Arabic Language Modeling with Finite State Transducers

Ilana Heintz
Department of Linguistics
The Ohio State University
Columbus, OH
heintz.38@osu.edu*

Abstract

In morphologically rich languages such as Arabic, the abundance of word forms resulting from increased morpheme combinations is significantly greater than for languages with fewer inflected forms (Kirchhoff et al., 2006). This exacerbates the out-of-vocabulary (OOV) problem. Test set words are more likely to be unknown, limiting the effectiveness of the model. The goal of this study is to use the regularities of Arabic inflectional morphology to reduce the OOV problem in that language. We hope that success in this task will result in a decrease in word error rate in Arabic automatic speech recognition.

1 Introduction

The task of language modeling is to predict the next word in a sequence of words (Jelinek et al., 1991). Predicting words that have not yet been seen is the main obstacle (Gale and Sampson, 1995), and is called the Out of Vocabulary (OOV) problem. In morphologically rich languages, the OOV problem is worsened by the increased number of morpheme combinations.

Berton et al. (1996) and Geutner (1995) approached this problem in German, finding that language models built on decomposed words reduce the OOV rate of a test set. In Carki et al. (2000), Turkish words are split into syllables for language modeling, also reducing the OOV rate (but not improving WER). Morphological decomposition is also used to boost language modeling scores in Korean (Kwon, 2000) and Finnish (Hirsimäki et al., 2006).

We approach the processing of Arabic morphology, both inflectional and derivational, with finite state machines (FSMs). We use a technique that produces many morphological analyses for each word, retaining information about possible stems, affixes, root letters, and templates. We build our language models on the morphemes generated by the analyses. The FSMs generate spurious analyses. That is, although a word out of context may have several morphological analyses, in context only one such analysis is correct. We retain all analyses. We expect that any incorrect morphemes that are generated will not affect the predictions of the model, because they will be rare, and the language model introduces bias towards frequent morphemes. Although many words in a test set may not have occurred in a training set, the morphemes that make up that word likely will have occurred. Using many decompositions to describe each word sets apart this study from other similar studies, including those by Wang and Vergyri (2006) and Xiang et al. (2006).

This study differs from previous research on Arabic language modeling and Arabic automatic speech recognition in two other ways. To promote cross-dialectal use of the techniques, we use properties of Arabic morphology that we assume to be common to many dialects. Also, we treat morphological analysis and vowel prediction with a single solution.

An overview of Arabic morphology is given in Section 2. A description of the finite state machine process used to decompose the Arabic words into

*This work was supported by a student-faculty fellowship from the AFRL/Dayton Area Graduate Studies Insititute, and worked on in partnership with Ray Slyh and Tim Anderson of the Air Force Research Labs.

Proceedings of the ACL-08: HLT Student Research Workshop (Companion Volume), pages 37–42,
Columbus, June 2008.

morphemes follows in Section 3. The experimental language model training procedure and the procedures for training two baseline language models are discussed in Section 4. We evaluate all three models using average negative log probability and coverage statistics, discussed in Section 5.

2 Arabic Morphology

This section describes the morphological processes responsible for the proliferation of word forms in Arabic. The discussion is based on information from grammar textbooks such as that by Haywood and Nahmad (1965), as well as descriptions in various Arabic NLP articles, including that by Kirchhoff et al. (2006).

Word formation in Arabic takes place on two levels. Arabic is a root-and-pattern language in which many vocalic and consonantal patterns combine with semantic roots to create surface forms. A root, usually composed of three letters, may encode more than one meaning. Only by combining a root with a pattern does one create a meaningful and specific term. The combination of a root with a pattern is a *stem*. In some cases, a stem is a complete surface form; in other cases, affixes are added.

The second level of word formation is inflectional, and is usually a concatenative process. Inflectional affixes are used to encode person, number, gender, tense, and mood information on verbs, and gender, number, and case information on nouns. Affixes are a closed class of morphemes, and they encode predictable information. In addition to inflection, cliticization is common in Arabic text. Prepositions, conjunctions, and possessive pronouns are expressed as clitics.

This combination of templatic derivational morphology and concatenative inflectional morphology, together with cliticization, results in a rich variation in word forms. This richness is in contrast with the slower growth in number of English word forms. As shown in Table 1, the Arabic stem /drs/, meaning *to study*, combines with the present tense verb pattern "*CC*u*C*u", where the 'C' represents a root letter, to form the present tense stem *drusu*. This stem can be combined with 11 different combinations of inflectional affixes, creating as many unique word forms.

Table 1 can be expanded with stems from the same root representing different tenses. For instance, the stem *daras* means *studied*. Or, we can combine the root with a different pattern to obtain different meanings, for instance, to teach or to learn. Each of these stems can combine with the same or different affixes to create additional word forms.

Transliteration	Translation	Affixes
adrusu	I study	a-
nadrusu	we study	na-
tadrusu	you (ms) study	ta-
tadrusina	you (fs) study	ta- ,-ina
tadrusAn	you (dual) study	ta-, -An
tadrusun	you (mp) study	ya-, -n
tadrusna	you (fp) study	ta-, -na
yadrusu	he studies	ya-
tadrusu	she studies	ta-
yadrusan	they (dual) study	ya-, -An
yadrusun	they (mp) study	ya-, -n
yadrusna	they (fp) study	ya-, -na

Table 1: An Example of Arabic Inflectional Morphology

Adding a single clitic to the words in Table 1 will double the number of forms. For instance, the word *adrusu*, meaning *I study*, can take the enclitic 'ha', to express *I study it*. Some clitics can be combined, increasing again the number of possible word forms.

Stems differ in some ways that do not surface in the Arabic orthography. For instance, the pattern "*CC*i*C*u" differs from "*CC*u*C*u" only in one short vowel, which is encoded orthographically as a frequently omitted diacritic. Thus, *adrisu* and *adrusu* are homographs, but not homophones. This property helps decrease the number of word forms, but it causes ambiguity in morphological analyses. Recovering the quality of short vowels is a significant challenge in Arabic natural language processing.

This abundance of unique word forms in Modern Standard Arabic is problematic for natural language processing (NLP). NLP tasks usually require that some analysis be provided for each word (or other linguistic unit) in a given data set. For instance, in spoken word recognition, the decoding process makes use of a language model to predict the words that best fit the acoustic signal. Only words that have been seen in the language model's training data will be proposed. Because of the immense number of possible word forms in Arabic, it is highly proba-

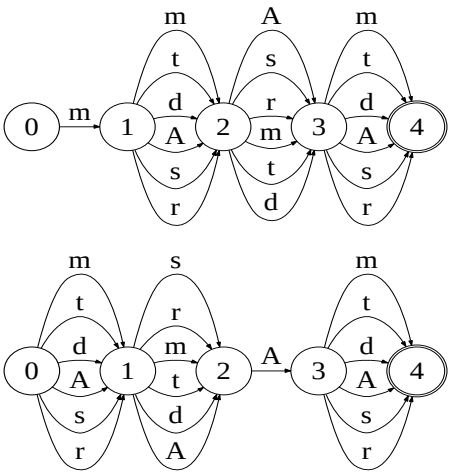

Figure 1: Two templates, m*CCC* and *CC*A*C* as finite state recognizers, with a small sample alphabet of letters A, d, m, r, s, and t.

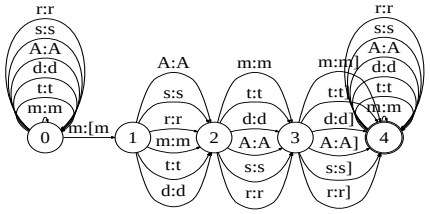

Figure 2: The first template above, now a transducer, with affixes accepted, and the stem separated by brackets in the output.

ble that the words in an acoustic signal will not have been present in the language model's training text, and incorrect words will be predicted. We use information about the morphology of Arabic to create a more flexible language model. This model should encounter fewer unseen forms, as the units we use to model the language are the more frequent and predictable morphemes, as opposed to full word forms. As a result, the word error rate is expected to decrease.

3 FSM Analyses

This section describes how we derive, for each word, a lattice that describes all possible morphological decompositions for that word. We start with a group of templates that define the root consonant positions, long vowels, and consonants for all Arabic regular and augmented stems. For instance, where *C* represents a root consonant, three possible templates are

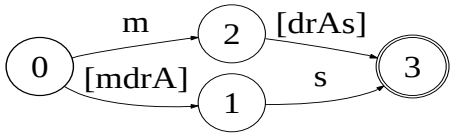

Figure 3: Two analyses of the word "mdrAs", as produced by composing a word FSM with the template FSMs above.

CCC, m*CCC*, and *CA*C*C*. We build a finite state recognizer for each of the templates, and in each case, the *C* arcs are expanded, so that every possible root consonant in the vocabulary has an arc at that position. The two examples in Figure 1 show the patterns m*CCC* and *CC*A*C* and a short sample alphabet.

At the start and end node of each template recognizer, we add arcs with self-loops. This allows any sequence of consonants as an affix. To track stem boundaries, we add an open bracket to the first stem arc, and a close bracket to the final stem arc. The templates are compiled into finite state transducers. Figure 2 shows the result of these additions.

For each word in the vocabulary, we define a simple, one-arc-per-letter finite state recognizer. We compose this with each of the templates. Some number of analyses result from each composition. That is, a single template may not compose with the word, may compose with it in a unique way, or may compose with the word in several ways. Each of the successful compositions produces a finite state recognizer with brackets surrounding the stem. We use a script to collapse the arcs within the stem to a single arc. The result is shown in Figure 3, where the word "mdrAs" has two analyses corresponding to the two templates shown. We store a lattice as in Figure 3 for each word.

The patterns that we use to constrain the stem forms are drawn from Haywood and Nahmad (1965). These patterns also specify the short vowel patterns that are used with words derived from each pattern. An option is to simply add these short vowels to the output symbols in the template FSTs. However, because several short vowel options may exist for each template, this would greatly increase the size of the resulting lattices. We postpone this effort. In this work, we focus solely on the usefulness of the unvoweled morphological decompositions.

We do not assess or need to assess the accuracy of

the morphological decompositions. Our hypothesis is that by having many possible decompositions per word, the frequencies of various affixes and stems across all words will lead the model to the strongest predictions. Even if the final predictions are not prescriptively correct, they may be the most useful decompositions for the purpose of speech decoding.

4 Procedure

We compare a language model built on multiple segmentations as determined by the FSMs described above to two baseline models. We call our experimental model FSM-LM; the baseline models use word-based n-grams (WORD), and pre-defined affix segmentations (AFFIX). Our data set in this study is the TDT4 Arabic broadcast news transcriptions (Kong and Graff, 2005). Because of time and memory constraints, we built and evaluated all models on only a subsection of the training data, 100 files of TDT4, balanced across the years of collection, and containing files from each of the 4 news sources. We use 90 files for training, comprising about 6.3 million unvoweled word tokens, and 10 files for testing, comprising about 700K word tokens, and around 5K sentences. The size of the vocabulary is 104757. We use ten-fold cross-validation in our evaluations.

4.1 Experimental Model

We extract the vocabulary of the training data, and compile the word lattices as described in Section 3. The union of all decompositions (a lattice) for each individual word is stored separately.

For each sentence of training data, we concatenate the lattices representing each word in that sentence. We use SRILM (Stolcke, 2002) to calculate the posterior expected n-gram count for morpheme sequences up to 4-grams in the sentence-long lattice. The estimated frequency of an n-gram N is calculated as the number of occurrences of that n-gram in the lattice, divided by the number of paths in the lattice. This is true so long as the paths are equally weighted; at this point in our study, this is the case.

We merge the n-gram counts over all sentences in all of the training files. Next, we estimate a language model based on the n-gram counts, using only the 64000 most frequent morphemes, since we expect this vocabulary size may be a limitation of our ASR system. Also, by limiting the vocabulary size of all of our models (including the baseline models described below), we can make a fairer comparison among the models. We use Good-Turing smoothing to account for unseen morphemes, all of which are replaced with a single "unknown" symbol.

In later work, we will apply our LM statistics to the lattices, and recalculate the path weights and estimated counts. In this study, the paths remain equally weighted.

We evaluate this model, which we call FSM-LM, with respect to two baseline models.

4.2 Baseline Models

For the WORD model, we do no manipulation to the training or test sets beyond the normalization that occurs as a preprocessing step (hamza normalization, replacement of problematic characters). We build a word-based 4-gram language model using the 64000 most frequent words and Good-Turing smoothing.

For the AFFIX model, we first define the character strings that are considered affixes. We use the same list of affixes as in Xiang et al. (2006), which includes 12 prefixes and 34 suffixes. We add to the lists all combinations of two prefixes and two suffixes. We extract the vocabulary from the training data, and for each word, propose a single segmentation, based on the following constraints:

1. If the word has an acceptable prefix-stem-suffix decomposition, such that the stem is at least 3 characters long, choose it as the correct decomposition.
2. If only one affix is found, make sure the remainder is at least 3 characters long, and is not also a possible affix.
3. If the word has prefix-stem and stem-suffix decompositions, use the longest affix.
4. If the longest prefix and longest suffix are equal length, choose the prefix-stem decomposition.

We build a dictionary that relates each word to a single segmentation (or no segmentation). We segment the training and test texts by replacing each word with its segmentation. Morphemes are separated by whitespace. The language model is built by counting 4-grams over the training data, then using only the most frequent 64000 morphemes in estimating a language model with Good-Turing smoothing.

	WORD	AFFIX	FSM-LM
Avg Neg Log Prob	4.65	5.30	4.56
Coverage (%):			
Unigram	96.03	99.30	98.89
Bigram	17.81	53.13	69.56
Trigram	1.52	11.89	27.25
Four-gram	.37	3.42	9.62

Table 2: Average negative log probability and coverage results for one experimental language model (FSM-LM) and two baseline language models. Results are averages over 10 folds.

5 Evaluation

For each model, the test set undergoes the same manipulation as the train set; words are left alone for the WORD model, split into a single segmentation each for the AFFIX model, or their FSM decompositions are concatenated.

Language models are often compared using the perplexity statistic:

$$PP(x_1 \dots x_n) = 2^{-\frac{1}{n}\sum_{x_i=4}^{n} logP(x_i|x_{i-1}^{i-3})} \quad (1)$$

Perplexity represents the average branching factor of a model; that is, at each point in the test set, we calculate the entropy of the model. Therefore, a lower perplexity is desired.

In the AFFIX and FSM-LM models, each word is split into several parts. Therefore, the value $\frac{1}{n}$ would be approximately three times smaller for these models, giving them an advantage. To make a more even comparison, we calculate the geometric mean of the n-gram transition probabilities, dividing by the number of *words* in the test set, not morphemes, as in Kirchhoff et al. (2006). The log of this equation is:

$$AvgNegLogProb(x_1 \dots x_n) = -\frac{1}{N}\sum_{i=4}^{n} logP(x_i|x_{i-1}^{i-3}) \quad (2)$$

where n is the number of morphemes or words in the test set, depending on the model, and N is the number of *words* in the test set, and $log\ P(x_i|x_{i-1}^{i-3})$ is the log probability of the item x_i given the 3-item history (calculated in base 10, as this is how the SRILM Toolkit is implemented). Again, we are looking for a low score.

In the FSM-LM, each test sentence is represented by a lattice of paths. To determine the negative log probability of the sentence, we score all paths of the sentence according to the equations above, and record the maximum probability. This reflects the likely procedure we would use in implementing this model within an ASR task.

We see in Table 2 that the average negative log probability of the FSM-LM is lower than that of either the WORD or AFFIX model. The average across 10 folds reflects the pattern of scores for each fold. We conclude from this that the FSM model of predicting morphemes is more effective than - or more conservatively, at least as effective as - a static decomposition, as in the AFFIX model. Furthermore, we have successfully reproduced the results of Xiang et al. (2006) and Kirchhoff et al. (2006), among others, that modeling Arabic with morphemes is more effective than modeling with whole word forms.

We also calculate the coverage of each model: the percentage of units in the test set that are given probabilities in the language model. For the FSM model, only the morphemes in the best path are counted. The coverage results are reported in Table 2 as the average coverage over the 10 folds. Both the AFFIX and FSM-LM models showed improved coverage as compared to the WORD model, as expected. This means that we reduce the OOV problem by using morphemes instead of whole words. The AFFIX model has the best coverage of unigrams because only new stems, not new affixes, are proposed in the test set. That is, the same fixed set of affixes are used to decompose the test set as the train set, however, unseem stems may appear. In the FSM-LM, there are no restrictions on the affixes, therefore, unseen affixes may appear in the test set, as well as new stems, lowering the unigram coverage of the test set. For larger n-grams, however, the FSM-LM model has the best coverage. This is due to keeping all decompositions until test time, then allowing the language model to define the most likely sequences, rather than specifying a single decomposition for each word.

A 4-gram of words will tend to cover more context than a 4-gram of morphemes; therefore, the word 4-grams will exhibit more sparsity than the morpheme 4-grams. We compare, for a single train-

	WORD	AFFIX	FSM-LM
unigrams	4.97	5.84	5.60
bigrams	4.95	5.70	4.61
trigrams	4.95	5.69	4.56
four-grams	4.95	5.69	4.57

Table 3: Comparison of n-gram orders across language model types.

test fold, how lower order n-grams compare among the models. The results are shown in Table 3. We find that for lower-order n-grams, the word model performs best. As the n-grams get larger, the sparsity problem favors the FSM-LM, which has the best overall score of all models shown. Apparently, the frequencies of 3- and 4-grams are not big enough to make a big difference in the evaluation. This is likely due to the small size of our corpus, and we expect the result would change if we were to use all of the TDT4 corpus, rather than a 100 file portion of the corpus.

6 Conclusion & Future Work

It has been shown that reduced perplexity scores do not necessarily correlate with reduced word error rates in an ASR task (Berton et al., 1996). This is because the perplexity (or in this case, average negative log probability) statistic does not take into account the acoustic confusability of the items being considered. However, the average negative log probability score is a useful tool as a proof-of-concept, giving us reason to believe that we may be successful in implementing this model within an ASR task.

The real test of this model is its ability to predict short vowels. The average negative log probability scores may lead us to believe that the FSM-LM is only marginally better than the WORD or AFFIX model, and the differences may not be apparent in an ASR application. However, only the FSM-LM model allows for the opportunity to predict short vowels, by arranging the FSMs as finite state transducers with short vowel information encoded as part of the stem patterns.

We will continue to tune the language model by applying the language model weights to the decomposition paths and re-estimating the language model. Also, we will expand the language model to include more training data. We will implement the model within an Arabic ASR system, with and without short vowel hypotheses. Furthermore, we are interested to see how well the application of these templates and this framework will apply to other Arabic dialects.

References

A Berton, P Fetter, and P Regel-Brietzmann. 1996. Compound words in large vocabulary German speech recognition system. In *Proceedings of ICSLP 96*, pages 1165–1168.

K. Carki, P. Geutner, and T. Schultz. 2000. Turkish lvcsr: Towards better speech recognition for agglutinative languages. In *ICASSP 2000*, pages 134–137.

William A. Gale and Geoffrey Sampson. 1995. Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 22:217–37.

P Geutner. 1995. Using morphology towards better large-vocabulary speech recognition systems. In *Proceedings of ICASSP-95*, volume 1, pages 445–448.

J.A. Haywood and H.M. Nahmad. 1965. *A New Arabic Grammar of the Written Language*. Lund Humphries, Burlington, VT.

Teemu Hirsimäki, Mathias Creutz, Vesa Siivola, Mikko Kurimo, Sami Virpioja, and Janne Pylkkönen. 2006. Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer Speech and Language*, 20:515–541.

F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss. 1991. A dynamic language model for speech recognition. In *Proc. Wkshp on Speech and Natural Language*, pages 293–295, Pacific Grove, California. ACL.

Katrin Kirchhoff, Dimitra Vergyri, Kevin Duh, Jeff Bilmes, and Andreas Stolcke. 2006. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech and Language*, 20(4):589–608.

Junbo Kong and David Graff. 2005. TDT4 multilingual broadcast news speech corpus.

Oh-Wook Kwon. 2000. Performance of LVCSR with morpheme-based and syllable-based recognition units. In *Proceedings of ICASSP '00*, volume 3, pages 1567–1570.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado.

Wen Wang and Dimitra Vergyri. 2006. The use of word n-grams and parts of speech for hierarchical cluster language modeling. In *Proceedings of ICASSP 2006*, pages 1057–1060.

Bing Xiang, Kham Nguyen, Long Nguyen, Richard Schwartz, and John Makhoul. 2006. Morphological decomposition for Arabic broadcast news transcription. In *Proc. ICASSP 2006*, pages 1089–1092.