An HMM Approach to Vowel Restoration in Arabic and Hebrew

Ya’akov Gal

Division of Engineering and Applied Sciences

Harvard University

Cambridge, MA 02138

gal@eecs.harvard.edu

Abstract

Semitic languages pose a problem to Natural Language Processing since most of the vowels are omitted from written prose, resulting in considerable ambiguity at the word level. However, while reading text, native speakers can generally vocalize each word based on their familiarity with the lexicon and the context of the word. Methods for vowel restoration in previous work involving morphological analysis concentrated on a single language and relied on a parsed corpus that is difficult to create for many Semitic languages. We show that Hidden Markov Models are a useful tool for the task of vowel restoration in Semitic languages. Our technique is simple to implement, does not require any language specific knowledge to be embedded in the model and generalizes well to both Hebrew and Arabic. Using a publicly available version of the Bible and the Qur’an as corpora, we achieve a success rate of 86% for restoring the exact vowel pattern in Arabic and 81% in Hebrew. For Hebrew, we also report on 87% success rate for restoring the correct phonetic value of the words.

1 Introduction

In both Hebrew and in Arabic, modern written texts are composed in script that leaves out most of the vowels of the words. Because many words that have different vowel patterns may appear identical in a vowel-less setting, considerable ambiguity exists at the word level.

In Hebrew, Levinger et al. (1995) computed that 55% out of 40,000 word tokens taken from a corpus of the Israeli daily *Ha’aretz* were ambiguous. For example, the non-voweled Hebrew word ספר, written in Latin transliteration as SPR, may represent the noun “book” (pronounced */sepher/*), the third person singular form of the verb “to count” (pronounced */saphar/*) or at least four other possible interpretations. In Arabic, there are almost five possible morphological analyses per word on average (Beesley 1998). Take, for example, the Arabic word كتاب , written in Latin transliteration as KTAAB. One possible interpretation is the noun “book” (pronounced */kitaab/*) and another is the plural of the noun “secretary”, (pronounced */kuttaab/*). Further contributing to this ambiguity is the fact that Hebrew and Arabic morphology is complex: most words are derived from roots that are cast in templates that govern the ordering of letters and provide semantic information. In addition, prefixes and suffixes can also be attached to words in a concatenative manner, resulting in a single string that represents verb inflections, prepositions, pronouns, and connectives.

Vowel restoration in Hebrew and Arabic text is a non-trivial task. In both languages, vowels are marked by both letters and diacritics. In Hebrew, there are twelve different vowel diacritics, and in general, most diacritics are left out of modern script. In Arabic, there are six vowels, which can be divided into three pairs consisting of a short vowel and a long vowel. Each pair corresponds to a different phonetic value. In written Arabic text, the short vowels are generally left out.

Surprisingly, native speakers of Arabic or Hebrew can, in most cases, accurately vocalize words in text based on their context and the speaker's knowledge of the grammar and lexicon of the language. However, speakers of Hebrew are not as successful in restoring the exact vowel diacritics of words. Since many vowels have the same pronunciation in modern Hebrew, and speakers of Hebrew generally use non-voweled script in reading and writing text, they are not familiar with the precise vowel pattern of words.

Throughout this paper, we refer to a word that is fully voweled,[1] i.e. supplied with its full diacritical marking, as *diacritisized* (Beesley 1998). A system that could restore the diacritisized form of scripts, i.e. supply the full diacritical markings, would greatly benefit non-native speakers, sufferers of dyslexia and could assist in diacritisizing children's and poetry books, a task that is currently done manually.

2 A Statistical Approach

Identifying contextual relationships is crucial in deciphering lexical ambiguities in both Hebrew and Arabic and is commonly used by native speakers. Hidden Markov Models have been traditionally used to capture the contextual dependencies between words (Charniak 1995). We demonstrate the utility of Hidden Markov Models for the restoration of vowels in Hebrew and Arabic. As we show, our model is straightforward and simple to implement. It consists of hidden states that correspond to diacritisized words from the training corpus, in which each hidden state has a single emission leading to an undiacritisized (non-voweled) word observation. Our model does not require any handcrafted linguistic knowledge and is robust in the sense that it generalizes well to other languages. The rest of this paper is organized as follows: in Section 3, we provide an explanation of the corpora we used in our experiment. Section 4 and 5 describe the models we designed as well as our experimental setup for evaluating them. Section 6 describes related work done in morphological analysis and vowel restoration in Hebrew and in Arabic. Finally, Section 7 discusses future work.

3 Evaluation Methodology

We compare a baseline approach using a unigram model to a bigram model. We train both models on a corpus of diacritisized text, and then check the models' performance on an unseen test set, by removing the vowel diacritics from part of the corpus. For both Hebrew and Arabic, we evaluate performance by measuring the percentage of words in the test set whose vowel pattern was restored correctly, i.e. the vowel pattern suggested by the system exactly matched the original. We refer to this performance measure as *word accuracy*. For Hebrew, we also divided the vowel symbols into separate groups, each one corresponding to a specific phonetic value. We then measured the percentage of words whose individual letters were fitted with a vowel diacritic belonging to the same phonetic group as the *correct* vowel diacritic in the test set. In other words, the restored vowels, while perhaps not agreeing exactly with the original pattern, all belonged to the correct phonetic group. This performance measure, which corresponds to vocalization of non-voweled text, is useful for applications such as text-to-speech systems.[2] We refer to this performance measure as *phonetic group accuracy.*

There is an unfortunate lack of data for vowel-annotated text in both modern Hebrew and Arabic. The only easily accessible sources are the Hebrew Bible and the Qur'an, for which on-line versions transliterated into Latin characters are available. Ancient Hebrew and Arabic bear enough syntactical and semantic resemblance to their modern language equivalents to justify usage of these ancient texts as corpora. For Hebrew, we used the Westminster Hebrew Morphological Database (1998), a corpus containing a complete transcription of the graphical form of the Massoretic text of the Hebrew Bible containing roughly 300,000 words. For the Qur'an, we used the transliterated version publicly available from the sacred text archive at

[1] In literature relating to Hebrew morphology analysis, this is often refered to as a *pointed* word.

[2] In modern Hebrew, it is generally sufficient to associate each vowel symbol with its phonetic group in order to vocalize the word correctly.

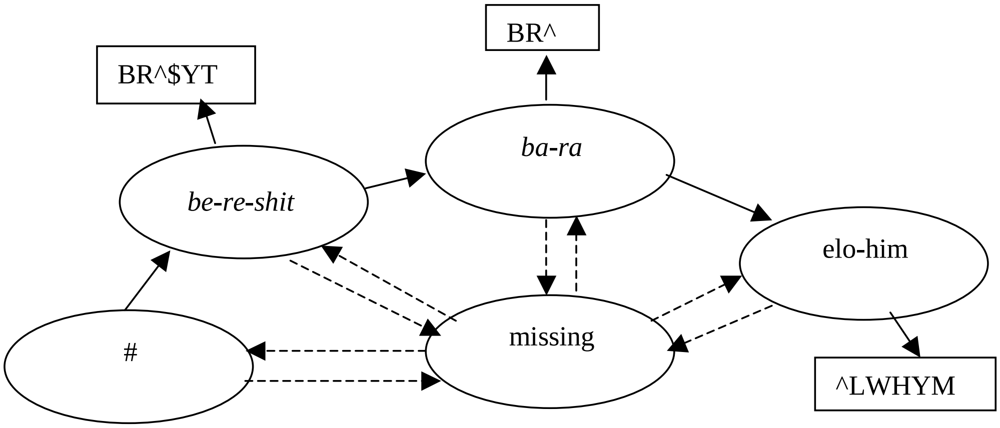


Figure 2. HMM path for the non-voweled phrase “in the beginning god created…” בראשית ברא אלוהים pronounced */be-reshit bara elohim/*

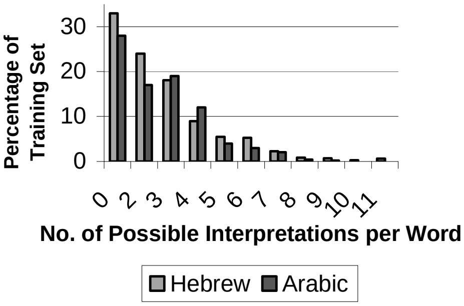
www.sacred-texts.com. This corpus contains roughly 90,000 words.

For both languages, we tested our model on 10% of the corpus. We measured performance by evaluating word accuracy for both Hebrew and Arabic. In addition, we measured phonetic group accuracy for Hebrew.

4 Baseline : A Unigram Model

To assess the difficulty of the problem, we counted the number of times each diacriticized word appeared in the training set. For each non-voweled word encountered in the test set, we searched through all of the words with the same non-voweled structure and picked the diacriticized word with the highest count in the table. Figure 1 shows the ambiguity distribution in the training set.

Figure 1. Ambiguity Distribution in Training Set



Note that for both languages, only about 30% of the words in the training set were unambiguous, i.e. had a single interpretation. For the baseline model, we achieved a word accuracy rate of 68% for Hebrew and 74% for Arabic. We note that even though the size of the Arabic training set was about a third of the size of the Hebrew training set, we still achieved a higher success rate of restoring vowels in Arabic. We attribute this to the fact that there are only three possible missing vowel diacritics in modern Arabic text, compared to twelve in Hebrew.

5 A Bigram Model

We constructed a bigram Hidden Markov Model (HMM) where hidden states were vowel-annotated (diacritisized) words, and observations were vowel-less words. One example of a path through the HMM for reconstructing a Hebrew sentence is given in Figure 2; ovals represent hidden states that correspond to diacritisized words; rectangles represent observations of vowel-less words; solid edges link the states that mark the transition through the model for generating the desired sentence; each edge carries with it a probability mass, representing the probability of transitioning between the two hidden states connected by the edge. This technique was used for Arabic in a similar way.

Our model consists of a set of hidden states $T_1,..,T_n$ where each hidden state corresponds to an observed word in our training corpus. Thus, each hidden state corresponds to a word containing its complete vowel pattern. From each hidden state T_i , there is a single emission, which simply consists of the word in its non-voweled form. If we make the assumption that the probability of observing a given word depends only on the previous word, we can compute the probability of observing a sentence $W_{1,n} = w_1,..., w_n$ by summing over all of the possible hidden states that the HMM traversed while generating the sentence, as denoted in the following equation.

$$p(W_{1,n}) = \sum_{T1,n+1} \prod_{i} p(Ti \mid Ti-1)$$

These probabilities of transitions through the states of the model are approximated by bigram counts, as described below. Note that the symbol "#" in the figure serves to "anchor" the initial state of the HMM and facilitate computation. Thereafter, the hidden states actually consist of vowel-annotated bigrams. The probability of *any* possible path in our model that generates this phrase can be computed as follows:

$$p(W_{1,n}) = \prod_{i} p(w_i \mid w_{i-1})$$

This equation decomposes into the following maximum likelihood probability estimations, denoted by $\hat{p}$, in which *c(word)* denotes the number of instances that *word* had occurred in the training set and *c(word1, word2)* denotes the number of joint occurrences of *word1* and *word2* in the training set.

$$\hat{p}(\#|\#) = 1,$$

$$\hat{p}(\# \mid be-re-shit) = c(be-re-shit),$$

$$\hat{p}(be-re-shit \mid ba-ra) = \frac{c(be-re-shit, ba-ra)}{c(ba-ra)}$$

$$\hat{p}(ba-ra \mid elo-him) = \frac{c(ba-ra, elo-him)}{c(elo-him)}$$

In order to be able to compute the likelihood of each bigram, we kept a look-up table consisting of counts for all individual and joint occurrences in the training set. We implemented the Viterbi algorithm to find the most likely path transitions through the hidden states that correspond to the observations. The likelihood of observing the sentence $W_{1,n}$ while traversing the hidden state path $T_{1,n}$ is taken to be $p(W_{1,n}, T_{1,n})$. We ignore the normalizing factor $p(W_{1,n})$. More formally, the most likely path through the model is defined as

$$\arg\max_{T1,n} pr(T_{1,n} \mid W_{1,n}) = \arg\max_{T1,n} \frac{p(W_{1,n}, T_{1,n})}{p(W_{1,n})}$$

$$= \arg\max_{T1,n} p(W_{1,n}, T_{1,n})$$

5.1 Dealing with Sparse Data

Because our bigram model is trained from a finite corpus, many words are bound to be missing from it. For example, in the unigram model, we found that as many as 16% of the Hebrew words in the test set were not present. The amount of unseen bigrams was even higher, as much as 20 percent. This is not surprising, as we expect some unseen bigrams to consist of words that were both seen before individually. We did not specifically deal with sparse data in the unigram base line model.

As many of the unseen unigrams were non ambiguous, we would have liked to look up the missing words in a vowel-annotated dictionary and copy the vowel pattern found in the dictionary. However, as noted in Section 2, morphology in both Hebrew and Arabic is non-concatenative. Since dictionaries contain only the root form of verbs and nouns, without a sound morphological analyzer we could not decipher the root. Therefore, proceeded as follows: We employed a technique proposed by Katz (1987) that combines a discounting method along with a back-off method to obtain a good estimate of unseen bigrams. We use the Good-Turing discounting method (Gale & Sampson 1991) to decide how much total probability mass to set aside for all the events we haven't seen, and a simple back-off algorithm to tell us how to distribute this probability. Formally, we define

$$\begin{cases} \hat{p}(w2 \mid w1) = P_d(w2 \mid w1) & \textit{if } c(w2,w1)>0 \\ \hat{p}(w2 \mid w1) = \alpha(w1)\, p(w2 \mid w1) & \textit{if } c(w2,w1)=0 \end{cases}$$

Here, P_d is the discounted estimate using the Good-Turing method, p is a probability estimated by the number of occurrences and $\alpha(w1)$ is a normalizing factor that divides the unknown probability mass of unseen bigrams beginning with *w1*.

$$\alpha(w1) = \frac{1 - \sum_{w2:c(w1,w2)>0} P_d(w2 \mid w1)}{1 - \sum_{w2:c(w1,w2)=0} p(w2 \mid w1)}$$

In order to compute *Pd* we create a separate discounting model for each word in the training set. The reason for this is simple: If we use only one model over all of the bigram counts, we would really be approximating $P_d(w2, w1)$. Because we wish to estimate $P_d(w2 \mid w1)$, we define the discounted frequency counts as follows:

$$c^*(w1, w2) = c(w1, w2) + \frac{n_{c(w1,w2)+1}}{n_{c(w1,w2)}}$$

where n_c is the number of different bigrams in the corpus that have frequency *c*. Following Katz, we estimate the probability of unseen bigrams to be

$$p(w2|w1) \cong \begin{cases} p(w2) & \text{if } c(w2) > 0 \\ p(unseen|w1) & \text{if } c(w2) = 0 \end{cases}$$

If the missing bigram is composed of two individually observed words, this technique allows us to estimate the probability mass of the unseen bigram. In some cases, the unseen bigram consists of individual words that have never been seen. In other words, *w2* itself is unseen and *c(w2)* cannot be computed. In this case, we estimate the probability for *p(w2|w1)* by computing *p(unseen|w1)*. We do this by allocating some probability mass to *unseen* words, keeping a special count for bigrams that were seen less then *k* times.[3] We allocate a separate hidden state for unseen words, as depicted in Figure 2. In this case, we do not attempt to fit any vowel pattern to the unseen word; the word is left bare of its diacritics. However, we can still assign a probability mass, *p(unseen|w1)*, to prevent the Viterbi algorithm from computing a zero probability. We can compute the probabilities *p(w2|unseen)* in a similar manner.

5.2 Results

Figure 3. Results of Bigram Model

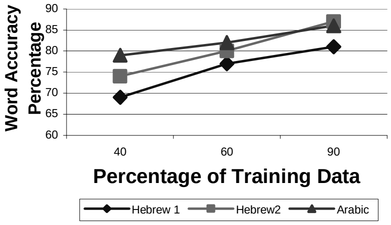


Figure 3 presents our results using the bigram HMM model, where “Hebrew 1” measures word accuracy be in Hebrew, “Hebrew 2” measures phonetic group accuracy, and “Arabic” measures word accuracy in Arabic. Using the bigram model for Hebrew, we achieved 81% word accuracy and 87% phonetic group accuracy. For Arabic, we achieved 86% word accuracy. For Hebrew, the system was more successful in restoring the phonetic group vowel pattern than restoring the exact diacritics. This is because the number of possible vowel symbols in Hebrew is larger than in Arabic. However, for text-to-speech systems, it is sufficient to associate each vowel with the correct phonetic group. For word accuracy, most of the errors in Hebrew (11%) and in Arabic (8%) were due to words that were not found in the training corpus. Therefore, we believe that acquiring a sufficiently large modern corpus of the language would greatly improve performance. However, the number of parameters for our model is quadratic in the number of word types in the training set. Therefore, we suggest using limited morphological analysis to improve performance of the system by attempting to identify the stem or root of the words in the test set, as well as the conjugation. Since conjugation templates in Semitic languages have fixed vowel patterns, even limited success in morphological analyses would greatly improve performance of the system, while not incurring a blowup in the number of parameters.

[3] *k* was arbitrarily set to three in our experiment. Alternatively, we could get a more exact estimation of the missing probability mass by discounting the unigram probabilities of *w2*.

6 Related Work

Performing a full morphological analysis of a Hebrew or Arabic sentence would greatly assist the vowel restoration problem. That is, if we could correctly parse each word in a sentence, we could eliminate ambiguity and restore the correct vowel pattern of the word according to its grammatical form and part of speech.

For Arabic, a morphological analyzer, developed by the Xerox Research Centre Europe (Beesley 1998) is freely available.[4] The system uses finite state transducers, traditionally used for modeling concatenative morphology. Since the system is word based, it cannot disambiguate words in context and outputs all possible analyses for each word. The system relies on handcrafted rules and lexicon that govern Arabic morphology.

For Hebrew, a morphological analyzer called *Nakdan Text* exists, as part of the *Rav Milim* project for the processing of modern Hebrew (Choueka and Neeman 1995). Given a sentence in modern Hebrew, *Nakdan Text* restores its vowel diacritics by first finding all possible morphological analyses and vowel patterns of every word in the sentence. Then, for every such word, it chooses the correct context-dependent vowel pattern using short-context syntactical rules as well as some probabilistic models. The authors report 95% success rate in restoring vowel patterns. It is not clear if this refers to word accuracy or letter accuracy.[5]

Segel (1997) devised a statistical Hebrew lexical analyzer that takes contextual dependencies into account. Given a non-voweled Hebrew texts as input and achieves 95% word accuracy on test data extracted from the Israeli daily *Ha'aretz*. However, this method requires fully analyzed Hebrew text to train on. Segel used a morphological hand-analyzed training set consisting of only 500 sentences. Because there is currently no tree bank of analyzed Hebrew text, this method is not applicable to other domains, such as novels or medical texts.

Kontorovich and Lee (2001) use an HMM approach to vocalizing Hebrew text. Their model consists of fourteen hidden states, with emissions for each word of the training set. Initially, the parameters of the model are chosen at random and training of the model is done using the EM algorithm. They achieve a success rate of 81%, when unseen words are discarded from the test set.

7 Future Work

Since most of the errors in the model can be attributed to missing words, we plan to address this problem from two perspectives. First, we plan to include a letter-based HMM to be used for fitting an unseen word with a likely vowel pattern. The model would be trained separately on words from the training set. Its hidden states would correspond to vowels in a language, making this model language dependent. We also plan to use a trigram model for the task of vowel restoration, backing off to a bigram model for sparse trigrams.

Second, we plan to use some degree of morphological analysis to assist us with the restoration of unseen words. At the very least, we could use a morphological analyzer as a dictionary for words that have unique diacritization, but are missing from the model. Since analyzers for Arabic that are commonly available (Beesley 1998) are word based, they output all possible morphological combinations of the word, and it is still unclear how we could choose the most likely parse given the context.

Finally, since the size of our corpora is relatively small, we also plan to use cross validation to get a better estimate of the generalization error.

8 Conclusion

In this paper, we demonstrated the use of a statistically based approach for vowel restoration in Semitic languages. We wish to demonstrate that HMMs are a useful tool for computational processing of Semitic languages, and that these models generalize to other languages. For the task of vocalizing the vowels according to their phonetic classification, the system we have proposed achieves an accuracy of 87% for Hebrew. For the task of restoring the exact vowel

[4] http://www.arabic-morphology.com/

[5] This program was demonstrated at BISFAI-95, the fifth Bar Ilan international symposium on Artificial Intelligence, but no summary or article was included in its proceedings, and to the best of our knowledge no article has been published describing the methods of *Nakdan text*.

pattern, we achieved an accuracy of 81% for Hebrew texts and 86% for Arabic texts. Thus, we have shown that the contextual information gained by using HMMs is beneficial for this task.

Acknowledgments

We would like to thank Stuart Shieber, Christian R. Lange, Jill Nickerson, Emir Kapanci, Wheeler Ruml and Chung-chieh Shan for useful discussions.

References

Academy of the Hebrew Language 1957. “The rules for Hebrew-Latin transcription,**”** *In Memiors of the Academy of the Hebrew Language*, pages 5-8 (in Hebrew)

Beesley, K. 1998. "Arabic finite-state morphological analysis and generation," *in COLING-96 Proceedings 1 : 89-94, Copenhagen.*

Charniak, E. 1995. *Statistical Language Learning*, MIT Press.

Choueka, Y. and Neeman, Y. 1995. “Nakdan-Text, (an In-Context Text-Vocalizer for Modern Hebrew),” BISFAI-95, The Fifth Bar Ilan Symposium for Artificial Intelligence

Dagan, D., Pereira P., and Lee L. 1994. “Similarity-based estimation of word cooccurrence probabilities,**”** *In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*.

Gale, W. A. and Sampson, G. 1995. “Good-Turing Frequency Estimation Without Tears,” *Journal of Quantitative Linguistics* 2, 217-237.

Katz, S., “Estimation of probabilities from sparse data for the language model component of a speech recognizer,**”** *In IEEE Transactions on Acoustics, Speech, and Signal Processin*g.

Kontorovich L.and Lee D. 2001. “Problems in Semitic NLP,” *NIPS Workshop on Machine Learning Methods for Text and Images*

Levinger, M., Ornan U., Itai A. 1995. “Learning Morpho-Lexical Probabilities from an Untagged Corpus with an Application to Hebrew, ” *Computational Linguistics*, 21(3): 383-404

Segel, A. 1997. “A probabilistic Morphological Analyzer for Hebrew undotted text,” MSc thesis, Israeli Institute of Technology, Technion. (in Hebrew)

Westminster Theological Seminar 1998. "The Hebrew Morphological Database", Philadelphia, PA