Conference paper

Sawalha, M. and Atwell, E.S. (2008) *Comparative evaluation of Arabic language morphological analysers and stemmers.* In: Proceedings of COLING 2008 22nd International Conference on Comptational Linguistics (Poster Volume)). COLING 2008, 18 Aug 2008 - 22 Aug 2008, Manchester. Coling 2008 Organizing Committee , Manchester, 107 - 110 .

Comparative Evaluation of Arabic Language Morphological Analysers and Stemmers

Majdi Sawalha
School of Computing,
University of Leeds,
Leeds LS2 9JT, UK.

sawalha@comp.leeds.ac.uk

Eric Atwell
School of Computing,
University of Leeds,
Leeds LS2 9JT, UK.

eric@comp.leeds.ac.uk

Abstract

Arabic morphological analysers and stemming algorithms have become a popular area of research. Many computational linguists have designed and developed algorithms to solve the problem of morphology and stemming. Each researcher proposed his own gold standard, testing methodology and accuracy measurements to test and compute the accuracy of his algorithm. Therefore, we cannot make comparisons between these algorithms. In this paper we have accomplished two tasks. First, we proposed four different fair and precise accuracy measurements and two 1000-word gold standards taken from the Holy Qur'an and from the Corpus of Contemporary Arabic. Second, we combined the results from the morphological analysers and stemming algorithms by voting after running them on the sample documents. The evaluation of the algorithms shows that Arabic morphology is still a challenge.

1   Three Stemming Algorithms

We selected three stemming algorithms for which we had ready access to the implementation and/or results.

Shereen Khoja Stemmer : We obtained a Java version of Shereen Khoja's stemmer (Khoja,1999). Khoja's stemmer removes the longest suffix and the longest prefix. It then matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words (Larkey & Connell 2001).

Tim Buckwalter Morphological analyzer: Tim Buckwalter developed a morphological analyzer for Arabic. Buckwalter compiled a single lexicon of all prefixes and a corresponding unified lexicon for suffixes instead of compiling numerous lexicons of prefixes and suffix morphemes. He included short vowels and diacritics in the lexicons[1].

Tri-literal Root Extraction Algorithm : Al-Shalabi, Kanaan and Al-Serhan developed a root extraction algorithm which does not use any dictionary. It depends on assigning weights for a word's letters multiplied by the letter's position, Consonants were assigned a weight of zero and different weights were assigned to the letters grouped in the word "سألتمونيها" where all affixes are formed by combinations of these letters. The algorithm selects the letters with the lowest weights as root letters (Al-Shalabi et al, 2003).

2   Our Approach: Reuse Others' Work

The reuse of existing components is an established principle in software engineering. We procured results from several candidate systems, and then developed a program to allow "voting" on the analysis of each word: for each word, examine the set of candidate analyses. Where all systems were in agreement, the common analysis is copied; but where contributing systems disagree on the analysis; take the "majority vote", the analysis given by most systems. If there is a tie, take the result produced by the system with the highest accuracy (Atwell & Roberts, 2007).

3   Experiments and Results

Experiments are done by executing the three stemming algorithms, discussed above, on a ran-

---

[1] Tim Buckwalter web site: http://www.qamus.org

domly selected chapter number 29 of the Qur'an **"Souraht Al-Ankaboot" "The Spider"** in Eng lish see figure 1; and a newspaper text taken from the Corpus of Contemporary Arabic developed at the University of Leeds, UK. We selected the test document from the politics, sports and economics section, taken from newspaper articles, see figure 2 (Al-Sulaiti & Atwell, 2006). Each test document contains about 1000 words. We manually extracted the roots of the test documents' words to compare results from different stemming systems. Roots extracted have been checked by Arabic Language scholars who are experts in the Arabic Language.

Table 1 shows a detailed analysis been done for the sample test documents, the Qur'an corpus as one unit, and a daily newspaper of contemporary Arabic test document, taken from Al-Rai

daily newspaper published in Jordan. The analysis also shows that function words such as **"في" "fi" "in", "من" "min" "from", "على" "Ala" "on"** and **"الله" "Allah" "GOD"** are the most frequent words in any Arabic text. On the other hand, non functional words with high frequency such as **"الجامعات" "Al-Jami'at" "Universities"** and **"الكويت" "Al-Kuwait" "Kuwait"** gives a general idea about the main topic of the article.

Simple tokenization is applied for the text of the gold standard documents. This will ensure that test documents can be used to test any stemming algorithm smoothly and correctly.

4 Four Accuracy measurements

In order to fairly compare between different stemming algorithms we applied four different

الم أَحَسِبَ النَّاسُ أَن يُتْرَكُوا أَن يَقُولُوا آمَنَّا وَهُمْ لَا يُفْتَنُونَ وَلَقَدْ فَتَنَّا الَّذِينَ مِن قَبْلِهِمْ فَلَيَعْلَمَنَّ اللَّهُ الَّذِينَ صَدَقُوا وَلَيَعْلَمَنَّ الْكَاذِبِينَ أَمْ حَسِبَ الَّذِينَ يَعْمَلُونَ السَّيِّئَاتِ أَن يَسْبِقُونَا سَاء مَا يَحْكُمُونَ مَن كَانَ يَرْجُو لِقَاء اللَّهِ فَإِنَّ أَجَلَ اللَّهِ لآتٍ وَهُوَ السَّمِيعُ الْعَلِيمُ وَمَن جَاهَدَ فَإِنَّمَا يُجَاهِدُ لِنَفْسِهِ إِنَّ اللَّهَ لَغَنِيٌّ عَنِ الْعَالَمِينَ وَالَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ لَنُكَفِّرَنَّ عَنْهُمْ سَيِّئَاتِهِمْ وَلَنَجْزِيَنَّهُمْ أَحْسَنَ الَّذِي كَانُوا يَعْمَلُونَ وَوَصَّيْنَا الْإِنسَانَ بِوَالِدَيْهِ حُسْنًا وَإِن جَاهَدَاكَ لِتُشْرِكَ بِي مَا لَيْسَ لَكَ بِهِ عِلْمٌ فَلَا تُطِعْهُمَا إِلَيَّ مَرْجِعُكُمْ فَأُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ وَالَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ لَنُدْخِلَنَّهُمْ فِي الصَّالِحِينَ

Figure 1: Sample from Gold Standard first document taken from Chapter 29 of the Qur'an.

ستبقى العولمة وإلى وقت ممتد مثيرة للأسئلة والأجوبة وفي هذا المقال وقفة تأمل عميقة في بعض هذه الأسئلة بدأت منذ فترة موجة جديدة من الكتابات تروج للعولمة باعتبارها الشكل الجديد لحياة البشر في ظل القطب الأمريكي وهناك نمط من هذه الكتابات يروج للنمط الأمريكي متعدد الأعراق والثقافات بوصفه النمط الأمثل للحياة في القرية الكونية الجديدة التي قاربت وسائل الاتصالات والمواصلات ونظم المعلومات ووسائل الإعلام بين أجزائه المختلفة ويبشر أصحاب هذه النظرة ببشر من نوع جديد بشر كوزموبوليتان

Figure 2: Sample from Gold Standard document taken from the Corpus of Contemporary Arabic.

Table 1: Summary of detailed analysis.

	Qur'an Corpus		Gold Standard First Document Chapter 29 of the Qur'an		Gold Standard Second Document "Corpus of Contemporary Arabic"		Al-Rai daily Newspaper Test Document	
Total number of Tokens	77,789		987		1005		977	
Word Types	19,278		616		710		678	
Top 10 Tokens	Token	Freq.	Token	Freq.	Token	Freq.	Token	Freq.
1	في	1179	في	21	في	35	في	39
2	مِن	872	اللَّهِ	17	من	21	من	16
3	مَا	832	مِن	14	على	12	على	13
4	الَّذِينَ	808	اللَّهُ	12	التي	12	التي	10
5	عَلَى	652	وَمَا	12	الكويت	11	إلى	9
6	وَمَا	640	إِلَّا	12	أن	10	المبنى	8
7	إِنَّ	605	الَّذِينَ	11	هذه	10	الجامعات	8
8	اللَّهِ	464	مَا	8	إلى	8	أن	7
9	أن	499	اللَّهَ	8	امام	8	السلام	7
10	قَالَ	416	كَانُوا	8	عن	7	جلالته	7

accuracy measurements. Each time we ran the experiment, a comparison of the results with the gold standard was performed.

The first experiment was done by comparing each root extracted using the three stemming algorithms with the roots of words in the gold standard.

The second experiment excludes from the words' list stop words. The third experiment compares all word-type roots to the gold standard's roots. Finally, word-type roots excluding the stop words are compared to the gold standard's roots. Tables 4-7 show the accuracy rates resulting from the four different accuracy measurements.

Table 2: Tokens Accuracy of stemming algorithms after testing on Qur'an gold standard

Number of Tokens including Stop words (978 tokens)			
Stemming Algorithm	Errors	Fault Rate	Accuracy
Khoja stemmer	311	31.8%	68.2%
Tim Buckwalter morph. Analyzer	419	42.8%	57.16%
Tri-literal Root algorithm	394	40.3%	59.71%
Voting algorithm Ex.1	434	44.4%	55.6%
Voting algorithm Ex.2	405	41.4%	58.6%
Number of Tokens excluding Stop words (554 tokens)			
Khoja stemmer	209	37.73%	62.27%
Tim Buckwalter morph. Analyzer	325	58.66%	41.34%
Tri-literal Root algorithm	279	50.36%	49.64%
Voting algorithm Ex.1	266	48.0%	52.0%
Voting algorithm Ex.2	229	41.3%	58.7%

Table 3: Word type Accuracy of stemming algorithms after testing on Qur'an gold standard

Number of Word Types including Stop words (616 word types)			
Stemming Algorithm	Errors	Fault Rate	Accuracy
Khoja stemmer	224	36.36%	63.64%
Tim Buckwalter morph. Analyzer	267	43.34%	56.66%
Tri-literal Root algorithm	266	43.18%	56.82%
Voting algorithm Ex.1	242	39.3%	60.7%
Voting algorithm Ex.2	219	35.6%	64.4%
Number of Word types excluding Stop words (451 word types)			
Khoja stemmer	155	34.37%	65.63%
Tim Buckwalter morph. Analyzer	251	55.65%	44.34%
Tri-literal Root algorithm	214	47.45%	52.55%
Voting algorithm Ex.1	174	38.6%	61.4%
Voting algorithm Ex.2	151	33.5%	66.5%

Table 4: Token Accuracy of stemming algorithms. Tested on newspaper gold standard

Number of Tokens including Stop words(1005 tokens)			
Stemming Algorithm	Errors	Fault Rate	Accuracy
Khoja stemmer	231	22.99%	77.01%
Tim Buckwalter morph. Analyzer	596	59.30%	40.70%
Tri-literal Root algorithm	234	23.28%	76.72%
Voting algorithm Ex.1	303	30.15%	69.85%
Voting algorithm Ex.2	266	26.47%	73.53%
Number of Tokens excluding Stop words (766 tokens)			
Khoja stemmer	212	27.7%	72.3%
Tim Buckwalter morph. Analyzer	431	60.70%	39.30%
Tri-literal Root algorithm	253	35.63%	64.37%
Voting algorithm Ex.1	303	39.56%	60.44%
Voting algorithm Ex.2	266	34.73%	65.27%

Table 5: Word type Accuracy of stemming algorithms. Tested on newspaper gold standard

Number of Word Types including Stop words (710 word types)			
Stemming Algorithm	Errors	Fault Rate	Accuracy
Khoja stemmer	232	32.68%	67.32%
Tim Buckwalter morph. Analyzer	431	60.70%	39.30%
Tri-literal Root algorithm	253	35.63%	64.37%
Voting algorithm Ex.1	248	34.93%	65.07%
Voting algorithm Ex.2	215	30.28%	69.71%
Number of Word types excluding Stop words (640 word types)			
Khoja stemmer	184	28.75%	71.25%
Tim Buckwalter morph. Analyzer	423	66.09%	33.91%
Tri-literal Root algorithm	224	35.00%	65.00%
Voting algorithm Ex.1	252	39.4%	60.6%
Voting algorithm Ex.2	195	30.5%	69.5%

Experiments are done for results generated from the three stemming algorithms after executing them on both gold standard documents.

The output analysis of the stemming algorithms is considered as input for the "voting" program. The program reads in these files, tokenizes them, and stores the words and the roots extracted by each stemming algorithm in temporary lists to be used by the voting procedures.

The temporary lists work as a bag of words that contains all the result analysis of the stemming algorithms. Khoja and the tri-literal stemming algorithms generate only one result analysis for each input word, while Tim Buckwalter morphological analyzer generates one or more result analysis. These roots are ranked in best-first order according to accuracy measurement done before. Khoja stemmer results are inserted to the list first then the results from tri-literal stemming algorithm and finally the results of Tim Buckwalter morphological analyzer.

After the construction of the lists of all words and their roots, a majority voting procedure is applied to it to select the most common root among the list. If the systems disagree on the analysis, the voting algorithm selects "Majority Vote" root as the root of the word. If there is a tie, where each stemming algorithm generates a different root analysis then the voting algorithm selects the root by two ways. Firstly, it simply selects the root randomly from the list using the `FreqDist()` Python function in experiment 1. Secondly, In experiment 2, the algorithm selects the root generated from the highest accuracy stemming algorithm which is simply placed in the first position of the list as the root of the word are inserted to the list using the best-first in terms of accuracy strategy.

After the voting algorithm, the selected root is compared to the gold standard. Tables 2-5 show the result of the voting algorithm which achieves promising accuracy results of slightly better than the best stemming algorithm in experiment 2 and a similar accuracy rates for the best stemming algorithms in experiment 1.

5   Conclusions

In this paper, we compared between three stemming algorithms; Shereen Khoja's stemmer, Tim Buckwalter's morphological analyzer and the Tri-literal root extraction algorithm.

Results of the stemming algorithms are compared with the gold standard using four different accuracy measurements. The four accuracy measurements show the same accuracy rank for the stemming algorithms: the Khoja stemmer achieves the highest accuracy then the tri-literal root extraction algorithm and finally the Buckwalter morphological analyzer.

The voting algorithm achieves about 62% average accuracy rate for Qur'an text and about 70% average accuracy for newspaper text. The results show that the stemming algorithms used in the experiments work better on newspaper text than Quran text, not unexpectedly as they were originally designed for stemming newspaper text.

All stemming algorithms involved in the experiments agreed and generate correct analysis for simple roots that do not require detailed analysis. So, more detailed analysis and enhancements are recommended as future work.

Most stemming algorithms are designed for information retrieval systems where accuracy of the stemmers is not important issue. On the other hand, accuracy is vital for natural language processing. The accuracy rates show that the best algorithm failed to achieve accuracy rate of more than 75%. This proves that more research is required. We can not rely on such stemming algorithms for doing further research as Part-of-Speech tagging and then Parsing because errors from the stemming algorithms will propagate to such systems.

Our experiments are limited to the three stemming algorithms. Other algorithms are not available freely on the web, and we have been unable so far to acquire them from the authors. We hope Arabic NLP researchers can cooperate further in open-source development of resources.

References

Al-Shalabi, R., Kanaan, G., & Al-Serhan, H. (2003, December). *New approach for extracting Arabic roots*. Paper presented at the International Arab Conference on Information Technology (ACIT'2003), Egypt.

Al-Sulaiti, Latifa; Atwell, Eric 2006. *The design of a corpus of contemporary Arabic*. International Journal of Corpus Linguistics, vol. 11, pp. 135-171. 2006.

Atwell, Eric and Roberts, Andy, 2007. *CHEAT: combinatory hybrid elementary analysis of text* in: Proceedings of Corpus Linguistics 2007.

Khoja, Shereen, 1999. Stemming Arabic Text. http://zeus.cs.pacificu.edu/shereen/research.htm

Larkey Leah. S. and Connell Margrate. E. 2001. *Arabic information retrieval at UMass*. In Proceedings of TREC 2001, Gaithersburg: NIST, 2001.