Memory-Based Vocalization of Arabic

Sandra Kübler, Emad Mohamed

Indiana University
Department of Linguistics
1021 E 3rd St.
Bloomington, IN-47405
USA
{skuebler,emohamed}@indiana.edu

Abstract

The problem of vocalization, or diacritization, is essential to many tasks in Arabic NLP. Arabic is generally written without the short vowels, which leads to one written form having several pronunciations with each pronunciation carrying its own meaning(s). In the experiments reported here, we define vocalization as a classification problem in which we decide for each character in the unvocalized word whether it is followed by a short vowel. We investigate the importance of different types of context. Our results show that the combination of using memory-based learning with only a word internal context leads to a word error rate of 6.64%. If a lexical context is added, the results deteriorate slowly.

1. Introduction

The problem of vocalization, or diacritization, is essential to many tasks in Arabic NLP. Arabic is generally written without the short vowels, which leads to one written form having several pronunciations with each pronunciation carrying its own meaning(s). The word form 'mskn' is an example for a highly ambiguous word. Its possible pronunciations include 'maskan' (home), 'musakkin' (analgesic), 'masakn' (they-*fem*. have held), or 'musikn' (they-*fem*. have been held). The importance of vocalization become clear when we look at how Google Translate renders 'A$tryt Almskn mn AlSydlyp' (I bought a pain killer from the pharmacy): as 'I bought the home from the pharmacy'. This error would not occur if the input to the translation system were vocalized in a first step before the actual translation process. However, vocalization is far from trivial: the example above shows that the vocalized words of a single unvocalized form differ in their parts-of-speech as well as in their meaning. This shows that vocalization performs implicit POS tagging and word sense disambiguation. It is also obvious that word forms cannot be vocalized in isolation, the task is heavily dependent on the context of the word.

In the experiments reported here, we investigate the importance of different types of context. We follow (Zitouni et al., 2006) in defining vocalization as a classification problem in which we decide for each character in the unvocalized word whether it is followed by a short vowel. We investigate how well the task can be performed if only context from the same word is available as compared to having access to a lexical context of 5 words on each side. We also investigate which types of features are the most important ones. Lastly, we investigate the learning curve to determine how much training data we need for reliable results.

2. Previous Research

The first approaches to the vocalization of Arabic defined the problem word-based, i.e. the task was to determine for each word the complete vocalized form. (Gal, 2002) uses a bigram HMM model for vocalizing the Qur'an and achieves a word error rate (WER) of 14%. His error analysis showed that the errors resulted mostly from unknown words. (Kirchhoff et al., 2002) extend a unigram model by a heuristic for unknown words, which retrieves the most similar unlexicalized word and then applies edit distance operations to turn it into the unknown word. They reach a WER of 16.5% on conversational Arabic. (Nelken and Shieber, 2005) tackle the problem with weighted finite-state transducers. For known words, morphological units are used for retrieving the vocalization while unknown words are vocalized based on the sequence of characters. They reach a WER of 12.8%. (Zitouni et al., 2006) use a maximum entropy model in combination with a character based classification. Their features are based on single characters of the focus word, morphological segments, and POS tags. They reach a WER of 7.9%. A comparison of the different approaches shows that the definition of vocalization as inserting vowels between characters results in the lowest WER. However, this study leaves the lexical context of words completely unexplored. In the present study, we will investigate this area of research.

3. Experimental Setup

3.1. Data

We used the Penn Arabic Treebank (Bies and Maamouri, 2003) as the data source. The treebank is encoded in Buckwalter transliteration (Buckwalter, 2002) and is available in a vocalized and an unvocalized version. From the treebank, we extracted 170 000 words from the AFP section (part 1 v 2.0) and approximately 160 000 words from the Ummah section (part 2 v2.0).

As mentioned previously, we defined vocalization as a classification problem: For each character in the focus word, the learner needs to decide whether the character is followed by a short vowel and what the short vowel is. We will call this character the focus charcter. The task also involves the restoration of the shadda (double consonant, long consonant, gemination) but, at present, it does not include case endings.

w_{-5}	w_{-4}	w_{-3}	w_{-2}	w_{-1}	c_{-5}	c_{-4}	c_{-3}	c_{-2}	c_{-1}	c	c_1	c_2	c_3	c_4	c_5	w_1	w_2	w_3	w_4	w_5	v
kl	"$y"	tgyr	fy	HyAp	_	_	_	_	_	A	l	m	t	$	r	styfn	knt	EndmA	Evrt	Ely	_
kl	"$y"	tgyr	fy	HyAp	_	_	_	_	A	l	m	t	$	r	d	styfn	knt	EndmA	Evrt	Ely	_
kl	"$y"	tgyr	fy	HyAp	_	_	_	A	l	m	t	$	r	d	_	styfn	knt	EndmA	Evrt	Ely	u
kl	"$y"	tgyr	fy	HyAp	_	_	A	l	m	t	$	r	d	_	_	styfn	knt	EndmA	Evrt	Ely	a
kl	"$y"	tgyr	fy	HyAp	_	A	l	m	t	$	r	d	_	_	_	styfn	knt	EndmA	Evrt	Ely	a
kl	"$y"	tgyr	fy	HyAp	A	l	m	t	$	r	d	_	_	_	_	styfn	knt	EndmA	Evrt	Ely	ĩ

Table 1: The word 'Almt$rd' represented with one instance per word; the class represents the vowel to be inserted after the character.

The features used for determining the short vowel following the focus character consist of the focus character itself (c), its local context in terms of neighboring characters within the focus word, and a more global context of neighboring words. For the local context, 5 characters to the left ($c_{-5} \ldots c_{-1}$), 5 characters to the right ($c_1 \ldots c_5$) are used; for the lexical context, 5 words to the left ($w_{-5} \ldots w_{-1}$), and 5 words to the right ($w_1 \ldots w_5$). The last value in the vector (v) provides the correct classification, i.e. the short vowel to be inserted after c, or - in cases where no vowel is inserted in that position. The instance for the Arabic word 'Almt$rd', for example, is shown in Table 1.

For most of the experiments, we used 10-fold cross validation, the only exception is the experiment concerning the size of the training set. In order to simulate a real-life situation, we did not build the folds randomly but rather sequentially, thus ensuring that a single fold contains consecutive articles, which may cover different topics from the other folds. However, we made sure that all instances of a word were put in the same fold.

3.2. Methods

For classification, we used a memory-based learner, TiMBL (Daelemans et al., 2007). Memory-based learning is a lazy-learning paradigm, which assumes that learning does not consist of abstraction of the training instances into rules or probabilities. Instead, the learner uses the training instance directly. As a consequence, training consists in storing the instances in an instance base, and classification finds the k nearest neighbor in the instance base and chooses their most frequent class as the class for the new instance. Memory-based learning has been proven to have a suitable bias for many NLP problems (Daelemans et al., 1999). One of the reasons for this success is that natural language exhibits a high percentage of subregularities or irregularities, which cannot be distinguished from noise. Eager learning paradigms smooth over all these cases while memory-based learning still has access to the original instance. Thus, if a new instance is similar enough to one of these irregular instances, it can be correctly classified as such.

Memory-based learning was chosen for two reasons: First, this approach weights features based on information gain or gain ratio (Daelemans et al., 2007), thus giving some indication of the most and the least important features. Additionally, it is a paradigm that is capable of handling symbolic features with a high number of different feature values. This allows us to use complete context words as features.

	CER	WER
baseline	–	47.2
character context	2.22	6.64
left word context	2.26	7.06
word context	2.35	6.86

Table 2: The results of the vocalization experiments with TiMBL.

Parameter settings for TiMBL were determined first. The best results were obtained for all experiments with the IB1 algorithm with similarity computed as weighted overlap, relevance weights computed with gain ratio, and the number of k nearest neighbors (or in TiMBL's case, nearest distances) equal to 1.

4. Results

The results of our experiments with regard to different contexts as well the baseline are shown in Table 2. We evaluate the error rate based on characters (CER) and based on words (WER). The baseline experiment was set up so that the classifier was presented with 11 words: the focus word, 5 context words to its left, and 5 context words to its right. The results for the baseline show that vocalization is a difficult task even in our data set where a word on average has only 1.67 vocalizations. This figure is considerably lower than the average on normal texts. (Debili et al., 2002) found that on average, each unvocalized word type has 2.9 vocalized versions, and there is an average of 11.6 vocalized versions per word token in a text.

Relevant features. The next three lines in Table 2 reports the results for the experiments in which we define the task as deciding for each character whether it is followed by a vowel. The experiment in line 2 uses only a character context of 5 characters to each side of the focus character but ignores the context words, i.e. the features from c_{-5} to c_5 in Figure 1 are used. The next experiment uses the lexical context to the left of the focus word in addition to the character context but ignores the context words on the right, i.e. the features from w_{-5} to c_5 are used. Finally, the last experiment uses all features shown in Figure 1, i.e. it uses the character context as well as the lexical feature to the left and to the right of the focus word. When going from classifying complete words to classifying characters separately, the results improve dramatically. This method results in a WER of 6.64%; to our knowledge, the highest reported result (but notice that (Zitouni et al., 2006) use a different training set). Surprisingly, adding the context words does

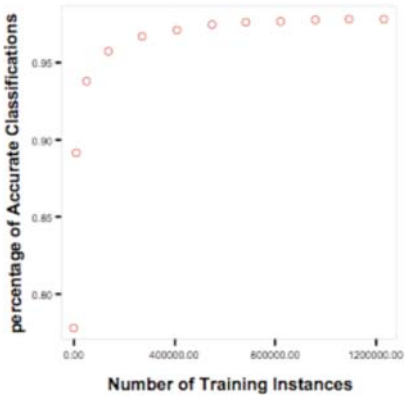

Figure 1: The learning curve.

not improve the classification results. On the contrary, it results in a lower WER. This is unexpected, we would have expected that at least in cases where the vocalizations have different parts of speech, the lexical context would provide important information. One possible explanation for these negative results may be data sparseness. However, if we use the lexical context on both sides of the focus word, the CER is lower but the WER is higher than in the experiment with the left context only. This shows that the individual decisions concerning single vowels become more difficult but the recognition of complete words becomes more stable. Thus, in some cases, the lexical context does improve classification. This also becomes evident when we compare the results of single folds in the 10-fold setting. Some of the folds have better results in the left context setting, and some in the full context setting.

Next, we look at the weights that TiMBL assigns to the different features in the character based experiments. Here, the results are very stable. If we look at the gain ratio weights, in all experiments over all folds, we get the same ordering of features. The feature with the highest weight is the character following the focus character, c_1. The next most important feature is the focus character, c. The third most important character is the next character to the left, c_{-1}, followed by all its preceding characters c_{-5} to c_{-2}, followed by all the characters to the right of c_1: c_2, c_5, c_3, and c_4.

Size of the training set. The next question to investigate concerns the importance of the training set size. In order to investigate how much training data we need for the task, we conducted an experiment in which we started with a small training set containing 1000 character instance, and then continually increased the training set size to the full training set size of 1 230 723 character instances. The test set was kept stable, we used one of the folds for testing. In order to ensure reliable results, we chose a fold that resulted in average results in the ten-fold experiments reported above. All the experiments were performed with the best feature set determined in the previous experiments, i.e. with characters from the focus word as the only features. The learning curve is shown in Figure 1. When training on a set of only 1 000 characters, the WER is 47%, but raising the size of the training set reduces the WER to 27%. The saturation point is reached at approximately 700 000 characters (which corresponds to 5 folds), with a WER of 6.9%. After this point, there are only minor improvements, and the WER reaches 6.64% for the whole training set[1].

5. Conclusion and Future Work

In the experiments reported here, we have investigated the vocalization of Arabic. The results show that the word internal context provides enough information for vocalizing a high percentage correctly. The best parameter and feature setting results in an error rate of 6.64%, which is more than one percent point lower than the results presented by (Zitouni et al., 2006) even though our system did not have access to either word segments or POS tags. Adding lexical context as additional features did not increase the performance of the memory-based classifier TiMBL. Interestingly, the most informative feature is the character following the focus character although in general, the left character context within the focus word is more informative than the right character context. The learning curve shows that at least in the experiments with features only from within the focus word, a training set of 700 000 characters is sufficient for reliable results. For the future, we are planning to use a stemmer for Arabic to reduce the lexical features to stems in order to alleviate the sparse data problem concerning the lexical features. Additionally, we will follow (Zitouni et al., 2006) and include part of speech information for all the words as well. Since the tagset of the Penn Arabic treebank is rather fine grained, we expect to reach the best results by reducing the tagset to a manageable level, following (Diab, 2007). A further line of investigation concerns the use of previous classification within a word for the classification of the next character.

6. References

Ann Bies and Mohamed Maamouri. 2003. Penn Arabic Treebank guidelines. Technical report, LDC, University of Pennsylvania.

Tim Buckwalter. 2002. Arabic morphological analyzer version 1.0. Linguistic Data Consortium.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34:11–43. Special Issue on Natural Language Learning.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2007. TiMBL: Tilburg memory based learner – version 6.1 – reference guide. Technical Report ILK 07-07, Induction of Linguistic Knowledge, Computational Linguistics, Tilburg University.

Fahti Debili, Hadhebi Achour, and Emna Souissi. 2002. De l'etiquetage grammatical a la voyellation automatique de l'arabe. Technical report, Correspondances de l'Institut de Recherche sur le Maghreb Contemporain.

Mona Diab. 2007. Towards an optimal POS tag set for Arabic processing. In *Proceedings of the International Con-*

[1]We will present results for the experiments with lexical features in the final version of the paper.

ference on Recent Advances in Natural Language Processing, RANLP 2007, pages 157–161, Borovets, Bulgaria.

Ya'akov Gal. 2002. An HMM approach to vowel restoration in Arabic and Hebrew. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA.

Katrin Kirchhoff, Jeff Bilmes, John Henderson, Richard Schwartz, Mohamed Noamany, Pat Schone, Gang Ji, Sourin Das, Melissa Egan, Feng He, Dimitra Vergyri, Daben Liu, and Nicolae Duta. 2002. Novel speech recognition models for Arabic - final report of the JHU summer workshop. Technical report, Johns Hopkins University.

Rani Nelken and Stuart Shieber. 2005. Arabic diacritization using weighed finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Language*, Ann Arbor, MI.

Imed Zitouni, Jeffrey Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, COLING-ACL-2006*, Sydney, Australia.