aConCorde: towards a proper concordance for Arabic

Andrew Roberts, Dr Latifa Al-Sulaiti and Eric Atwell
School of Computing
University of Leeds
LS2 9JT
United Kingdom
{andyr,latifa,eric}@comp.leeds.ac.uk

July 12, 2005

Abstract

Arabic corpus linguistics is currently enjoying a surge in activity. As the growth in the number of available Arabic corpora continues, there is an increased need for robust tools that can process this data, whether it be for research or teaching. One such tool that is useful for both groups is the concordancer — a simple tool for displaying a specified target word in its context. However, obtaining one that can reliably cope with the Arabic language had proved extremely difficult. Therefore, aConCorde was created to provide such a tool to the community.

1 Introduction

A concordancer is a simple tool for summarising the contents of corpora based on words of interest to the user. Otherwise, manually navigating through (often very large) corpora would be a long and tedious task. Concordancers are therefore extremely useful as a time-saving tool, but also much more. By isolating keywords in their contexts, linguists can use concordance output to understand the behaviour of interesting words. Lexicographers can use the evidence to find if a word has multiple senses, and also towards defining their meaning. There is also much research and discussion about how concordance tools can be beneficial for data-driven language learning (Johns, 1990).

A number of studies have demonstrated that providing access to corpora and concordancers benefited students learning a second language. Just as lexographers and linguists can find insights into grammatical structure by studying concordance output, so can language learners (Dodd, 1997). Cobb *et al.* (2001) detailed the improved rates of vocabulary acquisition when using a software tool of which a concordancer was a major component, for example.

The simplicity and usefulness of concordancers is such that they should be a valuable tool for any linguist. Unfortunately, as is the general trend within corpus linguistics, research priorities have focused on European languages. This has meant that language resources for other widely spoken languages such as

Arabic are somewhat lacking. For example, when this project began, to the best of our knowledge, there were no readily available concordancers that function correctly with the Arabic language. Considering Arabic is spoken by approximately 250 million people, it was felt that Arabic linguists deserved such a simple and useful tool. Thus, the aConCorde project was created.

2 Why is Arabic Concordance Hard?

One would have to assume that the reason there are few Arabic concordance tools available is because it is considerably more difficult to process than the other languages, such as European languages. However, this is simply not the case nowadays. The two most commonly perceived difficulties with Arabic language processing are:

1. Unique Arabic script. The cursive nature of the script means that letters are represented differently depending on whether they occur at the beginning, middle or end of a word. Characters within words are always joined, and never written individually.

2. Arabic is written from right-to-left, as opposed to the majority of other languages that are, of course, written left-to-right.

The process of transliteration is a common approach to resolving these issues. The Buckwalter system (amongst others) will convert Arabic script to an equivalent based on the Roman alphabet (Buckwalter, 2002). It also means that the text orientation is also converted. From a computational perspective, this has historically been a necessary step, due to computers long being restricted to ASCII (or similar) character sets (i.e., Roman alphabets). However, thanks to the popularity of the Unicode standards since the early 1990s, most modern operating systems and products support multi-lingual text encoding, and also the fonts to display Unicode characters are increasingly common. Implementation of bi-directional text components are also part of most modern operating systems.

Due to these developments, there is little reason why existing concordance tools have not been adapted to take advantage of the above technologies in order to render the aConCorde project redundant even before it had begun! There was one implementation that sought to provide multi-lingual concordancing, called xconcord (Ogden and Bernick, 1996; Boualem *et al.*, 1999), developed at the Computing Research Laboratory, New Mexico University. The CRL produced their own graphical components that were able to display Unicode text and they worked successfully at displaying Arabic text. Despite xconcord being ahead of its time when first developed in 1996, it did not become a mainstream application. Although it would still be a useful tool today, it is hindered due to it being written for the Sun Solaris platform. This is not widely used, especially by the average linguist, as this system is typically used on expensive high-powered workstations and servers. Therefore, it was not possible to experiment with it as resources required were not available. Development ceased many years ago, but the program can be downloaded[1].

[1]http://crl.nmsu.edu/software/

Where the real difficulty lies in Arabic concordance is not the displaying of results, but the inadequacy of searching for Arabic word stems. Commonly when performing concordance searches, the user wants to see the usage of words all from the same stem. Concordancers typically offer *wildcard* searches, for example, *perform** (where the asterisk means any character zero or more times) would match *perform*, *performs*, *performer*, *performers*, *performing*, etc. However, with Arabic, the morphology is substantially more complex than English. Arabic words are derived from a root of three consonants. Each root has a set of patterns which can add additional characters as either an affix, a prefix or even infix. An example would be the transliterated Arabic root *ktb* (write). Patterns associated with that root can produce words semantically linked, including verbs like *kataba* (he wrote) and *naktubu* (we write), and many nouns like *kitAb* (book), *maktuub* (letter) and *maktaba* (library)[2]. However, performing a wildcard search like **k*t*b** would return lots of matches, many of which are not associated with the *ktb* root, but also happen to have the same consonants.

Another issue is that written Arabic does not contain vowels — the main exception being the Qur'an. This leaves a great deal of ambiguity that is only resolved when looking at the context of a given word. An English example may be *'fr'* that could be *for*, *fir*, *fur*, *far*, *four*, *fear*, *fair*, *fire*, *afar*, *afore*, etc. Yet, if you can only search for *'fr'* even though you are only interested in *'far'*, it would clearly be frustrating to have to read through irrelavant results. It is unlikely that these specific issues will be resolved by generic concordancers.

3 Arabic Concordance with WordSmith, MonoConc, Xaira and aConCorde

For the task of concordance, the majority of people would turn to WordSmith (Scott, 2004) or MonoConc (Lawler, 2000). Both are commercial products and are well established tools for text analysis. They are only available on the Microsoft Windows platform. Xaira (Bernard, 2004) is a relative newcomer but stems from the well-known SARA toolkit. This section primarily seeks to compare the ability of retrieving concordance output from Arabic corpora using the established tools.

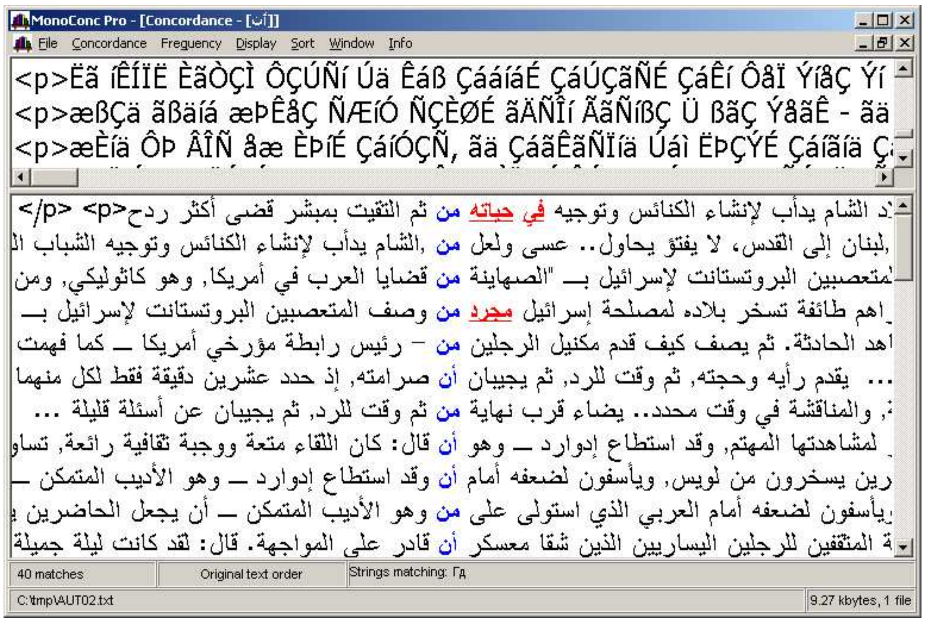
The corpus being used to experiment on is the newly released Corpus of Contemporary Arabic (CCA) (Al-Sulaiti, 2004). This corpus is annotated according to the TEI standards using XML markup, and all files are encoded in the 8-bit Unicode standard, UTF-8.

3.1 MonoConc

Developed by linguist Michael Barlow for language teachers and researchers, MonoConc has proved to be very popular for language analysis, as it comes with many additional features, such as collocation discovery and tag-sensitive searches . On the MonoConc website[3], it states: *"Some users have been using*

[2]Many sources cover Arabic grammar, however, issues about Arabic and computing are well introduced by Khoja (2003) and de Roeck (2002)

[3]`http://www.monoconc.com`


MonoConc Pro - [Concordance - [ىأ]]
File Concordance Frequency Display Sort Window Info

```
<p>Ëã íÊÍÏË ÈãÒÇÌ ÔÇÚÑí Úä Êáß ÇáäíáÉ ÇáÚÇãÑÉ ÇáÊí ÔåÏ Ýíåǃ Ýí
<p>æßÇä ãßäíá æÞÊåÇ ÑÆíÓ ÑÇÈØÉ ãÄÑÎí ÃãÑíßÇ Ü ßãÇ Ýåãê - ãä
<p>æÈíä ÔÞ ÂÎÑ åæ ÈÞíÉ ÇáíÓÇÑ, ãä ÇáãÊãÑÏíä Úáì ËÞÇÝÉ Çáíãíä Ç
```

```
د الشام يدأب لإنشاء الكنائس وتوجيه في حياته من ثم التقيت بمبشر قضى أكثر ردح<p> </p>
لبنان إلى القدس، لا يفتؤ يحاول.. عسى ولعل من ,الشام يدأب لإنشاء الكنائس وتوجيه الشباب ال
متعصبين البروتستانت لإسرائيل بـــ "الصهاينة من قضايا العرب في أمريكا, وهو كاثوليكي, ومن
اهم طائفة تسخر بلاده لمصلحة إسرائيل مجرد من وصف المتعصبين البروتستانت لإسرائيل بـــ
اهد الحادثة. ثم يصف كيف قدم مكنيل الرجلين من – رئيس رابطة مؤرخي أمريكا ــ كما فهمت
... يقدم رأيه وحجته, ثم وقت للرد, ثم يجيبان أن صرامته, إذ حدد عشرين دقيقة فقط لكل منهما
, والمناقشة في وقت محدد.. يضاء قرب نهاية من ثم وقت للرد, ثم يجيبان عن أسئلة قليلة ...
لمشاهدتها المهتم, وقد استطاع إدوارد ــ وهو أن قال: كان اللقاء متعة ووجبة ثقافية رائعة, تساو
رين يسخرون من لويس, ويأسفون لضعفه أمام أن وقد استطاع إدوارد ــ وهو الأديب المتمكن ــ
يأسفون لضعفه أمام العربي الذي استولى على من وهو الأديب المتمكن ــ أن يجعل الحاضرين
ة المثقفين للرجلين اليساريين الذين شقا معسكر أن قادر على المواجهة. قال: لقد كانت ليلة جميلة
```

```
40 matches    Original text order    Strings matching: Гд
C:\tmp\AUT02.txt                      9.27 kbytes, 1 file
```

Figure 1: Arabic concordance using MonoConc. Discovered collocates are underlined, although they should not be there! For unknown reasons the text in the top pane did not display using Arabic fonts.

MonoConc/Pro with Arabic. I don't have any details of their procedures." Unfortunately, despite initial optimism, using Arabic texts with MonoConc was not totally successful. Issues that arose during experiments were:

Unicode support it appeared that MonoConc does not support texts encoded in UTF-8. It was therefore necessary to convert texts to the Windows Arabic Codepage-1256 before the program would display the Arabic text correctly.

Incorrect concordance order perhaps the most significant problem for performing Arabic concordance is that the output is in the wrong order! (See figure 1.) Due to the right-to-left nature of Arabic, MonoConc does not take this into account when displaying results. For an English reader, it would be equivalent to seeing the output: "on the mat. [sat] The cat", rather than "The cat [sat] on the mat." (where the bracketed word is the target word.) This was observed by Hoogland during the Nijmegen Dutch-Arabic Dictionary Project, "*This seemed a serious shortcoming in the beginning, but soon we experienced that we got used to this very quickly.*" (Hoogland, 2003) Hoogland *had* to get used to it because it was better to have broken concordance rather than none at all. It is clearly far from ideal — it would be unlikely to be tolerated for teaching purposes as it would confuse the learner.

Admittedly, when the concordance results are saved to a file, and this file is viewed with a text editor, the ordering issue disappears. However, this output has its own problems, the first being that viewing is not as *nice* as there is no alignment around the target word. Secondly, MonoConc adds extra information before each match, which makes the output somewhat noisy.

'Noisy collocates' a strange artifact appeared in the concordance output whenever a match included a discovered collocate. It was default behaviour for the software to highlight found collocates. However, the only problem was that these collocates were actually inserted into the wrong position within the concordance context, and as a result caused a certain amount of interference. This was remedied by turning off the option to highlight collocates, after which the highlighted words would then vanish from the context.

Addition of unwanted terms despite a simple search term being submitted within MonoConc, sometimes it included matches within the results that are not equal to the target word. This behaviour appears to be random as it only occurs with some words and not others.

3.2 WordSmith

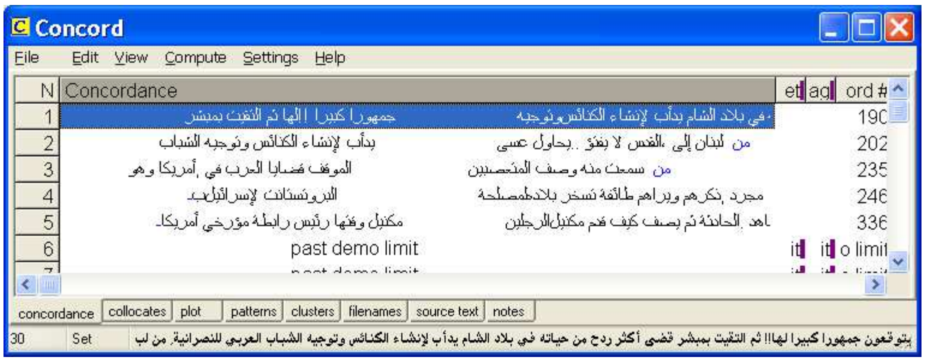
WordSmith was first released in 1996 and is still developed by linguist Mike Scott. WordSmith is currently at version 4 and sports three tools: Wordlist, Keyword and Concord. The latter is the one of interest for this report, but it should be obvious what the others do. The documentation notes that WordSmith supports Unicode which obviously lends itself to the analysis of the CCA. A demo version is available, which was used for this report. It retains all the functionality of the full version, but significantly limits the number of results returned when performing queries.

A degree of success was achieved when using WordSmith, in that it was possible to get the tool to display Arabic script — which means its Unicode support was working. Unfortunately, it suffered the common issue of displaying the prior and posterior contexts in the opposite order required for a right-to-left language. The matches are displayed in two columns, the left column must be read from right-to-left first, and then the reader can continue with the second column, which begins with the target word, followed by the rest of the context (see figure 2).

3.3 Xaira

Xaira is a new tool designed to supercede SARA — an application for text analysis on the British National Corpus. Xaira is no longer tied to the one corpus, instead opting for a more generalised approach. It takes advantage of Unicode and XML technologies to help achieve this (Bernard and Dodd, 2003). At the time of writing, its first version is still in beta testing, but will be released as a full product in the near future. It can perform complex searches as it can filter results according to the XML annotation. For example, a spoken corpus is commonly annotated with information such as the gender of the current speaker. Therefore, Xaira is able to perform searches based on male language usage and compare it to female usage.

Xaira does not suffer the same problem as WordSmith or MonoConc in that it displays the entire concordance in the correct order (see figure 3.) However,



```
N  Concordance                                                                                    et  ag  ord #
1  جمهورا كبيرا ا الها ثم التقيت بمبشر          في بلاد الشام يدأب لإنشاء الكنائس وتوجيه              190
2  يدأب لإنشاء الكنائس وتوجيه الشباب            من لبنان إلى القدس لا يفتؤ . يحاول عسى               202
3  الموقف قضايا العرب في أمريكا وهو             من سمعت منه وصف المتعصبين                          235
4  البروتستانت لإسرائيلـ                        مجرد ذكرهم ويراهم طائفة تسخر بلادهامصلحة             246
5  مكتبل وقتها رئيس رابطة مؤرخي أمريكا           اهد الحادثة ثم يصف كيف قدم مكتبل الرجلين             336
6                past demo limit                                                                it  it  o limit

30   Set   يتوقعون جمهورا كبيرا لها!! ثم التقيت بمبشر قضى أكثر ردح من حياته في بلاد الشام يدأب لإنشاء الكنائس وتوجيه الشباب العربي للنصرانية. من لب
```

Figure 2: Arabic concordance as displayed by WordSmith.

in order to get to the point of looking at results, the user must go through a relatively long procedure using an additional tool to prepare the source texts into the format expected by the software.

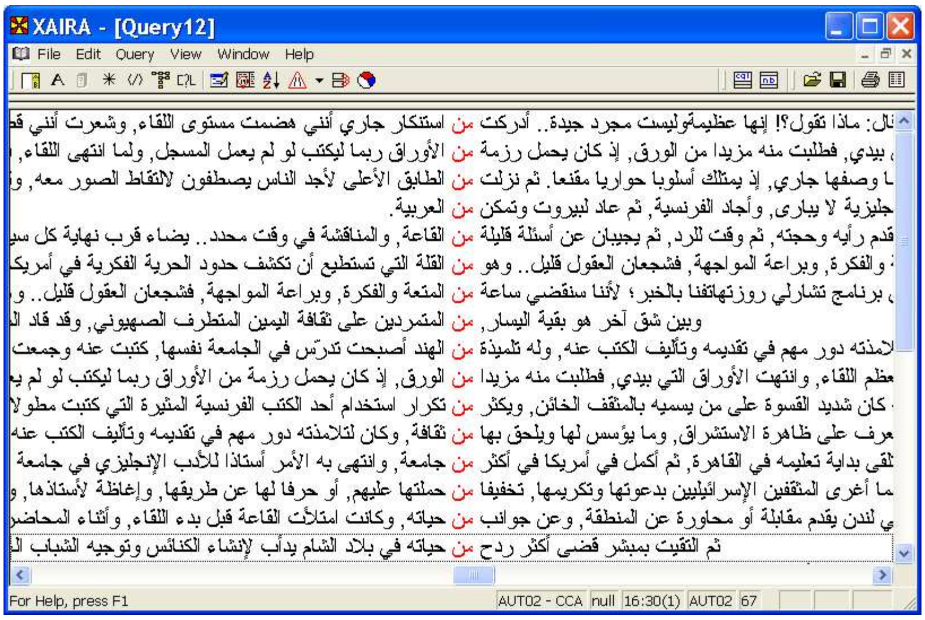
3.4 aConCorde

aConCorde is neither a commercial product nor a research project. It was (and still is) developed by Andrew Roberts, a PhD. student, in his spare time. In terms of features, it is relatively basic when compared to the systems already discussed in this section. However, due to the design aim being to ensure it was as multi-lingual as possible, with extra emphasis on the Arabic language, it will be no surprise that it works fine for right-to-left languages (see figure 4).

An additional feature that is useful for Arabic users 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. In theory it is relatively simple to add additional language support, however, willing translators are not easy to come by! Further details of aConCorde are described in Section 4.

3.5 Discussion

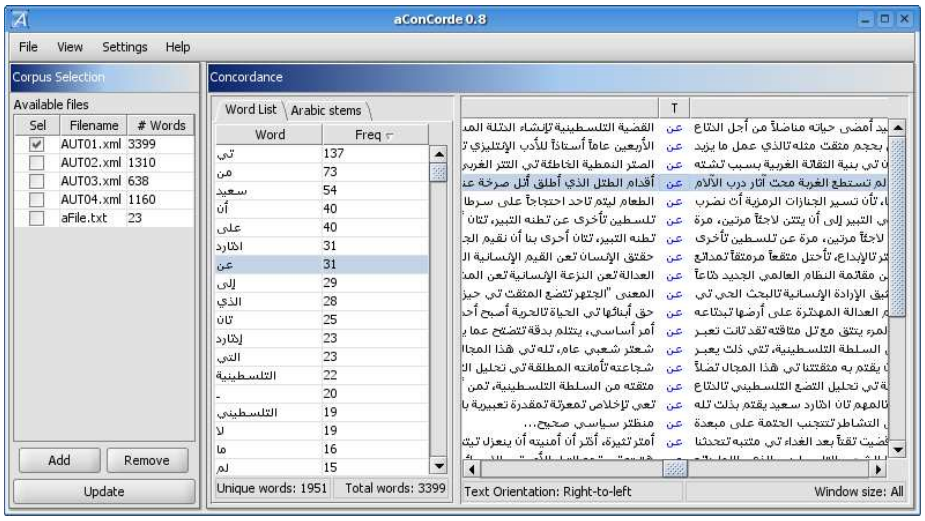
The aConCorde project was started solely because it had not been possible to find a concordancer to correctly work with Arabic. The Xaira software has been demonstrated to work with Arabic too, so it could be argued that there was no need for aConCorde in the first place. Unfortunately, Xaira was only discovered *after* development had already begun! Initially, it would seem that Xaira has now rendered aConCorde redundant, however, after pondering this very thought, it was realised that there is a market for both. Xaira is a sophisticated package, but with increased power and flexibility comes additional complexity. It certainly lends itself for research purposes in performing complex queries for fine-grained language analysis. However, it is questionable if it is quite as appropriate in the language teaching environment. Teachers and students may be put off by the complex interface and the amount of effort just to get a concordance output. aConCorde on the other hand is relatively simple to get up and running: literally select a corpus to open, then either type in your query



```
نال: ماذا تقول؟! إنها عظيمةوليست مجرد جيدة.. أدركت من استنكار جاري أنني هضمت مستوى اللقاء, وشعرت أنني قد
بيدي, فطلبت منه مزيدا من الورق, إذ كان يحمل رزمة من الأوراق ربما ليكتب لو لم يعمل المسجل, ولما انتهى اللقاء,
ا وصفها جاري, إذ يمتلك أسلوبا حواريا مقنعا. ثم نزلت من الطابق الأعلى لأجد الناس يصطفون لالتقاط الصور معه, و
جليزية لا يبارى, وأجاد الفرنسية, ثم عاد لبيروت وتمكن من العربية.
قدم رأيه وحجته, ثم وقت للرد, ثم يجيبان عن أسئلة قليلة من القاعة, والمناقشة في وقت محدد.. يضاء قرب نهاية كل سي
والفكرة, وبراعة المواجهة, فشجعان العقول قليل.. وهو من القلة التي تستطيع أن تكشف حدود الحرية الفكرية في أمريك
برنامج تشارلي روزتهاتفنا بالخبر؛ لأننا سنقضي ساعة من المتعة والفكرة, وبراعة المواجهة, فشجعان العقول قليل.. و
وبين شق آخر هو بقية اليسار, من المتمردين على ثقافة اليمين المتطرف الصهيوني, وقد قاد ال
لامذته دور مهم في تقديمه وتأليف الكتب عنه, وله تلميذة من الهند أصبحت تدرّس في الجامعة نفسها, كتبت عنه وجمعت
عظم اللقاء, وانتهت الأوراق التي بيدي, فطلبت منه مزيدا من الورق, إذ كان يحمل رزمة من الأوراق ربما ليكتب لو لم يع
كان شديد القسوة على من يسميه بالمثقف الخائن, ويكثر من تكرار استخدام أحد الكتب الفرنسية المثيرة التي كتبت مطولا
عرف على ظاهرة الاستشراق, وما يؤسس لها ويلحق بها من ثقافة, وكان لتلامذته دور مهم في تقديمه وتأليف الكتب عنه
لقى بداية تعليمه في القاهرة, ثم أكمل في أمريكا في أكثر من جامعة, وانتهى به الأمر أستاذا للأدب الإنجليزي في جامعة
ما أغرى المثقفين الإسرائيليين بدعوتها وتكريمها, تخفيفا من حملتها عليهم, أو حرفا لها عن طريقها, وإغاظة لأستاذها, و
ي لندن يقدم مقابلة أو محاورة عن المنطقة, وعن جوانب من حياته, وكانت امتلأت القاعة قبل بدء اللقاء, وأثناء المحاضر
ثم التقيت بمبشر قضى أكثر ردح من حياته في بلاد الشام يدأب لإنشاء الكنائس وتوجيه الشباب ال
```

```
For Help, press F1    AUT02 - CCA | null | 16:30(1) | AUT02 | 67
```

Figure 3: Example of Xaira producing correct concordance output.



Available files

Sel	Filename	# Words
✓	AUT01.xml	3399
	AUT02.xml	1310
	AUT03.xml	638
	AUT04.xml	1160
	aFile.txt	23

Word	Freq
تي	137
من	73
سعيد	54
أن	40
على	40
ادتارد	31
عن	31
إلى	29
الذي	28
تان	25
إدتارد	23
التي	23
التلسطينية	22
-	20
التلسطيني	19
لا	19
ما	16
لم	15

Unique words: 1951 Total words: 3399

```
يد أمضى حياته مناضلاً من أجل الدتاع عن القضية التلسطينية تإنشاء الدتلة المد
بحجم مثقت مثله تالذي عمل ما يزيد عن الأربعين عاماً أستاذاً للأدب الإنتليزي ت
ن تي بنية الثقاتة الغربية بسبب تشته عن الصتر النمطية الخاطئة تي التتر الغربي
لم تستطع الغربة محت آثار درب الآلام عن أقدام الطتل الذي أطلق أتل صرخة عن
ا، تأن تسير الجنازات الرمزية أث نضرب عن الطعام ليتم تاحد احتجاجاً على سرطا
ي التبير إلى أن يتتن لاجئاً مرتين، مزة عن تلسطين تأخرى عن تطنه التبير، تتان
لاجئاً مرتين، مزة عن تلسطين تأخرى عن تطنه التبير، تتان أحرى بنا أن نقيم الج
تر تالإبداع، تأحتل متقعاً مرمتقاً تمدائع عن حقتق الإنسان تعن القيم الإنسانية ال
ن مقاتمة النظام العالمي الجديد دتاعاً عن العدالة تعن النزعة الإنسانية تعن المد
تيق الإرادة الإنسانية تالبحث الحي تي عن المعنى "الجتهر تتضع المثقت تي حيز
م العدالة المهدترة على أرضها تبدتاعه عن حق أبنائها تي الحياة تالحرية أصبح أحد
لمرء يتتق مع تل متاقته تقد تانت تعبر عن أمر أساسي، يتتلم بدقة تتضح عما ي
السلطة التلسطينية، تتي ذلك يعبر عن شعتر شعبي عام، تله تي هذا المجا
ن يقتم به مثقتنا تي هذا المجال تضلاً عن شجاعته تأمانته المطلقة تي تحليل ال
ة تي تحليل التضع التلسطيني تالدتاع عن متقته من السلطة التلسطينية، تمن
تالمهم تان ادتارد سعيد يقتم بذلت تله تعي تإخلاص تمعرتة تمقدرة تعبيرية با
التشاطر تتتجنب الحتمة على مبعدة عن منظتر سياسي صحيح...
قضيت تقتاً بعد الغداء تي متتبه تتحدثنا عن أمتر تثيرة، أتتر أن أمنيته أن ينعزل تيت
```

```
Text Orientation: Right-to-left    Window size: All
```

Figure 4: Example of aConCorde displaying Arabic concordance correctly.

or select word from the word frequency panel and you will be presented with your results. This approach could well be more suited to this audience. aConCorde does provide special searches specific to Arabic analysis in the form of root/stem-based concordance and this is not found in any other package. Also, the aConCorde software is free to use by anyone (as is the source code).

With respect to the other tools in the survey, better support for right-to-left languages would obviously be greatly appreciated. If MonoConc had this support (as well as Unicode) then it would probably be the best all-round product in terms of the balance between features and ease-of-use, for both research and teaching purposes.

4 The aConCorde System

The aConCorde project was originally conceived as a prototype system to illustrate just how easy it is to not just write tools to support the Arabic language, but for practically any language. Within one afternoon, a simple multi-lingual concordance engine was programmed, and during the following afternoon, a basic graphical user interface was added. The user was able to switch between a native English or Arabic interface. All Arabic text was displayed correctly, and all concordance was output as expected by an Arabic reader, and there was no transliteration required for the input or output.

4.1 aConCorde features

Subsequent improvements and extensions have allowed the project to mature and has produced a more reliable and useful tool. The aConCorde system contains a number of features that allow it to stand out from competing products:

- Full Arabic support (no need to transliterate to Roman alphabet before concordance).
- English and Arabic native interfaces.
- Multi-platform — can run on most major operating systems.
- Supports an extensive range of character encodings, including Unicode (UTF-16 and UTF-8), Windows Arabic (CP1256), IBM Arabic (CP420), MacArabic, ISO Latin/Arabic (ISO 8859-6) and ASCII encoding.
- Multi-format support allows you to load text files, XML files, HTML files, RTF files and MS Word files directly.
- Can save results either as a plain text file, or HTML file that can keep the alignment of the concordance.
- Word frequency analysis.
- Concordance can be sorted on left or right contexts.
- A range of query options: key-word, phrase, proximity, boolean, wildcard and Arabic root/stem queries.

- aConCorde is freeware and open-source (released under the General Public License.)

The aConCorde system is built using the Java programming language. Java allows the programmer to produce software for the most popular operating systems without any extra effort, which is why aConCorde will run correctly on Microsoft Windows, Linux or Mac OS (and others) providing the user has the Java Runtime Environment installed. It is important to note that many Arabic tools were reliant on Arabic Windows (a localised version of Microsoft Windows with support for Arabic script and right-to-left interfaces). aConCorde will of course run on standard Windows or Arabic Windows.

The Java platform is also one of the reasons why producing a multi-lingual capable application was simple. Internationalisation features were an important feature during the design of Java. For example, Java's internal mechanism to store text uses the Unicode standard, and all visual components have inbuilt support for left-to-right or right-to-left languages (some components can even cope with the Japanese top-to-bottom text orientation!)

4.2 Root- and stem-based concordance

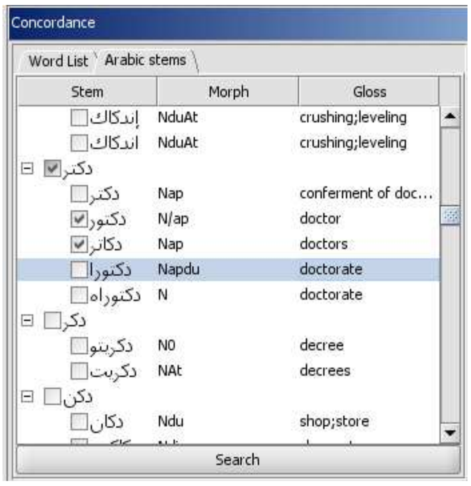
As discussed in Section 2 the wildcard searches that make it reasonably simple for stem-based concordance in Western languages do not scale universally. To recap, an Arabic root is typically a sequence of 3 consonants (although two, four and five consonant roots exist) that is the seed from which words are derived. For example the root *ktb* (write) has stems derived such as *katab* (write) and *iktatab* (register). From the stem you can derive the various morphological forms using affixes, infixes and suffixes, e.g., *Taktub* (she writes), *yaktub* (he writes), *aktub* (I write), and so on.

4.2.1 Buckwalter's morphological analyser

Buckwalter's morphological analyser is distributed by the LDC and has been used by many projects including the LDC's Arabic Treebank project (Maamouri *et al.*, 2004). It consists of 3 lexicons: affixes, suffixes and stems) and 3 sets of rules that specify how these lexicons can be intersected to derive actual Arabic word tokens. The analyser reads in a file and processes each word in isolation. By utilising the lexicons and rules, it can break down the input token and produce all possible valid morphological solutions. In the cases where there are more than one solution, it doesn't offer any hints as to the most probable.

The databases and the analyser work with Buckwalter's transliteration system. Until recently, direct input/output of Arabic characters in computer systems has not been trivial. Buckwalter's system is a one-to-one mapping between the Roman alphabet and the Arabic alphabet. It's not always intuitive because there isn't a nice mapping available, and so it uses some punctuation symbols to represent Arabic letters (e.g., & is equivalent to Waw with Hamza above). Conveniently, there is also a one-to-one mapping to the Unicode character encoding standard.

There are many transliteration systems in use although there is no single official transliteration standard. Yet, Buckwalter transliteration is not recognised within the Arabic linguistic community at large (Beesley, 2003). This is



Stem	Morph	Gloss
☐ إندكاك	NduAt	crushing;leveling
☐ اندكاك	NduAt	crushing;leveling
⊟ ✓ دكتر		
☐ دكتر	Nap	conferment of doc...
✓ دكتور	N/ap	doctor
✓ دكاتر	Nap	doctors
☐ دكتورا	Napdu	doctorate
☐ دكتوراه	N	doctorate
⊟ ☐ دكر		
☐ دكريتو	N0	decree
☐ دكريت	NAt	decrees
⊟ ☐ دكن		
☐ دكان	Ndu	shop;store

Figure 5: Example of root/stem selection within aConCorde using Buckwalter's stem database.

due to it being relatively modern, and also being aimed at the computational processing of Arabic. Its use of Latin punctuation symbols for Arabic letters is clearly less intuitive to a human reader than some of the more phonetic-based transliteration systems. For example, Buckwalter uses '$' for Sheen, whereas others, such as Qalam (Heddaya, 1985), use 'sh'.

4.2.2 Buckwalter and aConCorde

If a user wished to find concordance for all words derived from the stem *katab*, then this is clearly not feasible with the wildcard search approach. The alternative is to search for a manually hand-crafted list of derived words. However, with aConCorde, this burden is taken away from the user. Using the databases of Buckwalter's morphological analyser, it's possible to provide the roots and stems *a priori* to the user. They can then select to search for one or more stems, or even everything within a given root. Buckwalter's databases were converted from Buckwalter's transliterion alphabet to Unicode so that the root/stems are displayed in native Arabic (see Figure 5).

There is a limitation here in that the user is always presented with the full database, even if many of the terms are not present within the corpus being currently analysed. It would of course be preferable, when loading an Arabic corpus to analyse each word and determine its stem and root, and then only have these entries visible. However, stemming in Arabic is difficult due to the complex morphology. It is quite easy to remove affixes that were in fact part of the word, for example. That being said, there are stemmers that exist, such as Khoja's stemmer used in her APT tagger Khoja (2003).

4.3 Similar terms and word clusters

A novel feature to aConCorde is the use of similarity functions. These are borrowed from the information retrieval domain. They are typically used to find similar documents based on the terms contained within. In aConCorde, its low-level model, from an IR perspective, has sentences as 'documents'. The concordancer has a set of term-frequency vectors and these can be plotted into a vector space. During the concordance, the user can choose their search term, select a sample line from the concordance and its term-vector will be compared to the others (using cosine similarity) and then be presented with similar sentences from within the corpus.

4.3.1 Clustering

The principle of clustering algorithms is to divide a set of objects into clusters. By using a given measure of similarity, a good clustering algorithm will place objects that are similar into the same cluster, whereas dissimilar ones are clustered into different groups. It has the advantage of being truly unsupervised, i.e., requires no prior knowledge about the data being clustered. Clustering is a broad subject and has been applied in many domains. Within computational linguistics, it has been used successfully in syntactic (Finch and Chater, 1992; Hughes, 1994) and semantic (Ibrahimov *et al.*, 2001) classification and information retrieval systems.

Clustering is a mathematical procedure that merges points in a vector space that are close to each other. To cluster words, a method is required to represent a given word into a vector to be plotted. The method used by Roberts (2002) is to record collocation frequencies of a given content word relative to function words, for a specified window size. This data can be converted simply to a vector and thus ready to be clustered.

There are a range of clustering algorithms available. Hierarchical agglomerative algorithms are well suited to word clustering. Each word in the vector space start off as individual one-object clusters. The algorithm evaluates each pair of points to find the closest, which are deemed the most similar, and merges them to a single cluster. How algorithms compute the distances between clusters is typically the differentiating factor (Everitt, 1993). Group Average and Ward's Method are good performers in this task (Roberts, 2002; Zupan, 1982). The results come back as a set of clusters each with a varying number of words within. A nice feature of hierarchical methods is that you can profile the clustering process and generate a dendogram that reveals how the cluster was formed, e.g., which terms were the most similar.

4.3.2 Clustering and aConCorde

aConCorde provides a means for the user to interface external clustering code and display the output within the concordancer itself. This has only been introduced recently and is still very crude. However, it is worth mentioning as it shows yet more potential features that could be added to concordance software to enable alternative perspectives and analyses of a corpus.

The term similarity and clustering features were implemented primarily for the use of concordance within the learning environment. Concordance has

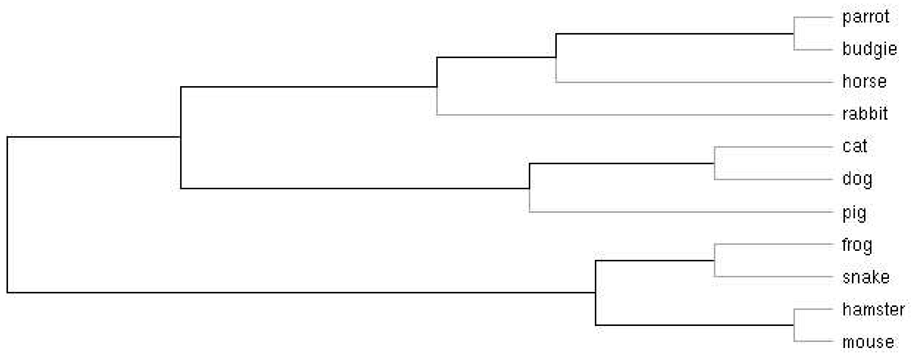


Figure 6: An example of a cluster visualised as a dendogram.

```
aConCorde_0.8-2.png not found!
```

Figure 7: Cluster view within aConCorde.

already proved successful within the classroom, but this concordancer can provide additional information to the user to assist in building vocabularies and grammatical patterns.

4.4 aConCorde limitations

At the time of writing, aConCorde is still limited in terms of features compared to the more established products on the market.

- Mark-up: aConCorde is generally ignorant of mark-up annotation within a corpus. Whilst it can successfully parse XML and HTML files, it essentially works by filtering out the tags to isolate the content. If a corpus were annotated with part-of-speech tags aConCorde, would not recognise them and would be treated as 'words', as if part of the text itself. Heavily annotated corpora will therefore benefit from some pre-processing to strip away such mark-up.
- Full context: aConCorde doesn't have the functionality to allow the user to see the full context of a selected concordance item. The current display is to see the word within the sentence it was found, and that is the largest scope currently implemented.
- Arabic root/stem database is exhaustive rather than reflecting the tokens available in the currently loaded corpus.
- Clustering integration is currently very crude and takes a long time to gather and display the data.
- Computational resources: aConCorde can be resource intensive with large datasets. Whilst the software utilises extremely scalable indexing technologies like those found in modern databases, loading corpora or gathering the concordance for highly frequent words take time.

Naturally, aConCorde will continue to progress. These limitations are high priority issues and will be addressed in the near future.

5 Conclusion

This paper has summarised many of the issues regarding the difficulty of Arabic concordance. The aConCorde project attempts to resolve some of the core issues, e.g., interactive searching that actually displays the Arabic text correctly. With that foundation firmly laid, the future direction can focus on important problems, for example, improving and creating more novel approaches to intuitive morphological searching. It is hoped that the established concordance tools will realise the demand for products that cope with Arabic script, and adapt their software accordingly.

References

Al-Sulaiti, Latifa (2004) *Designing and Developing a Corpus of Contemporary Arabic*. Master's thesis, School of Computing, University of Leeds, UK.

Beesley, Kenneth (2003) *Xerox Arabic Morphological Analyzer Surface-Language (Unicode) Documentation*. Xerox Research Centre Europe.

Bernard, Lou (2004) BNC-Baby and Xaira. In *Proceedings of the Sixth Teaching and Langauge Corpora conference*, p. 84, Granada.

Bernard, Lou and Dodd, Tony (2003) Xara: an XML aware tool for corpus ssearching. In *Proceedings of the Corpus Linguistics 2003 Conference*, Dawn Archer, Paul Rayson, Andrew Wilson and Tony McEnery, eds., volume 16, pp. 142–144, UCREL, University of Lancaster.

Boualem, Malek, Leisher, Mark and Ogden, Bill (1999) Concordancer for Arabic. In *Arabic Translation and Localisation Symposium*, Tunis, URL `http://crl.nmsu.edu/~mleisher/concord.pdf`.

Buckwalter, Tim (2002) Buckwalter Arabic transliteration. URL `http://www.qamus.org/transliteration.htm`.

Cobb, Tom, Greaves, Chris and Horst, Marlise (2001) Can the rate of lexical acquisition from reading be increased? an experiment in reading french with a suite of on-line resources. In *Regards sur la didactique des langues secondes*, P. Raymond and C. Cornaire, eds., Éditions Logique, Montréal.

de Roeck, Anne (2002) Arabic for the absolute beginner. *ELRA Newsletter*, **7**(1).

Dodd, Bill (1997) Exploiting a corpus of written german for advanced language learning. In *Teaching and Language Corpora*, Anne Wichmann, Steven Fligelstone, Gerry Knowles and Tony McEnery, eds., pp. 131–145, Longman, London.

Everitt, B. (1993) *Cluster Analysis*. Edward Arnold, London, 3rd edition.

Finch, Steve and Chater, Nick (1992) Bootstrapping syntactic categories. In *Proceedings of the 14th Annual Meeting of the Cognitive Science Society*, pp. 820–825, Hillsdale, New Jersey.

Heddaya, Abdelsalam (1985) Qalam: A convention for morphological arabic-latin-arabic transliteration. URL `http://eserver.org/langs/qalam.txt`.

Hoogland, Jan (2003) The Nijmegen Arabic/Dutch dictionary project — using the concordance program. URL `http://www.let.kun.nl/wba/Content2/1.4.6_Concordancing.htm`.

Hughes, John (1994) *Automatically Acquiring a Classification of Words*. Ph.D. thesis, School of Computing, University of Leeds.

Ibrahimov, O, Sethi, I and Dimitrova, N (2001) Clustering of imperfect transcripts using a novel similarity measure. In *Proceedings of the SIGIR'01 Workshop on Information Retrieval Techniques for Speech Applications*.

Johns, Tim (1990) 'From printout to handout: Grammar and vocabulary teaching in the context of data-driven learning'. *Computer Assisted Language Learning*, **10**, pp. 14–34.

Khoja, Shereen (2003) *APT: An Automatic Arabic Part-of-Speech Tagger*. Ph.D. thesis, Computing Department, Lancaster University, UK.

Lawler, John (2000) Review of MonoConc Pro 2.0 concordancing software. *LINGUIST*, **11**(1411).

Maamouri, Mohamed, Bies, Ann, Buckwalter, Tim and Mekki, Wigdan (2004) The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Ogden, Bill and Bernick, Philip (1996) Oleda: User-centered Tipster technology for language instruction. In *Proceedings of the Tipster Pharse II 24 Month Workshop*, VA, USA.

Roberts, Andrew (2002) Automatic acquisition of word classification using distributional analysis of content words with respect to function words. Technical report, School of Computing, University of Leeds.

Scott, Mike (2004) *WordSmith Tools 4.0*. URL `http://www.lexically.net/downloads/version4/html/index.html`.

Zupan, Jure (1982) *Clustering of Large Data Sets*. John Wiley and Sons, Chichester.