Automatic Diacritization for Low-Resource Languages Using a Hybrid Word and Consonant CMM

Robbie A. Haertel, Peter McClanahan, and Eric K. Ringger
Department of Computer Science
Brigham Young University
Provo, Utah 84602, USA
rah67@cs.byu.edu, petermcclanahan@gmail.com, ringger@cs.byu.edu

Abstract

We are interested in diacritizing Semitic languages, especially Syriac, using only diacritized texts. Previous methods have required the use of tools such as part-of-speech taggers, segmenters, morphological analyzers, and linguistic rules to produce state-of-the-art results. We present a low-resource, data-driven, and language-independent approach that uses a hybrid word- and consonant-level conditional Markov model. Our approach rivals the best previously published results in Arabic (15% WER with case endings), without the use of a morphological analyzer. In Syriac, we reduce the WER over a strong baseline by 30% to achieve a WER of 10.5%. We also report results for Hebrew and English.

1 Introduction

Abjad writing systems omit vowels and other diacritics. The ability to restore these diacritics is useful for personal, industrial, and governmental purposes—especially for Semitic languages. In its own right, the ability to diacritize can aid language learning and is necessary for speech-based assistive technologies, including speech recognition and text-to-speech. Diacritics are also useful for tasks such as segmentation, morphological disambiguation, and machine translation, making diacritization important to Natural Language Processing (NLP) systems and intelligence gathering. In alphabetic writing systems, similar techniques have been used to restore accents from plain text (Yarowsky, 1999) and could be used to recover missing letters in the compressed writing styles found in email, text, and instant messages.

We are particularly interested in diacritizing Syriac, a low-resource dialect of Aramaic, which possesses properties similar to Arabic and Hebrew. This work employs conditional Markov models (CMMs) (Klein and Manning, 2002) to diacritize Semitic (and other) languages and requires only diacritized texts for training. Such an approach is useful for languages (like Syriac) in which annotated data and linguistic tools such as part-of-speech (POS) taggers, segmenters, and morphological analyzers are not available. Our main contributions are as follows: (1) we introduce a hybrid word and consonant CMM that allows access to the diacritized form of the previous words; (2) we introduce new features available in the proposed model; and (3) we describe an efficient, approximate decoder. Our models significantly outperform existing low-resource approaches across multiple related and unrelated languages and even achieve near state-of-the-art results when compared to resource-rich systems.

In the next section, we review previous work relevant to our approach. Section 3 then motivates and describes the models and features used in our framework, including a description of the decoder. We describe our data in Section 4 and detail our experimental setup in Section 5. Section 6 presents our results. Finally, Section 7 briefly discusses our conclusions and offers ideas for future work.

2 Previous Work

Diacritization has been receiving increased attention due to the rising interest in Semitic languages, cou-

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 519–527,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

pled with the importance of diacritization to other NLP-related tasks. The existing approaches can be categorized based on the amount of resources they require, their basic unit of analysis, and of course the language they are targeting. Probabilistic systems can be further divided into generative and conditional approaches.

Existing methodologies can be placed along a continuum based on the quantity of resources they require—a reflection of their cost. Examples of resources used include morphological analyzers (Habash and Rambow, 2007; Ananthakrishnan et al., 2005; Vergyri and Kirchhoff, 2004; El-Sadany and Hashish, 1989), rules for grapheme-to-sound conversion (El-Imam, 2008), transcribed speech (Vergyri and Kirchhoff, 2004), POS tags (Zitouni et al., 2006; Ananthakrishnan et al., 2005), and a list of prefixes and suffixes (Nelken and Shieber, 2005). When such resources exist for a particular language, they typically improve performance. For instance, Habash and Rambow's (2007) approach reduces the error rate of Zitouni et al.'s (2006) by as much as 30% through its use of a morphological analyzer. In fact, such resources are not always available. Several data-driven approaches exist that require only diacritized texts (e.g., Kübler and Mohamed, 2008; Zitouni et al., 2006; Gal, 2002) which are relatively inexpensive to obtain: most literate speakers of the target language could readily provide them.

Apart from the quantity of resources required, diacritization systems also differ in their basic unit of analysis. A consonant-based approach treats each consonant[1] in a word as a potential host for one or more (possibly null) diacritics; the goal is to predict the correct diacritic(s) for each consonant (e.g., Kübler and Mohamed, 2008). Zitouni et al. (2006) extend the problem to a sequence labeling task wherein they seek the best *sequence* of diacritics for the consonants. Consequently, their approach has access to previously chosen diacritics.

Alternatively, the basic unit of analysis can be the full, undiacritized word. Since morphological analyzers produce analyses of undiacritized words, diacritization approaches that employ them typically fall into this category (e.g., Habash and Rambow, 2007; Vergyri and Kirchoff, 2004). Word-based, low-resource solutions tend to treat the problem as word-level sequence labeling (e.g., Gal, 2002).

Unfortunately, word-based techniques face problems due to data sparsity: not all words in the test set are seen during training. In contrast, consonant-based approaches rarely face the analogous problem of previously unseen consonants. Thus, one low-resource solution to data sparsity is to use consonant-based techniques for unknown words (Ananthakrishnan et al., 2005; Nelken and Shieber, 2005).

Many of the existing systems, especially recent ones, are probabilistic or contain probabilistic components. Zitouni et al. (2006) show the superiority of their conditional-based approaches over the best-performing generative approaches. However, the instance-based learning approach of Kübler and Mohamed (2008) slightly outperforms Zitouni et al. (2006). In the published literature for Arabic, the latter two have the best low-resource solutions. Habash and Rambow (2007) is the state-of-the-art, high-resource solution for Arabic. To our knowledge, no work has been done in this area for Syriac.

3 Models

In this work, we are concerned with diacritization for Syriac for which a POS tagger, segmenter, and other tools are not readily available, but for which diacritized text is obtainable.[2] Use of a system dependent on a morphological analyzer such as Habash and Rambow's (2007) is therefore not cost-effective. Furthermore, we seek a system that is applicable to a wide variety of languages. Although Kübler and Mohamed's (2008) approach is competitive to Zitouni et al.'s (2006), instance-based approaches tend to suffer with the addition of new features (their own experiments demonstrate this). We desire to add linguistically relevant features to improve performance and thus choose to use a conditional model. However, unlike Zitouni et al. (2006), we use a hybrid word- and consonant-level approach based on the following observations (statistics taken from the Syriac training and development sets explained in Section 4):

[1] We refer to all graphemes present in undiacritized texts as consonants.

[2] Kiraz (2000) describes a morphological analyzer for Syriac that is not publicly available and is costly to reproduce.

1. Many undiacritized words are unambiguous: 90.8% of the word types and 63.5% of the tokens have a single diacritized form.

2. Most undiacritized word types have only a few possible diacritizations: the average number of possible diacritizations is 1.11.

3. Low-frequency words have low ambiguity: Undiacritized types occurring fewer than 5 times have an average of 1.05 possible diacritizations.

4. Diacritized words not seen in the training data occur infrequently at test time: 10.5% of the diacritized test tokens were not seen in training.

5. The diacritics of previous words can provide useful morphological information such as person, number, and gender.

Contrary to observations 1 and 2, consonant-level approaches dedicate modeling capacity to an exponential (in the number of consonants) number of possible diacritizations of a word. In contrast, a word-level approach directly models the (few) diacritized forms seen in training. Furthermore, word-based approaches naturally have access to the diacritics of previous words if used in a sequence labeler, as per observation 5. However, without a "backoff" strategy, word-level models cannot predict a diacritized form not seen in the training data. Also, low-frequency words by definition have less information from which to estimate parameters. In contrast, abundant information exists for each diacritic in a consonant-level system. To the degree to which they hold, observations 3 and 4 mitigate these latter two problems. Clearly a hybrid approach would be advantageous.

To this end we employ a CMM in which we treat the problem as an instance of sequence labeling at the word level with less common words being handled by a consonant-level CMM. Let $\mathbf{u}$ be the undiacriatized words in a sentence. Applying an order o Markov assumption, the distribution over sequences of diacritized words $\mathbf{d}$ is:

$$P\left(\mathbf{d}|\mathbf{u}\right) = \prod_{i=1}^{\|\mathbf{d}\|} P\left(d_i|\mathbf{d}_{i-o\ldots i-1}, \mathbf{u}; \boldsymbol{\omega}, \boldsymbol{\gamma}, \alpha\right) \quad (1)$$

in which the local conditional distribution of a diacritized word is an interpolation of a word-level model ($\boldsymbol{\omega}_{u_i}$) and a consonant-level model ($\boldsymbol{\gamma}$):

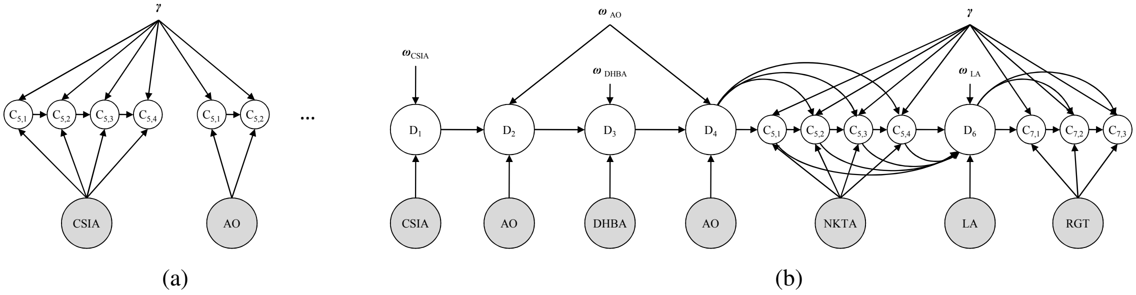
$$\begin{aligned} &P\left(d_i|\mathbf{d}_{i-o\ldots i-1}, \mathbf{u}; \boldsymbol{\omega}, \boldsymbol{\gamma}, \alpha\right) = \\ &\quad \alpha P\left(d_i|\mathbf{d}_{i-o\ldots i-1}, \mathbf{u}; \boldsymbol{\omega}_{u_i}\right) + \\ &\quad (1-\alpha) P\left(d_i|\mathbf{d}_{i-o\ldots i-1}, \mathbf{u}; \boldsymbol{\gamma}\right) \end{aligned}$$

We let the consonant-level model be a standard CMM, similar to Zitouni et al. (2006), but with access to previously diacritized words. Note that the order of this "inner" CMM need not be the same as that of the outer CMM.

The parameter α reflects the degree to which we trust the word-level model. In the most general case, α can be a function of the undiacritized words and the previous o diacritized words. Based on our earlier enumerated observations, we use a simple delta function for α: we let α be 0 when $\mathbf{u}_i$ is rare and 1 otherwise. We leave discussion for what constitutes a "rare" undiacritized type for Section 5.2.

Figure 1b presents a graphical model of a simple example sentence in Syriac. The diacritization for non-rare words is predicted for a whole word, hence the random variable D for each such word. These diacritized words D_i depend on previous D_{i-1} as per equation (1) for an order-1 CMM (note that the capitalized A, I, and O are in fact consonants in this transliteration). Because "NKTA" and "RGT" are rare, their diacritization is represented by a consonant-level CMM: one variable for each possible diacritic in the word. Importantly, these consonant-level models have access to the previously diacritized word (D_4 and D_6, respectively).

We use log-linear models for all local distributions in our CMMs, i.e., we use maximum entropy (maxent) Markov models (McCallum et al., 2000; Berger et al., 1996). Due to the phenomenon known as *d*-separation (Pearl and Shafer, 1988), it is possible to independently learn parameters for each word model $\boldsymbol{\omega}_{u_i}$ by training only on those instances for the corresponding word. Similarly, the consonant model can be learned independent of the word models. We place a spherical normal prior centered at zero with a standard deviation of 1 over the weights of all models and use an L-BFGS minimizer to find the MAP estimate of the weights for all the models (words and consonant).



(a)

(b)

Figure 1: Graphical models of Acts 20:33 in Syriac, *CSIA AO DHBA AO NKTA LA RGT* 'silver or gold or garment I have not coveted,' using Kiraz's (1994) transliteration for (a) the initial portion of a consonant-level-only model and (b) a combined word- and consonant-level model. For clarity, both models assume a consonant-level Markov order of 1; (b) shows a word-level Markov order of 1. For simplicity, the figure further assumes that additional features come only from the current (undiacritized) word.

Note that Zitouni et al.'s (2006) model is a special case of equation (1) where all words are rare, the word-level Markov order (o) is 0, and the consonant-level Markov order is 2. A simplified version of Zitouni's model is presented in Figure 1a.

3.1 Features

Our features are based on those found in Zitouni et al. (2006), although we have added a few of our own which we consider to be one of the contributions of this paper. Unlike their work, our consonant-level model has access to previously diacritized words, allowing us to exploit information noted in observation 5.

Each of the word-level models shares the same set of features, defined by the following templates:

- The prefixes and suffixes (up to 4 characters) of the previously *diacritized* words.
- The string of the actual diacritics, including the null diacritic, from each of the previous o diacritized words and n-grams of these strings; a similar set of features is extracted but without the null diacritics.
- Every possible (overlapping) n-gram of all sizes from $n = 1$ to $n = 5$ of undiacritized words contained within the window defined by 2 words to the right and 2 to the left. These templates yield 15 features for each token.
- The count of how far away the current token is from the beginning/end of the sentence up to the Markov order; also, their binary equivalents.

The first two templates rely on diacritizations of previous words, in keeping with observation 5.

The consonant-level model has the following feature templates:

- The current consonant.
- Previous diacritics (individually, and n-grams of diacritics ending in the diacritic prior to the current consonant, where n is the consonant-level Markov order).
- Conjunctions of the first two templates.
- Indicators as to whether this is the first or last consonant.
- The first three templates independently conjoined with the current consonant.
- Every possible (overlapping) n-gram of all sizes from $n = 1$ to $n = 11$ consisting of consonants contained within the window defined by 5 words to the right and 5 to the left.
- Same as previous, but available diacritics are included in the window.
- Prefixes and suffixes (of up to length 4) of previously diacritized words conjoined with previous diacritics in the current token, both individually and n-grams of such.

This last template is only possible because of our model's dependency on previous diacritized words.

3.2 Decoder

Given a sentence consisting of undiacritized words, we seek the most probable sequence of diacritized words, i.e., $\arg\max_{\mathbf{d}} P(\mathbf{d}|\mathbf{u}...)$. In sentences containing no rare words, the well-known Viterbi algorithm can be used to find the optimum.

However, as can be seen in Figure 1b, predictions in the consonant-level model (e.g., $C_{5,1...4}$) depend on previously diacritized words (D_4), and some diacritized words (e.g., D_6) depend on diacritics in the previous rare word ($C_{5,1...4}$). These dependencies introduce an exponential number of states (in the length of the word) for rare words, making exact decoding intractable. Instead, we apply a non-standard beam during decoding to limit the number of states for rare words to the n-best (locally). This is accomplished by using an independent "inner" n-best decoder for the consonant-level CMM to produce the n-best diacritizations for the rare word given the previous diacritized words and other features. These become the only states to and from which transitions in the "outer" word-level decoder can be made. We note this is the same type of decoding that is done in pipeline models that use n-best decoders (Finkel et al., 2006). Additionally, we use a traditional beam-search of width 5 to further reduce the search space both in the outer and inner CMMs.

4 Data

Although our primary interest is in the Syriac language, we also experimented with the Penn Arabic Treebank (Maamouri et al., 2004) for the sake of comparison with other approaches. We include Hebrew to provide results for yet another Semitic language. We also apply the models to English to show that our method and features work well outside of the Semitic languages. A summary of the datasets, including the number of diacritics, is found in Figure 2. The number of diacritics shown in the table is less than the number of possible predictions since we treat contiguous diacritics between consonants as a single prediction.

For our experiments in Syriac, we use the New Testament portion of the Peshitta (Kiraz, 1994) and treat each verse as if it were a sentence. The diacritics we predict are the five short vowels, as well as *Se̊yāmē*, *Rukkākhā*, *Quššāyā*, and *linea ocultans*.

lang	diacs	train	dev	test
Syriac	9	87,874	10,747	11,021
Arabic	8	246,512	42,105	51,664
Hebrew	17	239,615	42,133	49,455
English	5	1,004,073	80,156	89,537

Figure 2: Number of diacritics and size (in tokens) of each dataset

For Arabic, we use the training/test split defined by Zitouni et al. (2006). We group all words having the same P index value into a sentence. We build our own development set by removing the last 15% of the sentences of the training set. Like Zitouni, when no solution exists in the treebank, we take the first solution as the gold tag. Zitouni et al. (2006) report results on several different conditions, but we focus on the most challenging of the conditions: we predict the standard three short vowels, three *tanween*, *sukuun*, *shadda*, and all case endings. (Preliminary experiments show that our models perform equally favorably in the other scenarios as well.)

For Hebrew, we use the Hebrew Bible (Old Testament) in the Westminster Leningrad Codex (Zefania XML Project, 2009). As with Syriac, we treat each verse as a sentence and remove the paragraph markers (*pe* and *samekh*). There is a large number of diacritics that could be predicted in Hebrew and no apparent standardization in the literature. For these reasons, we attempt to predict as many diacritics as possible. Specifically, we predict the diacritics whose unicode values are 05B0-B9, 05BB-BD, 05BF, 05C1-C2, and 05C4. We treat the following list of punctuation as consonants: *maqaf*, *paseq*, *sof pasuq*, *geresh*, and *gershayim*. The cantillation marks are removed entirely from the data.

Our English data comes from the Penn Treebank (Marcus et al., 1994). We used sections 0–20 as training data, 21–22 as development data, and 23–24 as our test set. Unlike words in the Semitic languages, English words can begin with a vowel, requiring us to prepend a prosthetic consonant to every word; we also convert all English text to lowercase.

5 Experiments

For all feature engineering and tuning, we trained and tested on training and development test sets, respectively (as specified above). Final results are reported by folding the development test set into the training data and evaluating on the blind test set. We retain only those features that occur more than once.

For each approach, we report the Word Error Rate (WER) (i.e., the percentage of words that were incorrectly diacritized), along with the Diacritic Error Rate (DER) (i.e., the percentage of diacritics, including the null diacritic, that were incorrectly predicted). We also report both WER and DER for only those words that were not seen during training (UWER and UDER, respectively). We found that precision, recall, and f-score were nearly perfectly correlated with DER; hence, we omit this information for brevity.

5.1 Models for Evaluation

In previous work, Kübler et al. (2008) report the lowest error rates of the low-resource models. Although their results are not directly comparable to Zitouni et al. (2006), we have independently confirmed that the former slightly outperforms the latter using the same diacritics and on the same dataset (see Figure 4), thereby providing the strongest published baseline for Arabic on a common dataset. We denote this model as **`kübler`** and use it as a strong baseline for all datasets.

For the Arabic results, we additionally include Zitouni et al.'s (2006) lexical model (**`zitouni-lex`**) and their model that uses a segmenter and POS tagger (**`zitouni-all`**), which are not immediately available to us for Syriac. For yet another point of reference for Arabic, we provide the results from the state-of-the-art (resource-rich) approach of Habash and Rambow (2007) (**`habash`**). This model is at an extreme advantage, having access to a full morphological analyzer. Note that for these three models we simply report their published results and do not attempt to reproduce them.

Since **`kübler`** is of a different model class than ours, we consider an additional baseline that is a consonant-level CMM with access to the same information, namely, only those consonants within a window of 5 to either side (**`ccmm`**). This is equivalent to a special case of our hybrid model wherein both the word-level and the consonant-level Markov order are 0. The features that we extract from this window are the windowed n-gram features.

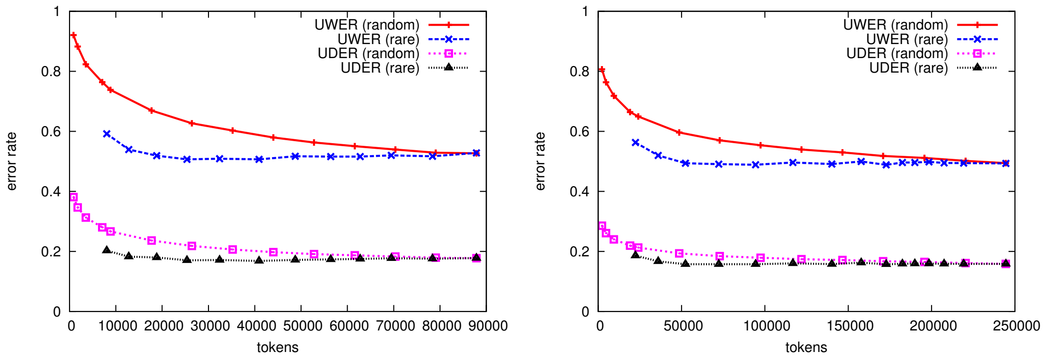
In order to assess the utility of previous diacritics and how effectively our features leverage them, we build a model based on the methodology from Section 3 but specify that all words are rare, effectively creating a consonant-only model that has access to the diacritics of previous words. We call this model **`cons-only`**. We note that the main difference between this model and **`zitouni-lex`** are features that depend on previous diacritized words.

Finally, we present results using our full hybrid model (**`hybrid`**). We use a Markov order of 2 at the word and consonant level for both **`hybrid`** and **`cons-only`**.

5.2 Consonant-Level Model and Rare Words

The hybrid nature of **`hybrid`** naturally raises the question of whether or not the inner consonant model should be trained only on rare words or on all of the data. In other words, is the distribution of diacritics different in rare words? If so, the consonant model should be trained only on rare words. To answer this question, we trained our consonant-level model (**`cons-only`**) on words occurring fewer than n times. We swept the value of the threshold n and compared the results to the same model trained on a random selection of words. As can be seen in Figure 3, the performance on unknown words (both UWER and UDER) using a model trained on rare words can be much lower than using a model trained on the same amount of randomly selected data. In fact, training on rare words can lead to a lower error rate on unknown words than training on all tokens in the corpus. This suggests that the distribution of diacritics in rare words is different from the distribution of diacritics in general. This difference may come from foreign words, especially in the Arabic news corpus.

While this phenomenon is more pronounced in some languages and with some models more than others, it appears to hold in the cases we tried. We found the WER for unknown words to be lowest for a threshold of 8, 16, 32, and 32 for Syriac, Arabic, Hebrew, and English, respectively.



(a) Syriac

(b) Arabic

Figure 3: Learning curves showing impact on consonant-level models when training on rare tokens for Syriac and Arabic. Series marked "rare" were trained with the least common tokens in the dataset.

	Approach	WER	DER	UWER	UDER
Syriac	kübler	15.04	5.23	64.65	18.21
	ccmm	13.99	4.82	**54.54**	**15.18**
	cons-only	12.31	5.03	55.68	19.09
	hybrid	**10.54**	**4.29**	55.16	18.86
Arabic	zitouni-lex	25.1	8.2	NA	NA
	kübler	23.61	7.25	66.69	20.51
	ccmm	22.63	6.61	57.71	16.10
	cons-only	**15.02**	**5.15**	48.10	15.76
	hybrid	17.87	5.67	**47.85**	**15.63**
	zitouni-all	18.0	5.5	NA	NA
	habash	14.9	4.8	NA	NA
Hebrew	kübler	30.60	12.96	89.52	36.86
	ccmm	29.67	12.05	80.02	**29.39**
	cons-only	23.39	10.92	75.70	33.34
	hybrid	**22.18**	**10.71**	**74.38**	32.40
English	kübler	10.54	4.38	**54.96**	**16.31**
	ccmm	11.60	4.71	58.55	16.34
	cons-only	8.71	3.87	58.93	17.85
	hybrid	**5.39**	**2.38**	57.24	16.51

Figure 4: Results for all languages and approaches

6 Discussion of Results

Since Syriac is of primary interest to us, we begin by examining the results from this dataset. Syriac appears to be easier to diacritize than Arabic, considering it has a similar number of diacritics and only one-third the amount of data. On this dataset, **`hybrid`** has the lowest WER and DER, achieving nearly 30% and 18% reduction in WER and DER, respectively, over **`kübler`**; it reduces both error rates over **`cons-only`** by more than 14%. These results attest to the effectiveness of our model in accounting for the observations made in Section 3.

A similar pattern holds for the Hebrew and English datasets, namely that **`hybrid`** reduces the WER over **`kübler`** by 28% to upwards of 50%; **`cons-only`** also consistently and significantly outperforms **`kübler`** and **`ccmm`**. However, the reduction in error rate for our **`cons-only`** and **`hybrid`** models tends to be lower for DER than WER in all languages except for English. In the case of **`hybrid`**, this is probably because it is inherently word-based. Having access to entire previous diacritized words may be a contributing factor as well, especially in **`cons-only`**.

When comparing model classes (**`kübler`** and **`ccmm`**), it appears that performance is comparable across all languages, with the maxent approach enjoying a slight advantage except in English. Interestingly, the maxent solution usually handles unknown words better, although it does not specifically target this case. Both models outperform **`zitouni-lex`** in Arabic, despite the fact that they use a much simpler feature set, most notably, the lack of previous diacritics. In the case of **`ccmm`** this may be attributable in part to our use of an L-BFGS optimizer, convergence criteria, feature selection, or other potential differences not noted in Zitouni et al. (2006). We note that the maxent-based approaches are much more time and memory intensive.

Using the Arabic data, we are able to compare our methods to several other published results.

The **cons-only** model significantly outperforms **zitouni-all** despite the additional resources to which the latter has access. This is evidence supporting our hypothesis that the diacritics from previous words in fact contain useful information for prediction. This empirically suggests that the independence assumptions in consonant-only models are too strict.

Perhaps even more importantly, our low-resource method approaches the performance of **habash**. We note that the differences may not be statistically significant, and also that Habash and Rambow (2007) omit instances in the data that lack solutions. In fact, **cons-only** has a lower WER than all but two of the seven techniques used by Habash and Rambow (2007), which use a morphological analyzer.

Interestingly, **hybrid** does worse than **cons-only** on this dataset, although it is still competitive with **zitouni-all**. We hypothesize that the observations from Section 3 do not hold as strongly for this dataset. For this reason, using a smooth interpolation function (rather than the abrupt one we employ) may be advantageous and is an interesting avenue for future research.

One last observation is that the approaches that use diacritics from previous words (i.e., **cons-only** and **hybrid**) usually have lower sentence error rates (not shown in Figure 4). This highlights an advantage of observation 5: that dependencies on previously diacritized words can help ensure a consistent tagging within a sentence.

7 Conclusions and Future Work

In this paper, we have presented a low-resource solution for automatic diacritization. Our approach is motivated by empirical observations of the ambiguity and frequency of undiacritized and diacritized words as well as by the hypothesis that diacritics from previous words provide useful information. The main contributions of our work, based on these observations, are (1) a hybrid word-level CMM combined with a consonant-level model for rare words, (2) a consonant-level model with dependencies on previous diacritized words, (3) new features that leverage these dependencies, and (4) an efficient, approximate decoder for these models. As expected, the efficacy of our approach varies across languages, due to differences in the actual ambiguity and frequency of words in these languages. Nevertheless, our models consistently reduce WER by 15% to nearly 50% over the best performing low-resource models in the literature. In Arabic, our models approach state-of-the-art despite not using a morphological analyzer. Arguably, our results have brought diacritization very close to being useful for practical application, especially when considering that we evaluated our method on the most difficult task in Arabic, which has been reported to have double the WER (Zitouni et al., 2006).

The success of this low-resource solution naturally suggests that where more resources are available (e.g., in Arabic), they could be used to further reduce error rates. For instance, it may be fruitful to incorporate a morphological analyzer or segmentation and part-of-speech tags.

In future work, we would like to consider using CRFs in place of MEMMs. Also, other approximate decoders used in pipeline approaches could be explored as alternatives to the one we used (e.g., Finkel et al., 2006). Additionally, we wish to include our model as a stage in a pipeline that segments, diacritizes, and labels morphemes. Since obtaining data for these tasks is substantially more expensive, we hope to use active learning to obtain more data.

Our framework is applicable for any sequence labeling task that can be done at either a word or a sub-word (e.g., character) level. Segmentation and lemmatization are particularly promising tasks to which our approach could be applied.

Finally, for the sake of completeness, we note that more recent work has been done based on our baseline models that has emerged since the preparation of the current work, particularly Zitouni et al. (2009) and Mohamed et al. (2009). We wish to address any improvements captured by this more recent work such as the use of different data sets and addressing problems with the *hamza* to decrease error rates.

Acknowledgments

We thank Imed Zitouni, Nizar Habash, Sandra Kübler, and Emad Mohamed for their assistance in reconstructing datasets, models, and features.

References

S. Ananthakrishnan, S. Narayanan, and S. Bangalore. 2005. Automatic diacritization of Arabic transcripts for automatic speech recognition. In *Proceedings of the International Conference on Natural Language Processing*.

A. L. Berger, S. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71.

Y. A. El-Imam. 2008. Synthesis of the intonation of neutrally spoken Modern Standard Arabic speech. *Signal Processing*, 88(9):2206–2221.

T. A. El-Sadany and M. A. Hashish. 1989. An Arabic morphological system. *IBM Systems Journal*, 28(4):600–612.

J. R. Finkel, C. D. Manning, and A. Y. Ng. 2006. Solving the problem of cascading errors: Approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 618–626.

Y. 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*, pages 1–7.

N. Habash and O. Rambow. 2007. Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56.

G. Kiraz. 1994. Automatic concordance generation of Syriac texts. In R. Lavenant, editor, *VI Symposium Syriacum 1992*, pages 461–471, Rome, Italy.

G. A. Kiraz. 2000. Multitiered nonlinear morphology using multitape finite automata: a case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105.

D. Klein and C. D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 9–16.

S. Kübler and E. Mohamed. 2008. Memory-based vocalization of Arabic. In *Proceedings of the LREC Workshop on HLT and NLP within the Arabic World*.

M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, pages 591–598.

E. Mohamed and S. Kübler. 2009. Diacritization for real-world Arabic texts. In *Proceedings of Recent Advances in Natural Language Processing 2009*.

R. Nelken and S. M. Shieber. 2005. Arabic diacritization using weighted finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 79–86.

J. Pearl and G. Shafer. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufman, San Mateo, CA.

D. Vergyri and K. Kirchhoff. 2004. Automatic diacritization of Arabic for acoustic modeling in speech recognition. In *Proceedings of the COLING 2004 Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73.

D. Yarowsky. 1999. A comparison of corpus-based techniques for restoring accents in Spanish and French text. *Natural language processing using very large corpora*, pages 99–120.

Zefania XML Project. 2009. Zefania XML bible: Leningrad codex. `http://sourceforge.net/projects/zefania-sharp/files/Zefania\%20XML\%20Bibles\%204\%20hebraica/Leningrad\%20Codex/sf_wcl.zip/download`.

I. Zitouni and R. Sarikaya. 2009. Arabic diacritic restoration approach based on maximum entropy models. *Computer Speech & Language*, 23(3):257–276.

I. Zitouni, J. S. Sorensen, and R. 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*, pages 577–584.