Predicting Phrase Breaks in Classical and Modern Standard Arabic Text

Majdi Sawalha[1], Claire Brierley[2], Eric Atwell[2]
University of Jordan[1] and University of Leeds[2]
[1] Computer Information Systems Dept., King Abdullah II School of IT, University of Jordan, Amman, Jordan
[2] School of Computing, University of Leeds, LS2 9JT, UK
E-mail: sawalha.majdi@gmail.com, scscb@leeds.ac.uk, eric@comp.leeds.ac.uk

Abstract

We train and test two probabilistic taggers for Arabic phrase break prediction on a purpose-built, “gold standard”, boundary-annotated and PoS-tagged Qur’an corpus of 77430 words and 8230 sentences. In a related LREC paper (Brierley *et al.*, 2012), we cover dataset build. Here we report on comparative experiments with off-the-shelf N-gram and HMM taggers and coarse-grained feature sets for syntax and prosody, where the task is to predict boundary locations in an unseen test set stripped of boundary annotations by classifying words as breaks or non-breaks. The preponderance of non-breaks in the training data sets a challenging baseline success rate: 85.56%. However, we achieve significant gains in accuracy with the trigram tagger, and significant gains in performance recognition of minority class instances with both taggers via Balanced Classification Rate. This is initial work on a long-term research project to produce annotation schemes, language resources, algorithms, and applications for Classical and Modern Standard Arabic.

Keywords: phrase break prediction, N-gram and HMM taggers, boundary-annotated and PoS-tagged Qur’an Corpus

1. Introduction

Chunking text via automatic assignment of sentence-medial and sentence-terminal prosodic-syntactic boundaries is a Natural Language Processing (NLP) and machine learning task which attempts to simulate human parsing and phrasing strategies. The latter are represented by “gold standard” boundary annotations in a speech corpus. Phrase break classifiers are typically trained and tested on such datasets, and assume prior sentence segmentation and part-of-speech (PoS) tagging for input text. In a related paper, we report on a purpose-built, boundary-annotated dataset: the 77430-word *Qur’an* corpus of Classical Arabic (Brierley *et al.*, 2012). Here, we utilise that language resource to develop and evaluate two probabilistic taggers (n-gram and HMM) for the phrase break prediction task, using two different feature sets. We regard the Qur’an as a reputable ‘gold standard’ for phrasing in Arabic because *recitation* is integral to this text, and many editions (§3) already carry prescriptive boundary mark-up representative of the long-established traditions of Arabic linguistics. Hence we plan to assess the naturalness and intelligibility of outputs from our best-performing tagger over a sample of Modern Standard Arabic (MSA) text (*ibid*).

2. Phrase Break Prediction

Automated phrase break prediction is a natural language processing (NLP) task within the Text-to-Speech (TTS) synthesis pipeline, and sub-divides input sentences into meaningful chunks to copy the way in which a native speaker might parse or phrase the utterance. This equates to classifying junctures between words, or the words themselves, in terms of a finite set of boundary types, for example `breaks` or `non-breaks`. Establishing these delimiters is an essential component of the symbolic linguistic representation of text as output to a speech synthesizer.

2.1 General Procedure for Phrase Break Prediction

Phrase break prediction assumes prior sentence segmentation and part-of-speech tagging for input text, and therefore punctuation and syntax are traditionally used as classificatory features. Another prerequisite is a boundary-annotated and part-of-speech (PoS) tagged corpus (*ibid*) as ‘gold standard’ for developing phrase break classifiers. The classifier is trained on a substantive sample of ‘gold-standard’ boundary-annotated text, and tested on a smaller, unseen sample from the same source *minus* the boundary annotations.

2.2 Machine Learning Approaches to Phrase Break Prediction

There are two generic approaches to machine learning: rule-based or probabilistic. Phrase break models exemplifying these two approaches are: (i) Liberman and Church’s *chinks ‘n’ chunks* algorithm (1992); and (ii) Taylor and Black’s Markov model (1998) used in Edinburgh’s *Festival*[1] Speech Synthesis system. In the former, chinks are closed-class function words, while chunks are open-class content words; the algorithm inserts a phrase break at every punctuation mark, and whenever a content word is immediately followed by a function word. Taylor and Black’s statistical model conditions the probability of juncture type (*i.e.* $P(j_i)$ in Equation 1) on: (i) the *prior probability* of each class given the immediate context (*i.e.* the PoS trigram in which that juncture is embedded or $P(C_i \mid j_i)$ in Equation 1); and (ii) the *likelihood* of each class given the previous sequence of N juncture types, where in this case, N = 6 (Equation 1).

(1) $$P(j_i) \propto P(j_i \mid J_{i-1}^{N}).P(C_i \mid j_i)$$

[1] http://www.cstr.ed.ac.uk/projects/festival/

2.3 Metrics for Phrase Break Prediction

Performance is primarily evaluated in terms of *accuracy*, namely: the number of correct predictions – or the sum of *true positives* and *true negatives* (TP + TN) – made during test. There are also other relevant metrics such as *f-score* and balanced classification rate (BCR). The former is the trade-off between, or weighted mean, of *recall* (*i.e.* TP total / total number of boundaries in the sample), and *precision* (*i.e.* TP total / total number of boundaries retrieved). The latter (*i.e.* BCR) mitigates against high accuracy scores arising from class imbalance, a typical scenario for phrase break prediction since instances of the majority class `(non-breaks)` greatly outnumber minority class instances `(breaks)` in the corpus. BCR is computed as the average of *breaks-correct* and *non-breaks-correct* and thus considers relative class distributions (Equation 2).

$$(2) \qquad BCR = 0.5 * \left(\frac{\text{TP}}{\text{total true positives}} + \frac{\text{TN}}{\text{total true negatives}}\right)$$

3. Experimental Dataset: the Qur'an

A prerequisite for developing and evaluating phrase break classifiers is a 'gold standard' boundary-annotated and PoS-tagged corpus. The 77430-word Qur'an is a reputable choice of experimental dataset principally because it comes complete with its own linguistically-informed and fine-grained boundary annotation scheme: the system of stops and starts (وَقْف وَ ٱبْتِدَاء or *waqf wa ibtidā*) as one component of traditional recitation mark-up or *Tajwīd* (*cf.* Denny, 1989). It is also an *original* choice of dataset, prompting the following related (and long-term) research questions (Brierley *et al.*, 2012):

1. Do Qur'anic Arabic speech rhythms still inform native speaker intuitions for processing (*e.g.* parsing and phrasing) Modern Standard Arabic?
2. Can prosodic-syntactic boundary correlates in the Qur'an be leveraged for Modern Standard Arabic natural language engineering applications?

An additional incentive is the availability of an open-source, PoS-tagged version of the Qur'an: we have used version 2.0 of *QAC* or the *Qur'anic Arabic Corpus* (Dukes, 2010). Traditional Arabic grammar classifies words into one of three syntactic categories **`{noun, verb, particle}`**, and we therefore retain this coarse-grained feature set as one experimental variant to see if any useful basic patterns emerge. We also map this sparse tagset to the ten major syntactic categories defined in QAC (Dukes and Habash, 2010): **`{nouns; pronouns; nominals; adverbs; verbs; prepositions; 'lām prefixes; conjunctions; particles; disconnected letters}`** for further experimentation. Boundary annotations in the Qur'an are very fine-grained, and we plan to make full use of this in future work. For the present, we have adopted a widely-used recitation style (*ḥafṣ bin 'Āṣim*), and collapsed eight degrees of boundary strength in a reputable edition of the text[2] into two sparse subsets: (i) **`breaks`** versus **`non-breaks`**; and (ii) **`{major, minor, none}`**. The original eight *Tajwīd* categories consist of three major boundary types; four minor boundary types, and one *prohibited* stop. Figure 1 shows the following data from a sample verse in our corpus: MSA word; coarse-grained PoS; finer-grained PoS; recitation mark-up (if any); major (| |) or minor (|) boundary (if any); break or non-break status; English transliteration.

Readers will note from Figure 1 that Arabic text in the Qur'an is fully diacritized: all short vowels are marked by diacritics in the text. This is not the case for Modern Standard Arabic (MSA), where short vowel diacritics are missing. Therefore, restoring full vowelization is an essential preprocessing step for morphological analysis, PoS-tagging, and parsing of MSA. Our approach to phrase break prediction for MSA will implement the SALMA Vowelizer (Sawalha, 2011), as one module within the SALMA Tagger (*ibid*), to automatically restore short vowels in MSA, since full vowelization is assumed in our algorithms.

4. N-gram and HMM Taggers

We implement a trigram tagger based on the ***N***atural ***L***anguage ***T***ool***K***it's (Bird *et al.*, 2009) **`Ngram Tagger`** class to assign boundaries to a corpus of Qur'anic Arabic which is segmented into sentences and PoS-tagged, and where outputs from the tagger can be evaluated against 'gold standard' boundary annotations in the dataset (Brierley *et al.*, 2012). We also implement an HMM or sequence model based on NLTK's **`HiddenMarkovModelTagger`** class. Input to the tagger is the same in both cases: our purpose-built Qur'an dataset (*ibid*) is segmented into 8230 sentence tokens, and each sentence token is represented as a list of tuples from which we specify permutations of features that match our research questions (§3, 5). A sample *Qur'anic sentence* is given in Figure 2.

4.1 The Trigram Tagger

Our trigram tagger is coded in Python and trained on Qur'an text represented as **`(PoS, boundary-type)`** or **`((word, PoS), boundary-type)`** pairings. For the former, it assigns the most likely boundary type (*e.g.* **`break`** or **`non-break`**) based on the current PoS, plus the two preceding boundary types as context. Figure 3 is an adaptation from Bird *et al.* (2009, p.204): shaded areas denote context, and the target for prediction is *italicised.*

Readers will note that this trigram tagger is based on Python dictionaries: a look-up table is consulted to determine an appropriate tag for each instance; and the tagger backs off to a majority class tagger (*i.e.* tags the instance as `non-break`) if look-up fails.

[2] http://tanzil.net/download

```
الرَّحْمَنِ N   NOMINAL   -    -    non-break   the-most-gracious
الرَّحِيمِ N   NOMINAL   ۞    ||   break       the-most-merciful
```

Figure 1: Corpus data from which to extract features as input to the tagger

[((بِسْمِ , N), non-break), ((اللَّهِ , N), non-break), ((الرَّحْمَنِ , N), non-break), ((الرَّحِيمِ , N, break)]

Figure 2: Arabic word and PoS tag pairings are mapped to boundary types and serve as input features to the classifier during training

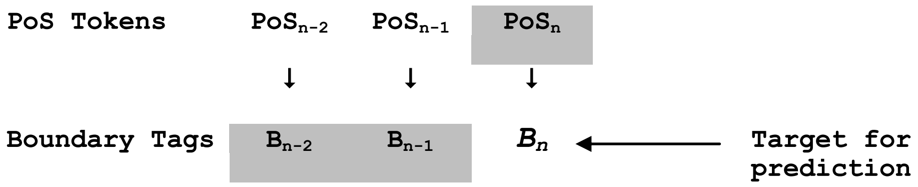

Figure 3: Abstract representation of trigram context used for predicting breaks or non-breaks

4.2 The HMM Tagger

One drawback of this method is that there is no way to revise previously assigned boundaries as the algorithm iterates through the list (*i.e.* the sentence). To resolve this, we also implement NLTK's HMM tagger for comparative evaluation (§5). This is based on Huang *et al* (2001, Chapter 8). For these initial experiments, we have simply used the `train()` and `evaluate()` methods with default parameter settings, plus the `train()` method with labeled and unlabeled sequences (*i.e.* training and test set splits), to determine the optimal/most probable combination of break types for each sentence via the Maximum Likelihood Estimate (MLE), which maximizes the joint probability of symbol/state sequences. The HMM tagger generates a probability distribution over all possible boundary types - either **`break`** versus **`non-break`** (the two-class problem), or **`major/minor/non-break`** (the three-class problem). The product of these probabilities then gives a probability score for each boundary sequence, and the highest-scoring sequence is then chosen.

5. Evaluation

In Section 3 of this paper, we have discussed some overarching research goals. The immediate research questions pertaining to this study are as follows:

1. Can we learn any reliable prosodic-syntactic boundary patterns for Arabic from coarse-grained data?
2. What basic patterns emerge to differentiate major and minor chunk boundaries?
3. Which sequence model (n-gram tagger or HMM tagger) gives best results with coarse-grained features?

5.1 Methodology

To address these questions, we comparatively evaluate the performance of a trigram tagger and an HMM tagger firstly on our Qur'an dataset, with different permutations of features. The first round of experiments uses tripartite PoS categories **`{noun, verb, particle}`** to predict: (i) **`breaks`** versus **`non-breaks`**; and (ii) boundaries of type: **`major, minor, none`**. The second round uses ten PoS features (§3) to resolve both tasks: binary classification, and the 3-class problem. The Qur'an dataset is split into the same partitions for training and test in both cases; the training set comprises 70112 words and 7381 sentences, and the test set comprises 7318 words and 849 sentences. The number of sentences in the test set also equates to the number of major breaks in the test set. Non-breaks in the test set total 6469, and this total sub-divides into 6261 non-breaks and 208 minor breaks for the 3-class problem. These are supervised machine learning experiments that assume the classes are mutually exclusive, such that each Arabic word will be resolved as an instance of one, *and only one*, specific boundary type.

5.1.1. Test Set Selection

Test set sentences were not randomly selected. There is agreement on the provenance of most Qur'anic verses in terms of whether they originate from the Prophet's period of residence in Mecca or Medina. However, there are 21 (out of 114) chapters where Mecca/Medina verse associations are in doubt (*cf.* Sharaf, 2012). Meccan and Medinan verses differ stylistically (*ibid*), and therefore the 21 disputed chapters were used as our test set, since they constitute a representative sample of both styles and a fair test for a tagger trained on the rest of the corpus.

5.2 Confusion Matrices

Tagger accuracy for each classification task can be expressed as an overall percentage calculated by summing the number of correct predictions for each boundary type, and dividing this total by the total word count (*i.e.* the total number of items to be classified). Output predictions are presented as a confusion matrix where false positives and false negatives (FPs and FNs) are used to infer basic issues in performance. Table 1 is an example of the confusion matrix for the two-class problem, where shaded area counts constitute the proportion of correct predictions (*true positives* and *true negatives*) retrieved during test for our trigram tagger using very coarse-grained PoS. Readers will note that class distributions in the test set are highly skewed: 6261 `non-breaks` versus 1057 `breaks`.

		Predicted +ve	Predicted -ve
Breaks	1057	**380**	677
Non-breaks	6261	167	**6094**

Table 1: Example confusion matrix for binary classification with the trigram tagger using (word, PoS) pairings

5.3 The Two-Class Problem

Table 2 displays results for binary classification experiments with both taggers, and feature set permutations which include/exclude words PoS-tagged at two different levels of granularity. What is immediately obvious is that data skew (*i.e.* the over-preponderance of `non-breaks`) sets a high baseline accuracy of 85.56%. Nevertheless, the trigram tagger in Runs 1 and 5 significantly outperforms the baseline for both syntactic feature sets: 88.47% for 3 PoS categories, and 88.44% for 10 PoS categories. Success rate for the HMM tagger is below par. We therefore use the alternative metric of BCR or Balanced Classification Rate (*cf.* Equation 2) to assess how well the model has learnt the concept. The *trigram* tagger in Run 1 correctly predicts 380 breaks set against the baseline prediction of zero. Hence BCR for Run 1 represents a significant gain in performance. What is additionally interesting is that the *HMM taggers* in Runs 2 and 6 also represent a statistically significant gain in performance in terms of BCR even when set against Run 1.

RUN	TAGGER	INCLUDE WORD?	NUMBER OF POSTAGS	NUMBER OF CLASSES	ACCURACY	BCR	TPs	TNs	FPs	FNs
Base	Baseline	✓	3 or 10	2	85.56%	0.50	0	6261	0	1057
Base	Baseline		3 or 10	2	85.56%	0.50	0	6261	0	1057
1	Trigram	✓	3	2	88.47%	0.67	380	6094	167	677
2	HMM	✓	3	2	82.63%	0.72	601	5446	815	456
3	Trigram		3	2	85.56%	0.50	0	6261	0	1057
4	HMM		3	2	85.56%	0.50	0	6261	0	1057
5	Trigram	✓	10	2	88.44%	0.66	372	6100	161	685
6	HMM	✓	10	2	82.66%	0.72	600	5449	812	457
7	Trigram		10	2	86.31%	0.55	108	6208	53	949
8	HMM		10	2	86.32%	0.55	114	6203	58	943

Table 2: Experimental results for binary phrase break classification on the Qur'an test set of 7318 words.

5.4 The Three-Class Problem

Table 3 records results for tripartite classification.

RUN	TAGGER	INCLUDE WORD?	NUMBER OF POSTAGS	NUMBER OF CLASSES	ACCURACY	BCR	TPs	TNs	FPs	FNs
Base	Baseline	✓	3 or 10	3	85.56%	0.50	0	6261	0	1057
Base	Baseline		3 or 10	3	85.56%	0.50	0	6261	0	1057
9	Trigram	✓	3	3	88.69%	0.65	333	6157	128	700
10	HMM	✓	3	3	81.46%	0.64	371	5590	821	536
11	Trigram		3	3	85.56%	0.50	0	6261	0	1057
12	HMM		3	3	85.56%	0.50	0	6261	0	1057
13	Trigram	✓	10	3	88.62%	0.65	323	6162	122	711
14	HMM	✓	10	3	81.18%	0.63	361	5580	834	543
15	Trigram		10	3	86.17%	0.54	98	6208	63	949
16	HMM		10	3	86.62%	0.55	117	6222	45	934

Table 3: Experimental results for tripartite phrase break classification on the Qur'an test set of 7318 words.

Significant gains in both accuracy and BCR over baseline performance were achieved by the trigram tagger for the 3-class problem using both feature sets in Runs 9 and 13: 88.69% and 88.62% respectively. The HMM tagger also achieved significant gains in terms of BCR (Runs 10 and 14), and in one experiment (Run 16), where words were disabled as a feature, improved on baseline success rate, albeit at the expense of BCR.

6. Conclusions and Further Work

The trigram and HMM taggers in these experiments are prototypes, using fairly coarse-grained syntactic features only. Our plans for future research include enriching our dataset with: (i) very fine-grained morpho-syntactic analyses using the SALMA tagger (Sawalha, 2011; Sawalha and Atwell, 2010); (ii) more fine-grained boundary annotations; and (iii) projected prosody (*cf.* Brierley, 2011; Brierley and Atwell, 2010) potentially as part of an ongoing project (Atwell *et al.*, 2011). Sharable experience and insights of interest to fellow corpus linguists are to be gained from the present implementation and evaluation of sequence models for Arabic phrase break prediction. As with English (Liberman and Church, 1992; Taylor and Black, 1998; Ingulfsen *et al.*, 2005), syntactic information proves a reliable feature, but what is especially interesting is that our highest accuracy scores have been achieved with a very coarse-grained feature set with a long-established history: the tripartite classification of Arabic words as **`{noun, verb, particle}`** in traditional Arabic grammar (*cf.* Brierley *et al.*, 2012, §4). This is original research in that: (i) our goal is to derive chunking algorithms for Arabic speech and language applications from traditional prosodic mark-up in the Qur'an; and (ii) our underpinning question is whether Qur'anic Arabic speech rhythms still inform native speaker intuition and judgment when processing Modern Standard Arabic. Our two papers for LREC 2012, along with an earlier paper (Brierley *et al.*, 2011), represent groundwork for a larger-scale project to produce annotation schemes, language resources, algorithms, and applications for Classical and Modern Standard Arabic.

7. References

Atwell, E., Brierley, C., Dukes, K., Sawalha, M. and Sharaf, A.M. 2011. 'An Artificial Intelligence Approach to Arabic and Islamic Content on the Internet.' *National Information Technology Symposium (NITS).* Riyadh, Saudi Arabia.

Bird, S., Klein, E. and Loper, E. 2009. *Natural Language Processing with Python.* Sebastopol, CA. O'Reilly Media, Inc.

Brierley, C. 2011. *'Prosody Resources and Symbolic Prosodic Feartures for Automated Phrase Break Prediction.'* PhD Thesis. School of Computing. University of Leeds.

Brierley, C. and Atwell, E. 2010. 'ProPOSEC: a Prosody and PoS Spoken English Corpus.' In *Proceedings of LREC 2010: Language Resources and Evaluation Conference*. Valetta, Malta. May 2010.

Brierley, C., Sawalha, M. and Atwell, E. 2012. 'Open-Source Boundary-Annotated Corpus for Arabic Speech and Language Processing.' In *Proceedings of LREC 2012: Language Resources and Evaluation Conference*. Istanbul, Turkey. May 2012.

Brierley, C., Sawalha, M. and Atwell, E. 2011. 'Arabic Phonetics and Phonology for Text Analytics and Natural Language Processing Applications.' PowerPoint presentation for Arabic Phonetics and Phonology PG Workshop. York.

Denny, F.M. 1989. 'Qur'an Recitation: A Tradition of Oral Performance and Transmission.' In *Oral Tradition*. 4/1-2: 5-26

Dukes, K. 2010. *The Quranic Arabic Corpus (v. 2.0).* Online. Accessed: August 2011. http://corpus.quran.com

Dukes, K. and Habash, N. 2010. 'Morphological Annotation of Qur'anic Arabic.' In *Proceedings of LREC 2010: Language Resources and Evaluation Conference.* Valletta, Malta.

Ingulfsen, T., Burrows, T. and Buchholz, S. 2005. 'Influence of Syntax on Prosodic Boundary Prediction.' In *Proceedings, INTERSPEECH 2005.* 1817-1820.

Liberman, M.Y. and Church, K.W. 1992. 'Text Analysis and Word Pronunciation in Text-to-Speech Synthesis.' In *Advances in Speech Signal Processing*. Furui S. and Sondhi, M.M. (eds.). New York. Marcel Dekker Inc.

Sawalha, M. 2011. *'Open-Source Resources and Standards for Arabic Word Structure Analysis: Fine Grained Morphological Analysis of Arabic Text Corpora.'* PhD. Thesis. School of Computing. University of Leeds.

Sawalha, M. and Atwell, E. 2010. 'Fine-Grain Morphological Analyzer and Part-of-Speech Tagger for Arabic Text.' In *Proceedings of LREC'10: Language Resources and Evaluation Conference*, Valetta, Malta. May 2010.

Sharaf, A.M. 2011. 'Macci and Madani Shurahs.' Online. Accessed: October 2011. http://www.textminingthequran.com/wiki/Makki_and_Madani_Surahs

Taylor, P. and Black, A.W. 1998. 'Assigning Phrase-Breaks from Part-of-Speech Sequences.' In *Computer Speech and Language*. 12.2: 99-117.