f	43.826
-	-	-	-	Min.se	0.10176

TABLE 12: Frequencies and association measures in ACPTs, WordSmith and Sketch Engine.

If the calculation of MI is based on measuring the log^2 of the result of (node+collocate * size of corpus) / (node * collocate), that is, the result is $(102 * 1m) / (175 * 974) = 9.225004$. The MI2 and MI3 are close as the calculation of both is 18.450009 and 27.675014, respectively, and they are quite close to the counterpart values given by GraphColl, AntConc and LancsBox. However, all tools concerned are extremely close to this result of MI, except ACPTs whose MI is 2.5027. As to the measure t-score, O^{11}-E^{11}/$\sqrt{O11}$ ((O1-E1) / (O1)$^{1/2}$), that is $102 - 0.298052 \div \sqrt{102}$ =

6.672007. This value is closer to the results of t-score in ACPTs than in GraphColl, LancsBox or WordSmith Tools.

The log-likelihood association measure given by GraphColl, LancsBox, ACPTs, WordSmith Tools and Sketch Engine is different, but the values processed have the p-value < 0.00001. This is inexplicable despite the corpus size containing exactly one million words. The association measures Z, Dice and Log Dice are also different between GraphColl, LancsBox, ACPTs and WordSmith Tools.

For the Z association measure, the formula for the intended node_collocate is O11-E11/$\sqrt{E11}$ ((O1-E1) / (E1)$^{1/2}$), that is $102 - 0.298052 \div \sqrt{0.298052}$ = 6.664682. This value is far from the values given by GraphColl and LancsBox. However, for ACPTs, the Z value doubled three times, though not for WordSmith Tools, which achieved 89.88 whose *log* is 6.489928, the best tool in terms of the reliability of statistical value. For the computing shortcomings of Z in GraphColl and LancsBox they come closest to accuracy after WordSmith Tools, achieving the *log*$_2$: 7.946313 and the *log*$_{2:}$ 7.946187 respectively.

Log Dice versus Dice is set at the value of 14, that is, the number of the association occurs 14 times per million words, while Dice is simply referred to by small values that basically indicate that the more decimal places the more significant the collocation relationship is. The zero count in GraphColl, LancsBox and WordSmith Tools is only one, but there are three zeros in ACPTs. According to their formulae and the observed and expected frequencies:

$$\text{Log Dice:} 14 + log_2\, 2\frac{102}{1353} = 11.270478$$

$$Dice\ \frac{2\times O_{11}}{R_1+C_1} = 0.150776,$$

statistical measures given by ACPTs is far from the pinpoint accuracy of the results calculated above. In addition, the association measure of Chi-square given by ACPTs is not the final parameter, as it still needs to be calculated to obtain the p-value. The LogRatio measures given by LancsBox and WordSmith Tools, except GraphColl whose result gives a computing error, are inexplicable. By assessing such contradictory LogRatio values, according to the formula.

$$log2\frac{O11 \times R2}{O21 \times R1}$$

the result is 8.562074 which means that LancsBox Tools provided a closer value, as opposed to WordSmith Tools. The remaining association measures are given only by GraphColl and LancsBox. Cohen is only given by GraphColl. Mu_groovy and DeltaP are given by both ACATs, and the respective values are so similar between them with a tiny difference in decimals. The minsens values between these two ACATs are completely different.

The evaluation of the intended ACATs shows that depending on one tool might be misled, and using more than two tools will pave the way to make a comparison between the basic corpus linguistic statistics and the association measures. The optimal solution for the differences in the results could also be given by using R (a free software environment for statistical computing) for any further developments and judgements.

7. CONCLUSION

For the users of the ACRs and ACTAs, the necessity of comparing while processing the Arabic corpus is of the utmost importance. For the developers and users of ACATs, the tools that show the exact and similar results in terms of the statistical corpus linguistic packages and association measures seem to be more reliable for linguistic research.

The evaluation of ACRs and ACATs has been justified according to multiple criteria. For ACRs, their functionality in terms of the medium, size, representativeness, and the options for the corpus query search engine were displayed, and KACSTAC, ICA, and arabiCorpus were evaluated by testing the type and token of the root *q-d-r* and how CMs show that an ICA's value is a perfectly calculated reliable outcome. The differences between the ACRs in the results of their CMs of the frequency average of the whole derivational and inflectional frequencies of the root *q-d-r* and the statistical values built in the ACATs selected were evaluated carefully, and researchers and users should be familiar with the shortcomings of each. In addition, these ACRs are representatives of unbalanced genres, domains, and geographies. The results vary between ACRs, and these detected changes in the results are due in large part to the applications and tools adopted while in their design. For the ACATs, I have tested the association measures that are shared among the ACATs examined, coming up with some similarities and some wrongly processed outcomes. ACATs need careful attention. I found that the frequencies are close, but the corpus linguistic statistics are not accurate. All the ACATs are sophisticated in design to serve corpus research, but I urge those using them for Arabic texts/corpora to look over the results of their functionalities and statistical packages. It was seen that ACPTs is the lowest rated tool; the statistical values are not as accurate as those of the other tools. This should be taken into consideration, and this would suggest that researchers who are familiar with corpus research consider more than two ACATs for the sake of evaluation and comparison on one hand and accuracy on the other hand. Corpus linguistics is a promising field, combining natural language processing and linguistic research, and both are extended to produce a productive experimental research environment that meets the conditions of accountability, falsifiability, and replicability, so one ACAT might not be highly reliable. This leads to the conclusion that one statistics-packaged tool is likely to not achieve a high confidence level of accountability. There is nothing 100% certain in statistical corpus linguistics, and the machine learning and designing simple data from texts or annotated data of texts or standalone software computer corpus processing tools might be exposed to unreplicability in some cases. Corpus research is based on huge datasets and attempts to make the result as precise as possible with clear cutting-edge software tools and techniques. This is normal because the analytical results of linguistic data are based on probabilities.

In terms of next steps, I recommend that using ACRs restricts corpus research, and such a problem requires technical skills in preparing corpora that meet the research questions and hypotheses. Moreover, the ACATs should be compared in terms of the association measures. For example, if a user wants to analyse a one million-word corpus, it is recommended to use more than two standalone corpus processing tools for comparison and accuracy if the user is not familiar with one of the programming languages and the cutting-edge tools that help sort, count, display, and tabulate the grams of association. This will guarantee a wide skim through the differences of the results from the corpus that is under investigation for any linguistic or social scientific questions/hypotheses.

8. REFERENCES

[1] "KACSTAC," KACST Arabic Corpus, [On-line]. Available: https://corpus.kacst.edu.sa/ [Dec. 14, 2019].

[2] K. McNeil and M. Faiza. "Tunisian Arabic Corpus TAC," [On-line]. Available: http://www.tunisiya.org [Dec. 14, 2019].

[3] J. Younes, H. Achour and E. Souissi. "Constructing linguistic resources for the Tunisian dialect using textual user-generated contents on the social web," in Proceedings of the 1st International Workshop on Natural Language Processing for Informal Text NLPIT in conjunction with The International Conference on Web Engineering (ICWE). 2015.

[4] "TAC," Tunisian Arabic Corpus, [On-line]. Available: http://www.tunisiya.org/ [Oct. 25, 2019].

[5] S. Sharoff. “Creating general-purpose corpora using automated search engine queries,” in WaCky! Working papers on the web as Corpus, 2006, pp. 63-98.

[6] D. Parkinson. “arabiCorpus,” [On-line]. Available: http://arabicorpus.byu.edu/. [Dec. 10, 2019].

[7] “ICA,” The International Corpus of Arabic, [On-line]. Available: http://www.bibalex.org/ica/ar/login.aspx. [Oct. 30, 2019).

[8] S. Almujaiwel and A. Al-Thubaity. “Arabic Corpus Processing Tools for Corpus Linguistics and Language Teaching,” in Proceedings of the International Conference on the Globalization of Second Language Acquisition and Teacher Education (G-SLATE), 2016, pp. 103-108.

[9] A. Roberts. “aConCorde.” [On-line]. Available: http://www.andy-roberts.net/coding/aconcorde [Dec. 14, 2019].

[10] L. Anthony. “AntConc: design and development of a freeware corpus analysis toolkit for the technical writing classroom,” in Proceedings of International Professional Communication Conference (IPCC), 2005.

[11] L. Anthony. “AntConc.” [On-line]. Available: http://www.antlab.sci.waseda.ac.jp/ [Dec. 14, 2019].

[12] M. Scott. “Developing Wordsmith.” International Journal of English Studies, vol. 8, no. 1, pp. 95-106, 2008.

[13] M. Scott. “WordSmith Tools version 6.” [On-line]. Available: http://www.lexically.net/wordsmith [Dec. 14, 2019].

[14] A. Kilgarriff, A. “Sketch Engine,” [On-line]. Available: http://www.sketchengine.co.uk/ [Oct. 25, 2019].

[15] A. Kilgarriff, P. Rychly, P. Smrz and D. Tugwell. “The sketch engine,” in Proceedings of the Euralex, 2004.

[16] “Sketch Engine,” Lexical Computing: Language corpus management and query system, [On-line]. Available: https://www.sketchengine.eu/ [Dec. 14, 2019].

[17] M. Barlow. “MonoConc.” [On-line]. Available: http://www.monoconc.com/ [Dec. 14, 2019).

[18] S. Tsukamoto. “KWIC Concordance.” [On-line]. Available: http://dep.chs.nihon-u.ac.jp/english_lang/tukamoto/kwic_e.html [Dec. 14, 2019].

[19] V. Brezina, T. McEnery, and S. Wattam. “Collocations in context: A new perspective on collocation networks.” International Journal of Corpus Linguistics, vol. 20, no. 2, pp. 139-173, 2015.

[20] Alfaifi, A., & Atwell, E. “Comparative evaluation of tools for Arabic corpora search and analysis”. International Journal of Speech and Technology, vol. 19, no. 2, pp. 347-357, 2016.

[21] S. Atkins, J. Clear and N. Ostler. “Corpus design criteria.” Literary and Linguistic Computing, vol. 7, no. 1, pp. 1-16, 1991.

[22] T. Arts, Y. Belinkov, N. Habash, A. Kilgarriff and V. Suchomel. "arTenTen: Arabic corpus and word sketches." Journal of King Saud University – Computer and Information Sciences, vol. 26, no. 4, pp. 357-371, 2014.

[23] S. Th. Gries and A. L. Berez. "Linguistic annotation in/for corpus linguistics," in Handbook of linguistic annotation, N. Ide and J. Pustejovsky, Eds. Berlin & New York: Springer, 2017, pp. 379-408.

[24] S. Bartsch, R. Eckart, M. Holtz and E. Teich. "Corpus-based register profiling of texts from mechanical engineering," in Proceedings of the Corpus Linguistics Conference (CL2005). 2005.

[25] K. Cohen, W. Baumgartner Jr and I. Temnikova. "SuperCAT: The (New and Improved) Corpus Analysis Toolkit," in Proceedings of the International Conference on Language Resources & Evaluation (LREC2016), 2016, pp.2784-2788.

[26] L. Anthony. "A critical look at software tools in corpus linguistics," *Linguistic Research*, vol. 30, no. 2, pp. 141-161, 2013.

[27] McEnery, T. and Hardie, A. Corpus linguistics: Method, theory and practice. Cambridge: Cambridge University Press, 2012, pp. 5-48.

[28] Davies, M. "corpus.byu.edu." [On-line]. Available: https://corpus.byu.edu/corpora.asp [Dec. 14, 2019].

[29] A. Hardie. "CQPweb." [On-line]. Available: http://cwb.sourceforge.net/cqpweb.php [Dec. 14, 2019].

[30] P. Rayson. "Wmatrix." [On-line]. Available: http://ucrel.lancs.ac.uk/wmatrix/ [Dec. 20, 2019].

[31] L. Bernard and T. Dodd. "Xara: an XML aware tool for corpus searching," in: Proceedings of the Corpus Linguistics Conference D. Archer, P. Rayson, A. Wilson and T. McEnery, Eds. Lancaster: University of Lancaster, 2003, pp. 142–144.

[32] "The Bank of English," WordbanksOnline, [On-line]. Available: https://wordbanks.harpercollins.co.uk/ [Dec. 20, 2019].

[33] D. Miller and D. Biber. "Evaluating reliability in quantitative vocabulary studies: The influence of corpus design and composition." International Journal of Corpus Linguistics, vol. 20, no. 1, pp. 30-53, 2015.

[34] S. Th. Gries. "The most underused statistical method in corpus linguistics: Multi-level (and mixed-effects) models." Corpora, vol. 10, no. 1, pp. 95-125, 2015.

[35] S. Th. Gries. "Quantitative designs and statistical techniques," in The Cambridge handbook of English corpus linguistics. D. Biber and R. Reppen, Eds. Cambridge: Cambridge University Press, 2015, pp. 50-71.

[36] S. Th. Gries. Ten lectures on quantitative approaches in cognitive linguistics: Corpus-linguistic, experimental, and statistical applications. Leiden & Boston: Brill, 2017.

[37] C. Gabrielatos, T. McEnery, P. Diggle and P. Baker. "The peaks and troughs of corpus-based contextual analysis." International Journal of Corpus Linguistics, vol. 37, no. 2, pp. 151-175, 2012.

[38] V. Brezina. Statistics in Corpus Linguistics: A Practical Guide. Cambridge: Cambridge University Press, 2018.

[39] S. Almujaiwel. “Grammatical construction of function words between old and modern written Arabic: A corpus-based analysis.” Corpus Linguistics and Linguistic Theory, vol. 15, no. 2, pp. 267-296, 2019.

[40] M. Brysbaert, P. Mandera and E. Keuleers. “The Word Frequency Effect in Word Processing: An Updated Review”. Current Directions in Psychological Science, vol. 27, no. 1, 2018, pp. 47-50.