Article:
Alfaifi, A and Atwell, ES orcid.org/0000-0001-9395-3764 (2016) Comparative evaluation of tools for Arabic corpora search and analysis. International Journal of Speech Technology, 19 (2). pp. 347-357. ISSN 1381-2416

https://doi.org/10.1007/s10772-015-9285-5

Comparative Evaluation of Tools for Arabic Corpora Search and Analysis

Abdullah Alfaifi[1]
Al Imam Mohammad Ibn Saud Islamic University (IMSIU)
scayga@leeds.ac.uk

Eric Atwell[2]
University of Leeds
e.s.atwell@leeds.ac.uk

Abstract

As the number of Arabic corpora is constantly increasing, there is an obvious and growing need for concordancing software for corpus search and analysis that supports as many features as possible of the Arabic language, and provides users with a greater number of functions. This paper evaluates seven existing corpus search and analysis tools based on eight criteria which seem to be the most essential for searching and analysing Arabic corpora, such as displaying Arabic text in its right-to-left direction, normalising diacritics and Hamza, and providing an Arabic user interface. The results of the evaluation revealed that three tools: Khawas, Sketch Engine, and aConCorde, have met most of the evaluation criteria and achieved the highest benchmark scores. The paper concluded that developers' conscious consideration of the linguistic features of Arabic when designing these three tools was the most significant factor behind their superiority.

Keywords: Arabic, corpus, concordance, usability

Introduction

A number of tools exist for searching and analysing Arabic corpora. Choosing a suitable tool for supporting Arabic seems to be difficult and requires a comparison between multiple tools, as their potentials and functions differ in terms of handling Arabic. This paper attempts to present a fundamental comparative evaluation of seven tools which are described as supporting multiple languages including Arabic. The purpose of this evaluation is twofold. First, to help users of Arabic corpora to confidently select the most appropriate tool for their corpus-based research; and second, to draw the attention of developers to the aspects that most need to be taken into account in further improving their tools in order to better support Arabic text.

[1] http://www.comp.leeds.ac.uk/scayga
[2] http://www.comp.leeds.ac.uk/eric

Background

Many tools are used for searching and analysing corpora. They generally provide some basic functions (e.g. frequent words and concordances), whereas some of these tools have more functions and statistics such as collocations, n-gram/clusters, keywords, etc. A number of these search and analysis tools are web-based, e.g. The Sketch Engine (Kilgarriff et al., 2004; Kilgarriff, 2014), IntelliText Corpus Queries (Wilson et al,. 2010; Sharoff, 2014), CQPweb at Lancaster (Hardie, 2012, 2014), so in order to use them, researchers need to be persistently online. Other tools are PC-based, so they can be downloaded on computers and used offline, such as the KACST Arabic Corpora Processing Tool "Khawas" (Al-thubaity et al., 2013, 2014), aConCorde (Roberts et al., 2006; Roberts, 2014), AntConc (Anthony, 2005, 2014a,b), WordSmith Tools (Scott, 2008, 2012). The developers of these tools assert that Arabic is one of the languages supported by their tools; therefore, we included the newest versions of these tools in this evaluation.

With respect to Arabic corpora, their number is constantly increasing. For some examples see Al-Sulaiti & Atwell (2006), Al-Sulaiti (2010), Alansary et al. (2007), Atwell & Hardie (2013) and Al-Khalifa and Al-Thubaity (2014). Some of these Arabic corpora are searchable online and have their own analysis tools; other Arabic corpora are open source and can be downloaded to users' PCs. Previous surveys have reviewed concordance tools but not specfically for Arabic corpora, for example Wiechmann and Fuhs (2006) reviewed ten corpus concordance programs tested on English corpora. Other surveys have covered Arabic text analysis resources, for example Atwell et al (2004) reviewed a sample of tools for Arabic morphological analysis and Part-of-Speech tagging, Machine-Readable Dictionaries, and corpus visualization tools as well as concordancing. Thus, there is need for a survey focused on Arabic corpus search and processing tools that support as many features as possible of the Arabic language, and that provide users with a greater number of functions..

Methodology

In this paper, seven tools designed to search and analyse corpora were selected to be evaluated against eight criteria. Each of these tools was evaluated separately against each benchmark. The evaluation was repeated, with the second one conducted two months after the first, on the same tool versions used in the first evaluation, in order to be sure that the criteria were properly covered. One of the tools was not available in the first evaluation, but the opportunity was taken to include it in the second. A

sample of Arabic corpus texts was used in two formats, UTF-8 and Unicode. More details about the evaluation method appear in the following sections.

Tools investigated

This paper includes seven tools:

1. The KACST (King Abdulaziz City for Science and Technology) Arabic Corpora Processing Tool "Khawas" 3.0 (Al-thubaity et al., 2013, 2014)
2. aConCorde 0.4.3 (Roberts et al., 2006; Roberts, 2014)
3. AntConc 3.4.0 (Anthony, 2005, 2014a, 2014b)
4. WordSmith Tools 6.0 (Scott, 2008, 2012)
5. The Sketch Engine (Kilgarriff et al., 2004; Kilgarrif 2014)
6. IntelliText Corpus Queries (Wilson et al,. 2010; Sharoff, 2014)
7. CQPweb at Lancaster (Hardie, 2012, 2014)

As mentioned previously, the tools selected were designed to support Arabic along with other languages. There may be further software programs beyond those that the researchers selected for evaluation, and more can be included in an extended evaluation in the future.

Evaluation criteria

Given the fact that functions of the tools examined here differ from one to the next, most of the criteria used were based on linguistic features, particularly those related to Arabic. While many benchmarks could be examined in an evaluation of these tools, eight points were selected that seemed to be the most essential criteria for searching and analysing Arabic corpora. Wiechmann and Fuhs (2006) reviewed ten corpus concordance programs; they mainly used general software evaluation criteria such as: platform, price, ease of installation, help, and performance. They also compared a range of functionalities, such as: input/output formats, text search, frequency and collocation outputs. However all bar one of the systems they evaluated were developed for English text, and they did not investigate in detail how well the systems adapted to corpora in other languages such as Arabic. There was one exception: aConCorde was explicitly targeted at Arabic.

1. Reading Arabic text files in UTF-8 format

This point examines whether the tools being tested are able to read Arabic text files in UTF-8 format and show the characters correctly. According to Burnard (2005), the Unicode Standard has three UTFs: UTF-16, UTF-8 and UTF-32 (in chronological order), UTF-16 is known as "Unicode", and

UTF-8 is superior to the other two, so Burnard recommends using UTF-8 as a universal format for data exchange in Unicode, and for corpus construction.

2. Reading Arabic text files in Unicode format

This is to examine whether the tools are able to read Arabic text files in Unicode format and show the characters correctly. In spite of the fact that UTF-8 is recommended for corpus construction (Burnard, 2005), Microsoft applications advise the user to use UTF-16. Notepad is one application in particular upon which many people rely to create and save their corpus files. However, when a user tries to save a text including Arabic characters in different encoding formats such as ANSI, Notepad advises the user to use "Unicode" (which refers to UTF-16), ignoring UTF-8, which is also available among the other encoding formats. Thus, corpora tools may or may not be able to handle the Unicode encoding format besides the UTF-8 format that is most widely used in corpus construction. For this reason the ability of reading Arabic characters in Unicode was included in this evaluation.

3. Displaying diacritics correctly

The ability to show Arabic diacritics—if there are any—is tested under this point, e.g. "هِمّةً". Displaying diacritics might be essential in some cases, particularly with similar forms that cannot be distinguished if they have no diacritics, e.g. ذهبَ (past tense of the verb "went") and ذهبْ (noun: "gold").

4. Displaying Arabic text in the correct direction (right to left)

As Arabic is written from right to left, the tools were examined to ascertain whether they can show Arabic text in the correct direction, particularly in concordances, where the contexts must also be ordered correctly.

5. Normalising diacritics

This is to check if the tool is able to normalise the diacritics, so that the user has an option to search Arabic texts which include diacritics using a single word form in the query. For example, if a text includes the word "هِمّةً" (with diacritics) and the word "همة" (without diacritics), is the user able to search for both using the single form "همة"? This is significant in searching Arabic corpora, as one form may have several sub-forms with diacritics. Unless the diacritics are normalised, the user may face difficulty in counting them, and accordingly in combining them into a single query.

6. Normalising Hamza "ء"

This is similar to the previous benchmark. Here, we check to see whether the tool has the ability to normalise words that have Hamza, so the user has an option to search Arabic texts, which include Hamza using a single word form in the query. For example, if a text includes the word "إلى" (with Hamza) and the word "الى" (without Hamza), is the user able to search for both using the single form "الى"?

7. Providing Arabic user interface

This is to determine whether these tools provide an Arabic user interface for Arabic users, as some researchers may not be able to use a tool should its interface be in a language different from their mother tongue, and thus cannot benefit from its functions.

8. Enabling users to upload or open their Arabic personal corpora

Researchers may desire to use particular Arabic corpora, or even build their own corpora from scratch and use some tools to search and analyse these resources. Therefore, the tools here are examined to see whether they accept external data files.

Evaluation sample

The current evaluation was based on a sample from the Arabic Learner Corpus (ALC)[1]. This open-source corpus was developed at Leeds University, and is comprised of 282,732 words collected from learners of Arabic in Saudi Arabia over the course of 2012 and 2013. The corpus includes written and spoken data produced by 942 students from 67 different nationalities studying at pre-university and university levels (Alfaifi et al., 2014).

We randomly selected a few files from ALC to be used as a sample of our examination. The evaluation includes testing as to whether Arabic characters can be read in UFT-8 and Unicode formats, and since ALC files are already in Unicode format, we made an additional copy of the sample in UTF-8.

Results and discussion

Each tool will be explored in detail with its benchmark results, which will then be followed by a brief overall comparison that has been provided at the end of this section.

[1] The ALC may be accessed here: http://www.arabiclearnercorpus.com

Khawas

The KACST (King Abdulaziz City for Science and Technology) Arabic Corpora Processing Tool "Khawas" (Al-thubaity et al., 2013, 2014) is an open-source tool that Abdulmohsen Al-thubaity and his team at KACST developed specifically for processing Arabic language with an Arabic/English interface. It is free to download and can provide analysis including frequency lists, concordance N-grams lexical patterns and corpora comparison. Khawas was developed using Java which means it can be run on many operating systems. The developers claim that this tool works with texts from all languages in principle, and it was tested on Arabic, English, and French (Al-thubaity & Al-Mazrua, 2014).

Khawas was able to read Arabic texts in UTF-8 format; however this was not the case with texts in Unicode, as nothing readable was displayed. Khawas is set to remove diacritics by default in order to normalise the text, but they can be shown by changing the settings. Consequently, searching the data follows the diacritics settings; i.e. if the diacritics are shown, the search results will include those words that match the query word including its exact diacritics, and the same words with other diacritics will be excluded. Khawas displays words in the correct right to left orientation (Figure 1); however, some words or parts of words were missed from concordances when the tool was run on Microsoft Windows (Figure 2). All of the missing words appeared when Khawas was run on Mac OS X. This tool has an option to normalise Hamza, which enables both those words that have, or should have but are missing Hamza, to be included in the search results. Users need to be aware that Hamza normalisation means all Hamzas will be removed from the texts, so the query word should not include one, otherwise no results will be returned. Khawas has an Arabic/English interface, and this tool was developed to open external data, i.e. users are able to open their personal corpora on Khawas. This tool garnered 7 points out of 8 in the benchmark evaluation (Table 1).

Figure 1: Khawas Shows Arabic words in a right-to-left order

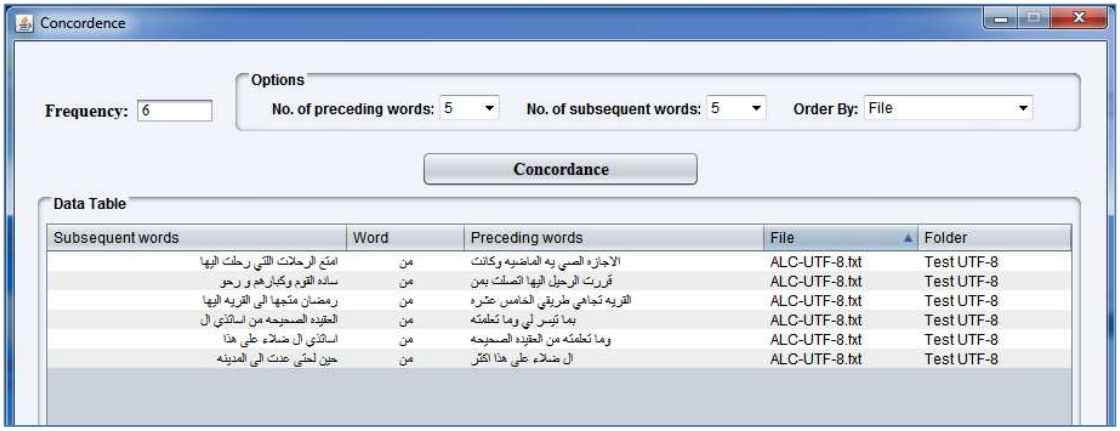

Figure 2: Some Arabic words were missed from concordances when Khawas was run on Windows

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	Yes
2. Reading Arabic Unicode files	No
3. Displaying Arabic diacritics	Yes
4. Displaying Arabic text in a right-to-left direction	Yes
5. Normalising diacritics	Yes
6. Normalising Hamza	Yes
7. Providing Arabic interface	Yes
8. Enabling Arabic personal corpus	Yes
Score	7/8

Table 1: Benchmark score of the Khawas tool

aConCorde

aConCorde (Roberts et al., 2006, Roberts, 2014) is a free tool which was created by Andrew Roberts in his spare time while he was a PhD student at Leeds University. It is relatively basic in comparison to the others included in this paper, as it only provides users with concordances and a word frequency list. However, one of the distinctive features of aConCorde is "the provision of an Arabic interface. Not only does this provide Arabic translations for all the menus, buttons etc., but even switches the entire application layout to right-to-left" (Roberts et al., 2006, 6).

aConCorde was able to read Arabic texts in both UTF-8 and Unicode formats. It also correctly shows Arabic diacritics as well as words in a right-to-left direction (Figure 3). However, diacritics and Hamza cannot be normalised, so the search results will literally match the query word. aConCorde has an Arabic/English interface, and enables users to open their

personal corpora. aConCorde achieved 6 points in this evaluation (Table 2).

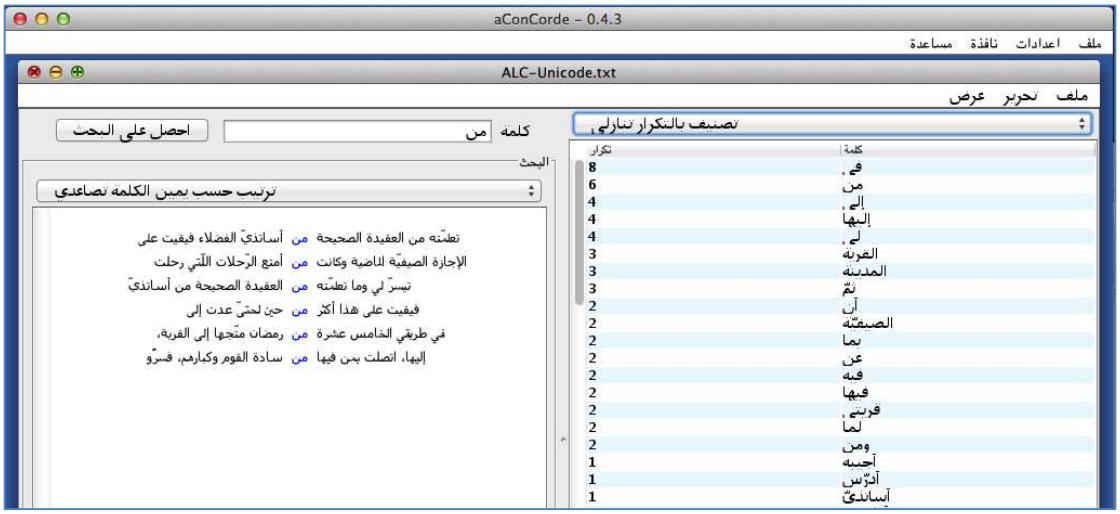

Figure 3: Frequency and concordances in aConCorde

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	Yes
2. Reading Arabic Unicode files	Yes
3. Displaying Arabic diacritics	Yes
4. Displaying Arabic text in a right-to-left direction	Yes
5. Normalising diacritics	No
6. Normalising Hamza	No
7. Providing Arabic interface	Yes
8. Enabling Arabic personal corpus	Yes
Score	6/8

Table 2: Benchmark score of the aConCorde tool

AntConc

AntConc (Anthony, 2005, 2014a, 2014b) is a free corpus analysis tool developed by Laurence Anthony, a professor in the faculty of science and engineering at Waseda University, Japan. AntConc provides users with concordances, clusters/n-grams, collocates, word list, and keyword list. This tool was "developed in Perl using ActiveState's PerlApp compiler to generate executables for the different operating systems" (Anthony, 2014b, 1).

Although AntConc reads Arabic texts in UTF-8 and Unicode formats, it behaves unexpectedly when the user clicks on any of the text words. Diacritics were displayed within the texts; however, AntConc does not normalise diacritics or Hamza. Additionally, columns in the concordances screen were shown in the opposite direction, as the right side should be the

left and vice versa (Figure 4). AntConc does not provide an Arabic interface, only English is available. Users are able to open their corpora on this tool. AntConc was awarded four of eight points in this benchmark evaluation (Table 3).

Figure 4: Columns of Arabic concordances in AntConc were shown in the opposite direction

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	Yes
2. Reading Arabic Unicode files	Yes
3. Displaying Arabic diacritics	Yes
4. Displaying Arabic text in a right-to-left direction	No
5. Normalising diacritics	No
6. Normalising Hamza	No
7. Providing Arabic interface	No
8. Enabling Arabic personal corpus	Yes
Score	4/8

Table 3: Benchmark score of the AntConc tool

WordSmith Tools

WordSmith Tools (Scott, 2008, 2012) is a commercial project developed by Lexical Analysis Software Ltd. The user can download the complete package with no registration code, but it will run in demo mode which will only show a sample of the output. WS Tools are developed for use on Mac, Linux or Windows, with an emulator for Windows. These tools provide users with a word list, concordances, and keywords, and they support many languages, including Arabic. WordSmith Tools even has an Arabic manual[1]; however, the interface of these tools is only in English.

WordSmith Tools were able to read Arabic texts in both UTF-8 and Unicode formats, and they also display Arabic text correctly in the right-

[1] The manual can be accessed here: http://www.lexically.net/wordsmith/step_by_step_Arabic6/index.html

to-left direction. However, WordSmith Tools did not put the diacritics in their correct positions (Figure 5). Instead, they are put on small circles, e.g. ◌ّ, ◌ً, ◌ِ or ◌ُ. Diacritics and Hamza were not normalised in this tool, so similar words with differences in diacritics and/or Hamza will not be retrieved in the results. As mentioned above, WordSmith Tools do not have an Arabic interface, as the only language available is English. Users can open their corpora files on these tools. The evaluation resulted in 4 out of 8 points for WordSmith Tools (Table 4).

Figure 5: Diacritics do not appear in their correct positions in WordSmith Tools

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	Yes
2. Reading Arabic Unicode files	Yes
3. Displaying Arabic diacritics	No
4. Displaying Arabic text in a right-to-left direction	Yes
5. Normalising diacritics	No
6. Normalising Hamza	No
7. Providing Arabic interface	No
8. Enabling Arabic personal corpus	Yes
Score	4/8

Table 4: Benchmark score of the WordSmith Tools

Sketch Engine

The Sketch Engine (Kilgarriff et al., 2004, 2014) is a commercial web-based tool for corpus analysis developed by Lexical Computing Ltd. In addition to the corpora searching tool, the users are provided with corpora in many languages including Arabic. Along with the usual features of such tools (e.g. concordance, word lists, key words, collocation, and corpus comparison), Sketch Engine has some unique features such as Word

Sketches that provide summaries of a word's grammatical and collocational behaviour, Word Sketch Difference to compare and contrast words visually, and WebBootCat, which lets users create specialised corpora from the Web.

The Sketch Engine correctly read Arabic texts in both UTF-8 and Unicode formats, and displayed Arabic texts in the proper right-to-left direction. Diacritics and Hamza were normalised when using the built-in Arabic Segmenter and Tagger (Figure 6), so researchers can use a single word form for those words with differences in diacritics and Hamza; however, the diacritics will not show throughout if they are normalised. The Sketch Engine interface can be used in several languages, but Arabic is not yet included. Sketch Engine provides users with a large number of corpora in many languages, and also accepts personal corpora via upload in several file formats. When it came to the criteria of this evaluation, Sketch Engine obtained 7 out of 8 possible points (Table 5).

Figure 6: Sketch Engine removed the diacritics when normalising the texts

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	Yes
2. Reading Arabic Unicode files	Yes
3. Displaying Arabic diacritics	Yes
4. Displaying Arabic text in a right-to-left direction	Yes
5. Normalising diacritics	Yes
6. Normalising Hamza	Yes
7. Providing Arabic interface	No
8. Enabling Arabic personal corpus	Yes
Score	7/8

Table 5: Benchmark score of the Sketch Engine web tool

IntelliText Corpus Queries

IntelliText Corpus Queries (Wilson et al,. 2010, Sharoff, 2014) is a web-based system developed by the Centre for Translation Studies (CTS) at the University of Leeds for the purpose of facilitating and enhancing teaching and research in various areas of the humanities. IntelliText provides a number of corpora including Arabic, as well as a number of functions to search these corpora, such as concordances, collocations, affixes, compare frequencies, key words, and phrases.

IntelliText Corpus Queries enables users to upload their own corpora in several languages. Arabic is not one of them, although this tool includes some built-in Arabic corpora. Uploading UTF-8 and Unicode files of Arabic is unfortunately not supported, however. In the built-in Arabic corpora, Arabic texts were displayed in the correct direction, right to left, and diacritics were presented correctly (Figure 7), but diacritics and Hamza were not normalised, and the search results therefore do not include the query form that shows differences in diacritics or Hamza. The interface of IntelliText is available only in English. The score IntelliText achieved in this evaluation is 2 of 8 possible points (Table 6).

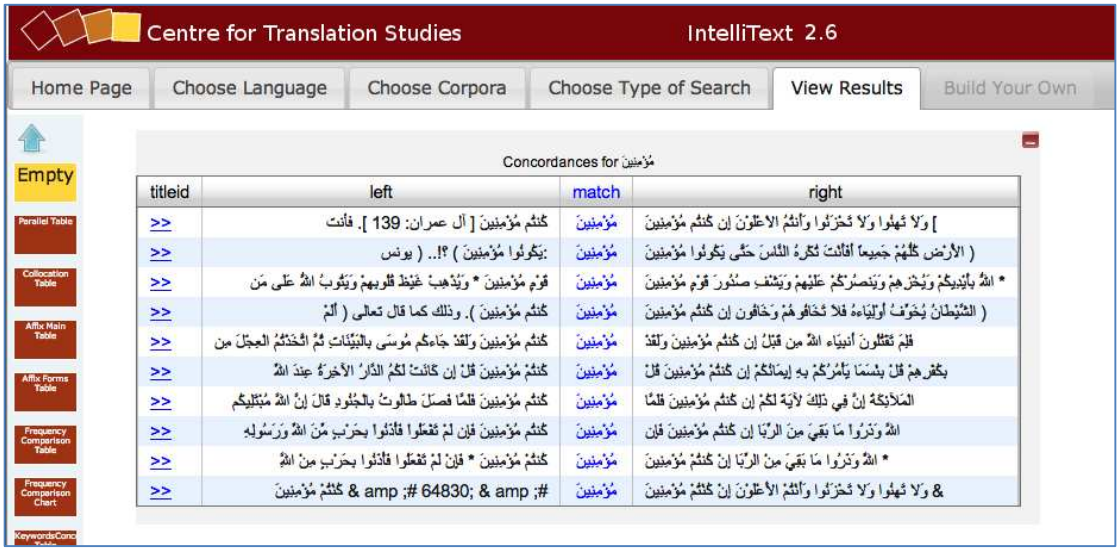

Figure 7: Diacritics displayed correctly in IntelliText Corpus Queries

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	No
2. Reading Arabic Unicode files	No
3. Displaying Arabic diacritics	Yes
4. Displaying Arabic text in a right-to-left direction	Yes
5. Normalising diacritics	No
6. Normalising Hamza	No

7. Providing Arabic interface	No
8. Enabling Arabic personal corpus	No
Score	2/8

Table 6: Benchmark score for IntelliText Corpus Queries

CQPweb at Lancaster

CQPweb (Evert, 2010) is a front-end to the IMS Open Corpus Workbench (CWB). The CQPweb software has been installed at a number of websites for use by corpus linguists, for example at Beijing Foreign Studies University[1] and at the University of Lisbon[2]. For this comparison of tools for Arabic Corpora search and analysis, we evaluate the CQPweb server run at Lancaster University by Andrew Hardie[3] (2012, 2014), probably the best-known CQPweb server for corpus linguistics research and teaching. We do not attempt to evaluate the full potential functionality of the CQPweb software or the IMS Open Corpus Workbench. The aim of CQPweb at Lancaster is to support research and teaching at Lancaster University, so access to this tool is partially restricted. However, researchers from other institutions can be allowed to use it as well, and with no charge. CQPweb provides functions such as concordance, frequency lists, and keywords, and it has many corpora in several languages, including Arabic.

The CQPweb software reads corpora from UTF-8 (not UTF-16). However, Uploading own corpora is restricted to administrators and those users who have this privilege, only Andrew have such privileges on CQPweb at Lancaster. CQPweb does have some built-in Arabic corpora. Searching in these corpora revealed that diacritics were shown correctly (Figure 8), and it correctly displays right-to-left text. CQPweb is a pure search system, it does not have normalisation modules, Diacritics and Hamza thus cannot be not normalised by this tool. The interface is available only in English. This means the tool meets just 2 out of 8 benchmarks in terms of evaluating its suitability for searching and analysing Arabic corpora (Table 7).

[1] It can be accessed from: http://124.193.83.252/cqp/
[2] It can be accessed from: http://alfclul.clul.ul.pt/CQPweb/
[3] It can be accessed from: https://cqpweb.lancs.ac.uk/

Figure 8: Diacritics displayed correctly in The CQPweb tool

Evaluation criteria	Applicability
1. Reading Arabic UTF-8 files	Yes
2. Reading Arabic Unicode files	No
3. Displaying Arabic diacritics	Yes
4. Displaying Arabic text in a right-to-left direction	Yes
5. Normalising diacritics	No
6. Normalising Hamza	No
7. Providing Arabic interface	No
8. Enabling Arabic personal corpus	No
Score	3/8

Table 7: Score of CQPweb

Comparing the results

Comparing all results of the evaluation reveals some significant points as follows:

1. Although none of the tools examined fulfilled all the evaluation criteria and achieved 8 points, three tools (Khawas, aConCorde and Sketch Engine), met more than 75% of the criteria and achieved the highest scores (Table 8).

	PC-based tools				Web-based tools		
Evaluation criteria	Khawas	aConCorde	AntConc	WS Tools	Sketch Engine	IntelliText	CQPweb at L.
1. Reading Arabic UTF-8 files	✓	✓	✓	✓	✓		✓
2. Reading Arabic Unicode files		✓	✓	✓	✓		
3. Displaying Arabic diacritics	✓	✓	✓		✓	✓	✓
4. Arabic text in R-to-L direction	✓	✓		✓	✓	✓	✓
5. Normalising diacritics	✓				✓		
6. Normalising Hamza	✓				✓		
7. Providing Arabic interface	✓	✓					
8. Arabic personal corpus	✓	✓	✓	✓	✓		

Score	7/8	6/8	4/8	4/8	7/8	2/8	3/8

Table 8: Comparison of the tools included in this evaluation

2. The most significant commonalities that Khawas, aConCorde, and Sketch Engine share are that they paid more attention to the features of Arabic such as diacritics and Hamza, specifically in Khawas and Sketch Engine, which have the highest points (7 for each), and Arabic was one of the languages that these tools were developed for, Khawas and aConCorde in particular.

3. Khawas and aConCorde are PC-based software while Sketch Engine is a web-based tool. While there is no difference in terms of the basis of the tools (PC or web) with regard to handling Arabic language, taking Arabic features into consideration when developing these tools may help to make them more appropriate for Arabic corpora.

4. Both Khawas and Sketch Engine are strong competitors as tools for searching and analysing Arabic corpora. Khawas provides an Arabic interface which might be a significant factor to some users, while this was the only shortcoming in Sketch Engine. By contrast, Khawas reads only text files in the UTF-8 format, whereas Sketch Engine can read many types of data files (e.g., .doc, .docx, .html, .pdf, .ps, .tar.gz, .txt, .xml, .zip, and other formats). Sketch Engine can also download the content of a website and store it as a corpus, and text from any external source can be pasted into the tool. Such flexibility helps when there is a need to use a diversity of data resources.

Conclusion

Seven tools for searching and analysing Arabic corpora were covered and evaluated against eight criteria. The results showed that three of these tools met most of the evaluation criteria and achieved high scores, 6 or greater, while the others ranged between 2 and 4. The paper highlighted the need to improve the current tools, as well as create new tools more appropriate for use with Arabic corpora, that provide more functions compatible with features of the Arabic language, such as diacritics and Hamza. It revealed also that although PC-based tools had higher scores than those based on web, Sketch Engine was a strong competitor to the PC-based tools, particularly Khawas. This may indicate that in principle there are no significant technical differences between PC-based and Web-based tools in terms of handling Arabic language. What is required, therefore, is that concordance developers in general pay more attention to the unique features of Arabic language.

Acknowledgements

The authors would like to thank the developers, Abdulmohsen Althubaity, Andrew Roberts, Laurence Anthony, Mike Scott, Adam Kilgarriff, James Wilson, and Andrew Hardie for their valuable comments and suggestions to improve the quality of the paper.

References

Alansary, Sameh, Magdy Nagi & Noha Adly. 2007. Building an International Corpus of Arabic (ICA): Progress of Compilation Stage. Paper presented at the Seventh Conference of Language Engineering ESOLEC (5-6 December 2007), Cairo, Egypt.

Alfaifi, Abdullah, Eric Atwell & Ibraheem Hedaya. 2014. Arabic Learner Corpus (ALC) v2: A New Written and Spoken Corpus of Arabic Learners. In S. Ishikawa (Ed.), Learner corpus studies in Asia and the world. Papers from LCSAW2014, (Vol. 2). Kobe, Japan: School of Languages and Communication, Kobe University, pp77-89.

Al-Khalifa, Hend & Abdulmohsen Al-Thubaity. 2014. Proceedings of the Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools. Reykjavik, Iceland.
http://www.kacstac.org.sa/osact/proceedings.rar

Al-Sulaiti, Latifa & Eric Atwell. 2006. The design of a corpus of contemporary Arabic. International Journal of Corpus Linguistics 11: 135–171.

Al-Sulaiti, Latifa. 2010. Arabic Corpora. 10 August 2012. The University of Leeds, Latifa Al-Sulaiti's Homepage: http://www.comp.leeds.ac.uk/eric/latifa/arabic_corpora.htm

Al-thubaity, Abdulmohsen, Marwa Khan, Manal Al-Mazrua & Maram AlMoussa. 2013. New Language Resources for Arabic: Corpus Containing More Than Two Million Words and a Corpus Processing Tool. Proc IALP International conference on Asian Language Processing, Urumqi, pp67-70.

Al-thubaity, Abdulmohsen & Manal Al-Mazrua. 2014. Khawas: Arabic Corpora Processing Tool USER GUIDE. Retrieved 6 April 2014, from: http://sourceforge.net/projects/kacst-acptool/files/?source=navbar

Al-thubaity, Abdulmohsen, Marwa Khan, Manal Al-Mazrua & Maram AlMoussa. 2014. KACST Arabic Corpora Processing Tool "Khawas" [Computer Software]. Retrieved 6 April 2014, from: http://kacst-acptool.sourceforge.net/

Anthony, Laurence. 2005. AntConc: design and development of a freeware corpus analysis toolkit for the technical writing classroom. Proceedings of IPCC International Professional Communication Conference, pages 729 - 737.

Anthony, Laurence. 2014a. AntConc, (Version 3.4.2) [Computer Software]. Tokyo, Japan: Waseda University. Available from http://www.antlab.sci.waseda.ac.jp/

Anthony, Laurence. 2014b. AntConc 3.4.2 - Readme. Tokyo, Japan: Waseda University. Available from http://www.laurenceanthony.net/software/antconc341/AntConc_readme.pdf

Atwell, Eric & Andrew Hardie (editors). 2013. Proceedings of WACL'2 Second Workshop on Arabic Corpus Linguistics. http://www.comp.leeds.ac.uk/eric/wacl/wacl2proceedings.pdf

Burnard, Lou. 2005. Metadata for corpus work. In M. Wynne (Ed.), Developing Linguistic Corpora: a Guide to Good Practice. Oxford: Oxbow Books. pp30-46.

Evert, Stefan. 2010. CQPweb [Computer Software]. Erlangen, Germany: Friedrich-Alexander-Universität Erlangen-Nürnberg. Available from http://cwb.sourceforge.net/

Hardie, Andrew. 2012. CQPweb - combining power, flexibility and usability in a corpus analysis tool. International Journal of Corpus Linguistics, 17(3), pp380-409

Hardie, Andrew. 2014. CQPweb at Lancaster [Computer Software]. Retrieved 6 April 2014, https://cqpweb.lancs.ac.uk/

Kilgarriff, Adam, Pavel Rychly, Pavel Smrz & David Tugwell. 2004. The Sketch Engine. In the proceedings of the Euralex, 6-10 July 2004, Lorient, France.

Kilgarriff, Adam. 2014. Sketch Engine [Computer Software]. Retreived 6 April 2014, http://www.sketchengine.co.uk/

Roberts, Andrew. 2014. aConCorde [Computer Software]. Retrieved 6 April 2014, http://www.andy-roberts.net/coding/aconcorde

Roberts, Andrew, Latifa Al-Sulaiti & Eric Atwell. 2006. aConCorde: Towards an open-source, extendable concordancer for Arabic. Corpora, 1(1), pp39-60

Scott, Mike. 2008. Developing Wordmsith. International Journal of English Studies. Vol.8(1), pp95-106

Scott, Mike. 2012. WordSmith Tools version 6 [Computer Software], Liverpool: Lexical Analysis Software, http://www.lexically.net/wordsmith

Sharoff, Serge. 2014. IntelliText Corpus Queries [Computer Software]. Retrieved 6 April 2014, http://corpus.leeds.ac.uk/itweb/htdocs/Query.html

Wiechmann, Daniel & Stefan Fuhs. 2006. Concordancing software. Corpus Linguistics and Linguistic Theory Journal, Volume 2, Issue 1. Pages 107-127.Wilson, James, Anthony Hartley, Serge Sharoff & Paul Stephenson. 2010. Advanced corpus solutions for humanities researchers. Proceedings of PACLIC 24, Sendai, Japan.