Identifying Broken Plurals in Unvowelised Arabic Text

Abduelbaset Goweder
University of Essex
Dept. of Computer Science
Wivenhoe Park,
Colchester CO4 3SQ, UK
agowed@essex.ac.uk

Massimo Poesio
University of Essex
Dept. of Computer Science
Wivenhoe Park,
Colchester CO4 3SQ, UK
poesio@essex.ac.uk

Anne De Roeck
The Open University
Dept. of Computing
Walton Hall, Milton Keynes
Buckinghamshire, MK7 6AA, UK
A.DeRoeck@open.ac.uk

Jeff Reynolds
University of Essex
Dept. of Computer Science
Wivenhoe Park,
Colchester CO4 3SQ, UK
reynt@essex.ac.uk

Abstract

Irregular (so-called ***broken***) plural identification in modern standard Arabic is a problematic issue for information retrieval (IR) and language engineering applications, but their effect on the performance of IR has never been examined. Broken plurals (BPs) are formed by altering the singular (as in English: tooth → teeth) through an application of interdigitating patterns on stems, and singular words cannot be recovered by standard affix stripping stemming techniques. We developed several methods for BP detection, and evaluated them using an unseen test set. We incorporated the BP detection component into a new light-stemming algorithm that conflates both regular and broken plurals with their singular forms. We also evaluated the new light-stemming algorithm within the context of information retrieval, comparing its performance with other stemming algorithms.

1. Introduction

Broken plurals constitute ~10% of texts in large Arabic corpora (Goweder and De Roeck, 2001), and ~41% of plurals (Boudelaa and Gaskell, 2002). Detecting broken plurals is therefore an important issue for light-stemming algorithms developed for applications such as information retrieval, yet the effect of broken plural identification on the performance of information retrieval systems has not been examined. We present several methods for BP detection, and evaluate them using an unseen test set containing 187,309 words. We also developed a new light-stemming algorithm incorporating a BP recognition component, and evaluated it within an information retrieval context, comparing its performance with other stemming algorithms.

We give a brief overview of Arabic in Section 2. Several approaches to BP detection are discussed in Section 3, and their evaluation in Section 4. In Section 5, we present an improved light stemmer and its evaluation. Finally in Section 6, our conclusions are summarised.

2. Arabic Morphology and its Number System

Arabic is a heavily inflected language. Its grammatical system is traditionally described in terms of a root-and-pattern structure, with about 10,000 roots (Ali, 1988). Roots such as *drs* (درس) and *ktb* (كتب) are listed alphabetically in standard Arabic dictionaries like the Wehr-Cowan (Beesley, 1996). The root is the most basic verb form. Roots are categorized into: triliteral, quadriliteral, or rarely pentaliteral. Most words are derived from a finite set of roots formed by adding *diacritics*[1] or affixes (prefixes, suffixes, and infixes) through an application of fixed *patterns* which are templates to help in deriving inflectional and derivational forms of a word. Theoretically, several hundreds of Arabic words can be derived from a single root. Traditional Arab grammarians describe Arabic morphology in terms of patterns associated with the basic root *f3l* (فعل, “to do”)- where *f*, *3*, and *l* are like wildcards in regular expressions: the letter *f* (ف ,“pronounced fa”) represents the first consonant (sometimes called a radical), the letter *3* (ع , “pronounced ain”) represents the second, and the letter *l* (ل , “pronounced lam”) represents the third

[1] Special characters which are superscript or subscript marks added to the word.

respectively. Adding affixes to the basic root *f3l* (فعل, “to do”) allows additional such patterns to be formed. For instance, adding the letter *Alef* (ا) as a prefix to the basic root *f3l* (فعل, “to do”) we get the pattern *Af3l* (افعل) which is used to form words such as: *anhr* (انهر, “rivers”), *arjl* (ارجل, “legs”), and *asqf* (اسقف, “ceilings”). Some examples of the word patterns are *Yf3l* (يفعل), *Mf3Wl* (مفعول), *Af3Al* (أفعال), *MfA3l* (مفاعل), etc.

The following example shows how we can use patterns to form words. the verb *yktb* (يكتب , “he writes or he is writing”) is formed by mapping the consonants of the triliteral root *ktb* (ك ت ب) to the pattern *yf3l* (يفعل), where the letters *f* (ف), *3* (ع), and *l* (ل) in the second, third, and fourth positions of the pattern respectively represent slots for a root consonant. Figure 1 depicts the process of matching the root *ktb* (ك ت ب) to the pattern *yf3l* (يفعل) to produce the verb *yktb* (يكتب , “he writes or he is writing”), then adding prefixes and/or suffixes to obtain a word.

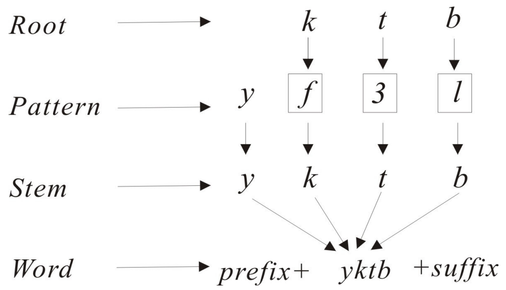

Figure 1: The process of mapping the root *ktb* (ك ت ب) to the pattern *yf3l* (يفعل).

The Arabic number system has singular, dual, and plural. Plurals are traditionally distinguished into two categories: the ***regular*** (so-called ***sound***) plurals, and the ***irregular*** (so-called ***broken***) plurals. Sound Plurals are formed by appropriate suffixation (like English: hand → hands). The sound masculine plural is formed by adding the suffix *oun* (ون) in the nominative case and the suffix *een* (ين) in the accusative & genitive cases. The sound feminine plural is formed by attaching the suffix *at* (ات) to the singular.

Irregular or broken plurals apply mostly to triliteral roots and are formed by altering the singular (as in English: tooth → teeth). Many nouns and adjectives have broken plurals (Haywood and Nahmad, 1976). In all cases, singulars are affected by applying a number of different patterns that alter long vowels (*Alef* (ا), *Waw* (و), *Yeh* (ي), and *Alef-Maqsura* (ى)), within or outside the framework of the consonants (Cowan, 1958). Table 1 gives some examples of BPs and their patterns.

Table 1: Broken Plural examples.

Singular	BP Pattern	Plural
qlm(قلم, “pen”)	*Af3Al* (افعال)	*AqlAm*(اقلام, “pens”)
qlb(قلب, “heart”)	*f3Wl* (فعول)	*qlWb*(قلوب, “hearts”)
ktab(كتاب, “book”)	*f3l* (فعل)	*ktb*(كتب, “books”)

The complexity of Arabic morphology has motivated a great deal of studies. Some of which especially concerned with broken plurals (McCarthy and Prince, 1990b; Kiraz, 1996a; Idrissi, 1997). These are successful to varying degrees, but have a main practical drawback in the context of information retrieval: they assume that words are fully vowelised. Unfortunately, short vowels are usually not written in published Arabic text, with the exception of the religious texts (e.g., the Holy Quran), poetry, and books for school children (Abuleil and Evens, 1998).

3. Different Approaches to BP Identification

We tested several different approaches for BP identification: simple BP matching, restricted BP matching (hand restricted & decision tree restricted), and a dictionary approach.

3.1 The Simple BP Matching Approach

Given the characterisation of broken plurals one finds in standard Arabic grammars, the most obvious method for identifying a broken plural is to light stem it (strip off any prefixes and/or suffixes), then trying to match the obtained stem against BP patterns found in standard grammars. Since this method is widely used, we adopted it as a baseline.

As a first step towards a simple BP matching algorithm, we developed a basic light stemmer for Arabic by modifying an existing root stemmer (Khoja and Garside, 1999). This basic light stemmer was incorporated in a simple BP identification module based on matching, using a list of 39 BP patterns found in grammar books such as Haywood and Nahmad (1976) and Cowan (1958). The simple BP matching algorithm takes in a word; light-stems it to produce morphological information such as stem, prefix and suffix; and returns TRUE if the stem matches one of the BP patterns in the list. The stem matches a BP pattern if and only if they have the same number of letters and the same letters in the same positions, excluding the consonants *f* (ف), *3* (ع), and *l* (ل) of the basic root *f3l* (فعل, "to do") found in the pattern.

In information retrieval and statistical natural language processing, recall and precision are common measures used to gauge a system's performance. Recall (R) is the fraction of target items that a system selected, while the precision (P) is the fraction of selected items that a system got right. A third measure known as F-measure (F)[2] (combines R and P) is used in some situation where R is very high and P is very low (Manning and Schutze, 1999). We implemented R, P, and F to evaluate approaches we present in this paper.

The simple BP matching algorithm was preliminarily evaluated on a subset of an Arabic corpus (referred to as A_Corpus1) obtained from Khoja (1999). It contains 7172 words whose BP instances were identified (this first set of BPs is referred below as data set1). The results showed that the simple BP matching approach has very high recall (99.71%), but low precision (13.73%).

We also tested two slightly modified versions of the simple BP matching algorithm, exploiting information about affixes information and proper name detection, respectively. The first variant was based on the observation that only a limited set of prefixes and suffixes can be attached to a BP stem. In addition, some BP prefixes and suffixes cannot both be concatenated to a BP stem at the same time. These observations led to a first variant of the simple matching algorithm incorporating two post-processing strategies for refining the decisions made by the simple BP matching algorithm. The first refining strategy checks if the produced prefix or suffix is in the list of BP prefixes or suffixes; if it isn't, the stem will be classified as 'Not Broken Plural (NBP)'. The second refining strategy checks if the prefix is a definite article (e.g., *al* (ال, "the"), *wal* (وال , "and the"), *bal* (بال , "with the"), etc.) and the suffix is a BP suffix, and changes the output accordingly. An evaluation of the performance of the simple BP matching algorithm with affix-based refinement strategies on data set1 revealed a slight improvement in precision (16.74%).

We also made a preliminary test evaluating the possible usefulness of incorporating a proper name detector in the system. We manually identified the proper names in data set1, then modified the simple BP matching algorithm to ignore proper names. Our results only showed a small (if significant) improvement in precision (19.86%), that we didn't feel would justify the considerable effort required to develop a proper name detector. As a result, we looked for simpler but more effective ways to improve the algorithm.

3.2 The Restricted BP Matching Approach

The main problem with the simple BP matching approach is that the BP patterns are too general to achieve a good performance. Another way to improve the precision of the algorithm is therefore to obtain more specific BP patterns by restricting the original ones. The idea is to allow only a subset of the alphabet to be used in the meta characters *f* (ف), *3* (ع), and *l* (ل) positions of the patterns (see Section 2), producing a number of more restrictive patterns out of each original BP pattern. A larger number of instances of each BP pattern is required to develop this approach. For this purpose, we used a large corpus of ~18.5 million words (Goweder and De Roeck, 2001). In the remainder of the paper, we refer to this corpus as A_Corpus2. The simple BP matching algorithm with affix-based refinement strategies was used to extract all instances of BPs that occurred in A_Corpus2. We adopted two approaches. In a first experiment we tried to produce the more restrictive patterns by hand. Later we tried to achieve the same goal using a decision tree technique. We discuss the first experiment here, the second in section 3.4. The procedure we followed to identify the BPs in A_Corpus2 is as follows:

1. A word frequencies tool was used to generate word frequencies for A_Corpus2, obtaining 444,761 distinct word types.
2. Each word type was light-stemmed.
3. The word frequencies tool was run again on the stemmed word types, producing roughly 127,000 stem types.
4. The 127,000 stem types were fed to the simple BP matching system to retrieve all

[2] F=2PR/(P+R) for equal weighting.

stems that match BP patterns. The output file, categorised according to each BP pattern, contained about 30,000 cases. Each specific pattern contained a list of stems matching this pattern.

We then studied separately each BP pattern. Some BP patterns were straightforward to restrict. For example, all the stem types matching the BP pattern *Af3lAa* (افعلاء), are shown in Figure 2. There are 107 cases in total. An analysis of the results reveals that only 18 cases are BPs, highlighted (bold and underlined). In the BP pattern *Af3lAa* (افعلاء), the meta characters *f* (ف), *3* (ع), and *l* (ل) are in positions 2, 3, and 4 respectively. The remaining characters - *Alef* (ا) in positions 1 & 5, and *Hamza* (ء) in position 6 - are fixed. Our analysis showed that the stems which have the letter *Ta* (ت) in the 3rd position are not BPs; they are nominalizations of verbs. For example, the word *abtdaa* (ابتداء , "starting") listed on the first row and last column is a noun derived from the verb *yabtdi* (يبتدئ , "he starts"). There are 62 cases of this type. An exceptional rule could be induced to handle nouns derived from verbs. The rule could be written as:

If (the letter *Ta* (ت) matches the meta character *3* (ع)), then Classification = "NBP".

List of all words retrieved by the pattern (افعلاء)===>

ابتناء	ابتلاء	ابتغاء	ابتعاء	ابتداء
اجتزاء	اجتراء	**اثرياء**	**اتقياء**	**ابرياء**
احتشاء	احتساء	احتذاء	اجوراء	اجتلاء
اختباء	احلباء	احتواء	احتماء	احتفاء
ادعياء	اداعاء	اختلاء	اختفاء	اختشاء
ارتشاء	ارتداء	ارتخاء	اربعاء	**اذكياء**
ارdياء	ارتواء	ارتماء	ارتقاء	ارتضاء
استعاء	استضاء	ازيلاء	ازدهاء	ازدراء
استواء	استلاء	استقاء	استفاء	استغاء
اشتهاء	اشتكاء	**اسوياء**	اسوعاء	استياء
اعتلاء	اعتداء	اصطفاء	**اصدقاء**	**اشقياء**
اغتناء	اغتذاء	**اغبياء**	اعتناء	اعتماء
اقتداء	افرقاء	افتراء	افتداء	**اغنياء**
اقوياء	**اقرباء**	اقتناء	اقتفاء	اقتضاء
اكفياء	اكتواء	اكتفاء	اكتساء	اكتراء
انتداء	**انبياء**	انبناء	امتلاء	امتطاء
انتهاء	انتماء	انتقاء	انتفاء	انتشاء
انزواء	انحناء	انجلاء	انجراء	انثناء
انفكاء	انطواء	انطفاء	انضواء	انسباء
اهتداء	انمحاء	انكفاء	**انقياء**	انقضاء
ايلياء	**اولياء**	**اوفياء**	**اوصياء**	اهتراء
			إتقاء	ازيلاء

Total number of cases is 107

Figure 2: Results of the pattern *Af3lAa* (افعلاء).

The simple BP matching algorithm was modified to use the manually restricted BP patterns. The performance of the manual restriction method was evaluated using the same data set used before, data set1. The results show that precision is noticeably improved, to 53.92%. Recall is improved as well, to 100%. The improvement in both recall and precision caused a big increase in the F-measure, to roughly 70%. These results suggested to us that attempting to restrict the BP patterns is worthwhile. In section 3.4, we discuss attempts to find restrictions automatically, using decision tree methods. But the classification of all words in A_Corpus2 as BP or NBP also allowed us to bootstrap a dictionary-based approach. We discuss this next.

3.3 The Dictionary Approach

In information retrieval applications, "the most common measures of system performance are time and space. The shorter the response time, the smaller the space used, the better the system is considered to be" (Baeza-Yates and Ribeiro-Neto, 1999). The fastest way to detect BPs is to use a look-up table which lists all BP stems.

Considering some of the facts about Arabic, discussed in Section 2, it is quite clear that it will be fairly difficult to build look-up tables listing either BP stems or full words from language dictionaries. A_Corpus2, on the other hand - a large resource of modern, unvowelised, freeflowing Arabic text - provided a good foundation, and after the development of the simple and restricted BP matching algorithms discussed in the previous sections, only minor additional effort was required for building such a table (without such tools, collecting the table entries would have been prohibitively expensive).

The dictionary was built as follows:

1. The manually restricted BP matching system was run on the 127,000 stem types, extracted from A_Corpus2 (see section 3.2), to retrieve all types that match (restricted matching) BP patterns. The results were about 12,000 instances in total.
2. We then went through these 12,000 instances, identifying the actual BPs. A list of roughly 3,600 BP stems, alphabetically ordered and categorised according to each BP pattern, was extracted.

3. The list was further revised in collaboration with a linguist, who is an Arabic native speaker. The revised list contained exactly 3,580 BP stems.

We implemented the dictionary approach using hash tables, in which search, insertion, and deletion operations can be done in constant time.

Before carrying an extensive comparison of the dictionary approach to the previous approaches, its performance was first tested on the same data set already used to evaluate both simple and restricted BP matching approaches, data set1. The results of the evaluation show that precision significantly improves (to 81.18%), while recall is still perfect (100%). The F-measure recorded an increase (89.61%) due to the improvement in the precision. The results suggest that the dictionary approach outperforms both the simple and manually restricted BP matching approaches.

3.4 Learning Restrictions Automatically

Decision tree learning is one of the most widely used classification methods. The decision tree learning algorithm C4.5 developed by Quinlan (1993) was used to generate a set of rules in the form of a decision tree and decision rules (if-then statements). Because we are interested in how we could restrict the BP patterns, specifically restricting the meta characters of the BP patterns (Fa, Ain, and Lam), we chose them to be the attributes which describe our data. The outcome (class) of each case is given as BP or NBP. Figure 3 shows the classes and the name & description of each attribute.

BP, NBP.
Fa: *discrete (list of Arabic alaphabet).*
Ain: *discrete (list of Arabic alaphabet).*
Lam: *discrete (list of Arabic alaphabet).*

Figure 3: Set of attributes.

Table 2 lists some examples of the BP pattern *Af3lAa* (افعلاء) to show how instances of the data can be described according to the set of proposed attributes and a classification for each instance.

Table 2: Sample of examples.

Word	Set of Attributes			Class
	Fa	Ain	Lam	
Asdqaa(اصدقاء , "friends")	ص	د	ق	BP
Abtdaa(ابتداء , "starting")	ب	ت	د	NBP
Akhtbaa(اختباء , "hiding")	خ	ت	ب	NBP
Athryaa(اثرياء , "wealthy")	ث	ر	ي	BP

Data balance was an issue to be dealt with before conducting decision tree experiments. For some BP patterns, the number of BP cases is much smaller than the number of NBP cases. In such a situation, we are required to have equal cases for each class (50% for BP and 50% for NBP) because C4.5 tends to classify all the cases as one class with some error rate if there are an insufficient, or small number of cases of one type compared to the other. Balancing the data was achieved by duplicating the infrequent cases until we have an equal number of cases for both classes.

Training data are generated using the simple BP matching algorithm, on the text file containing 127,000 stem types extracted from A_Corpus2 (see section 3.2). The simple BP matching algorithm listed all instances that match every particular BP pattern. So far, we have a list of instances, which are labeled as BP, for each BP pattern, however, many of the cases are not BPs. As a result, we need to revise automatically the classification of each case using the dictionary-based approach (discussed in section 3.3). After the revision, all the cases which are labeled as BPs by the simple BP matching algorithm will be corrected by the dictionary approach. At this stage, each BP pattern will have a list of BP and NBP cases. The BP system will check which class has fewer cases in order to duplicate them to achieve the balancing. Thirty nine output files, one for each BP pattern, were produced by the BP system.

Test data for each BP pattern are also generated by invoking the BP system on a large unseen data set, containing 187,309 words (referred to as data set2) extracted from the Arabic Newswire corpus (a third corpus referred to as A_Corpus3, and totally different from A_Corpus1 and A_Corpus2) in order to test the models produced by C4.5 system.

We generated one classification model for each of the 39 (mutually exclusive) BP patterns, and examined their performance on unseen test cases. Each classification model was trained on a dataset specific to that BP pattern and consisting of 10,000 cases, 50% for each class. The classification models were then evaluated on 39 different test sets (one for each BP pattern). Most of the classifiers were able to achieve the task with very low error rates and high recall & precision. Some models performed the classification without any errors and had a very simple decision tree (e.g., the decision tree and set of rules produced for the BP pattern *Af3lAa* (افعلاء)). This implies that the results are promising; however, some classifiers had large decision trees and suffered from overfitting.

A summary of recall and precision results for both decision trees and set of rules are drawn as histograms to give us a better insight of how each BP pattern performed as shown in Figures 4, 5, 6, and 7. The analysis of the results shows that most of the models (Figures 4&6), representing BP patterns, achieved high recall (except a few of them, such as patterns 16, 27, where the recall was low ≤ 40%). On the other hand, some models (Figures 5&7) performed poorly (precision ≤ 40%), such as patterns 4, 10, 16, 17, 21, and 28. The performance of all combined models achieved an overall recall and precision of approximately 95% and 75% respectively.

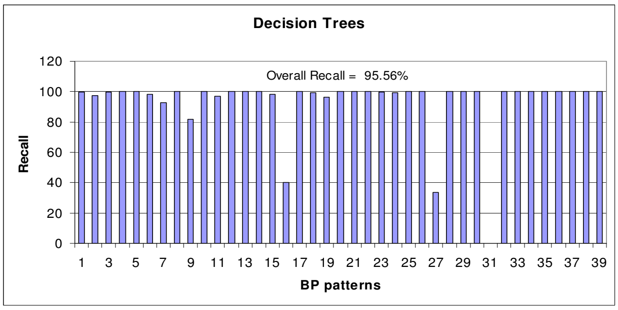

Figure 4: Recall of decision trees for all BP patterns.

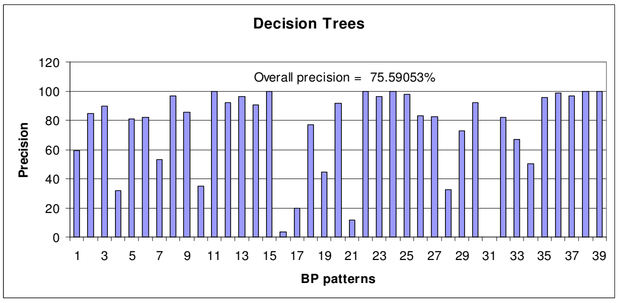

Figure 5: Precision of decision trees for all BP patterns.

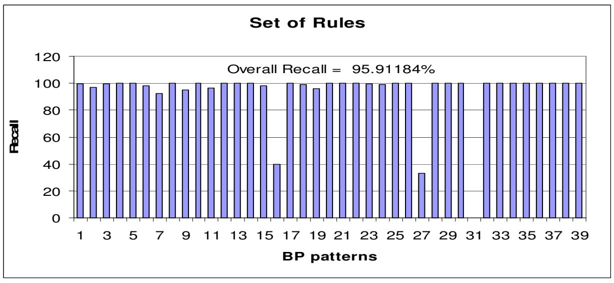

Figure 6: Recall of set of rules for all BP patterns.

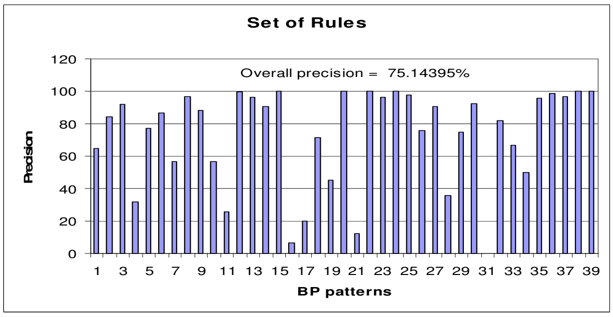

Figure 7: Precision of set of rules for all BP patterns.

4. Comparing the Performance of the BP Identification Approaches

The BP matching methods discussed in the previous section were evaluated on a larger unseen data set, data set2 (the same data set already used to generate test cases to evaluate the decision tree approach, see section 3.4). The BPs in this data set were tagged as follows:

1. A modified version of the dictionary-based BP identification algorithm was run on data set2 to tag all the occurrences of BPs.
2. We manually went through the output twice to revise any mistakes made by the BP identification algorithm.

The evaluation results for the different algorithms on data set2 are listed in Table 3. These results confirm that the simple BP matching approach performed poorly, the restricted BP matching approach improved the performance significantly, a more significant improvement achieved by the decision tree technique, and the dictionary approach outperformed all the approaches. The results also suggest that affix-based refinement strategies improved the performance of the simple matching, the restricted matching, and the dictionary algorithms.

Table 3: An Evaluation of different BP identification algorithms using a large data set (data set2).

BP Ident. Method	Evaluation Criteria		
	R	P	F
SM	99.5%	13.8%	24.2%
SMR	**100%**	**14.5%**	**25.4%**
MRM	99%	49.7%	66.2%
MRMR	**100%**	**52%**	**68.4%**
Dic	98.8%	86.9%	92.5%
DicR	**100%**	**92.3%**	**96.0%**
DT	**95.9%**	**75.1%**	**84.3%**

Acronyms:

Simple Matching → SM

Simple Matching with Refinement → SMR

Manually Restricted Matching → MRM

Manually Restricted Matching with Refinement → MRMR

Dictionary → Dic

Dictionary with Refinement → DicR

Decision Trees → DT

5. An Improved Light-Stemmer and its Task-Based Evaluation

The dictionary-based BP detector with restriction was included in a revised version of the light stemmer described earlier (henceforth: Basic-LStemmer). This revised stemmer (henceforth: BP-LStemmer) first runs the Basic-LStemmer on a word, then invokes the (dictionary-based) BP detector. If the BP detector returns TRUE, the singular form of the word is output; otherwise, the output of the Basic-LStemmer.

The BP-LStemmer was evaluated in an information retrieval task by developing a new indexing method, referred to as “stem+BP”. This new indexing method was compared with the three standard indexing methods (full word, root, and ‘basic’ stem). The Greenstone digital library, developed at the University of Waikato in New Zealand, was used as an information retrieval system for our experiment. A collection of 385 documents (7 different domains) and a set of 50 queries (plausible queries that we might use ourselves were created to search for particular information in different domains) with their relevance judgments, were used to evaluate the four indexing methods.

The results (Figure 8) clearly indicate that the proposed “stem+BP” indexing method significantly outperforms the three standard indexing/stemming methods (p (1-tailed) < .01 both by the Sign test and the Wilcoxon signed-rank test). This suggests that stemming has a substantial effect on information retrieval for highly inflected languages such as Arabic, confirming the results obtained by Al-Kharashi and Evens (1994), Hmeidi et al. (1997), Abu-Salem et al. (1999), Larkey and Connell (2001), and Larkey et al. (2002).

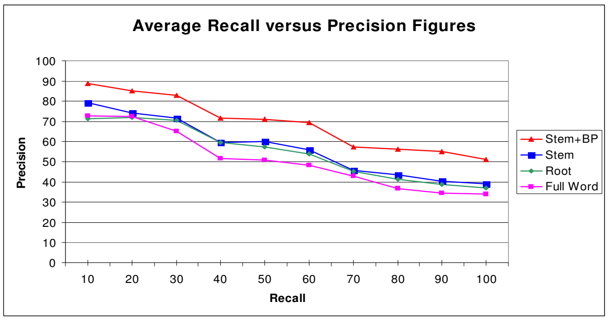

Figure 8: The Average Recall vs. Precision Figures of the Four Indexing Methods for 50 Queries.

6. Conclusion

We discussed several different methods for BP identification: simple BP matching, affix-based simple BP matching, simple BP matching+POS, manually-and-DT restricted, and dictionary-based. Although the simplest methods had poor or mediocre results, they were used to bootstrap better performing methods.

The baseline, the simple BP matching method, has a high recall but a low precision (~14%). We attempted to improve the performance of the BP identification algorithm by (i) using affix information, (ii) identifying proper names, and (iii) restricting the BP patterns. Having implemented the simple and restricted methods, and used them to analyse all the BPs in a large corpus (A_Corpus2), made a dictionary approach possible. All methods were evaluated on a larger data set of 187,000 words. The results confirmed that the restricted method clearly improved the overall performance and the dictionary approach outperformed the other ones.

We also developed a new light-stemming algorithm that conflates both regular and broken plurals with their singular forms. The new light-stemming algorithm was assessed in an information retrieval context, comparing its performance with other stemming algorithms. Our work provides evidence that identifying broken plurals results in an improved performance for information retrieval systems. We found that any form of stemming improves retrieval for Arabic; and that light-stemming with broken plural recognition outperforms standard light-stemming, root-stemming, and no form of stemming.

7. Acknowledgments

We would like to express our gratitude to Shereen Khoja for providing her root stemmer. We would also like to thank the Libyan Secretariat of Education for supporting this work.

8. References

Abuleil, Saleem and Evens, Martha W. (1998). “Discovering Lexical Information by Tagging Arabic Newspaper Text.” Computational Approaches to Semitic Languages, Proceedings of the Workshop.

Abu-Salem, Hani; Al-Omari, Mahmoud; and Evens, Martha W. (1999). “Stemming Methodologies over Individual Query Words for an Arabic Information Retrieval System.” JASIST, 50(6):524-529.

Ali , N. (1988). “Computers and the Arabic Language.” Cairo, Egypt: Al-khat Publishing Press, Ta’reep.

Al-Kharashi, I. and Evens, Martha W. (1994). “Comparing words, stems and roots as index terms in an Arabic Information retrieval system.” JASIST, 45(8):548-560.

Baeza-Yates, Ricardo and Ribeiro-Neto, Berthier (1999). “Modern Information Retrieval”. Addison-Wesley, ACM Press.

Beesley, K. R. (1996) “Arabic finite-state morphological analysis and generation.” In COLING-96: Proceedings of the 16th international conference on Computational Linguistics, vol. 1, pp. 89--94.

Boudelaa , Sami; Gaskell, M. Gareth (2002). “A re-examination of the default system for Arabic plurals.” Psychology Press Ltd, vol. 17, pp. 321-343, 2002.

Cowan, David (1958). “Modern Literary Arabic.” Cambridge University Press.

Goweder, Abduelbaset and De Roeck, Anne (2001). “Assessment of a Significant Arabic Corpus.” ACL 2001. Arabic language Processing. pp. 73-79, 2001.

Haywood, J. A. and Nahmad, H. M. (1976). “A new Arabic Grammar of the written language.” Lund Humphries London.

Hmeidi, Ismail; Kanaan, Ghassan; and Evens, Martha (1997). “Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents.” Journal of the American Society for Information Science. 48(10) (pp. 867-881), 1997.

Idrissi, Ali (1997). “Plural Formation in Arabic.” In Current issues in Linguistic Theory, Perspectives on Arabic Linguistics X. Edited by Mushira Eid and Robert Ratcliffe. Vol 153, pp 123-145.

Khoja, S. and Garside, R. (1999) “Stemming Arabic text.” Computing Department, Lancaster University, Lancaster, United Kingdom. http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps

Kiraz, G. (1996a). Analysis of the Arabic broken plural and diminutive. In Proceedings of the 5th International Conference and Exhibition on Multi-Lingual Computing. Cambridge.

Larkey, L. S. and Connell, M. E. (2001) “Arabic information retrieval at UMass in TREC-10.” In TREC 2001. Gaithersburg: NIST, 2001.

Larkey, L.; Ballesteros, L.; and Connell, M.E (2002). “Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis.” In SIGIR’02, August 11-15, 2002, Tampere, Finland, pp 275–282, 2002.

Manning, Christopher D. and Schutze, Hinrich (1999). “Foundations of Statistical Natural Language Processing.”

McCarthy, John J.; and Prince, Alan S (1990). “Foot and Word in Prosodic Morphology: The Arabic Broken Plural.” Natural Language and Linguistic Theory 8, 209–282.

Quinlan, J. R. (1993). “C4.5: Programs for Machine Learning.” San Mateo: Morgan Kaufmann.