Morphology-Based Language Modeling for Conversational Arabic Speech Recognition

Katrin Kirchhoff[a,*], Dimitra Vergyri[b], Jeff Bilmes[a],
Kevin Duh[a], Andreas Stolcke[b]

[a] *Department of Electrical Engineering, University of Washington, Box 352500, Seattle, WA, 98195-2500, USA*

[b] *SRI International, Menlo Park, California, 94720, USA*

Abstract

Language modeling for large-vocabulary conversational Arabic speech recognition is faced with the problem of the complex morphology of Arabic, which increases the perplexity and out-of-vocabulary rate. This problem is compounded by the enormous dialectal variability and differences between spoken and written language. In this paper we investigate improvements in Arabic language modeling by developing various morphology-based language models. We present four different approaches to morphology-based language modeling, including a novel technique called factored language models. Experimental results are presented for both rescoring and first-pass recognition experiments.

Key words: language modeling, morphology, speech recognition, Arabic

1 Introduction

In the past decade, research on large-vocabulary conversational speech recognition (LVCSR) has been extended from a small number of 'mainstream' languages like English, German, and French, to an increasingly wider range of languages, e.g. Chinese, Turkish, or Serbo-Croatian. It has been shown that many core speech recognition techniques (such as channel normalization or

[*] Corresponding author.
Email addresses: katrin@ee.washington.edu (Katrin Kirchhoff), dverg@speech.sri.com (Dimitra Vergyri), bilmes@ee.washington.edu (Jeff Bilmes), duh@ee.washington.edu (Kevin Duh), stolcke@speech.sri.com (Andreas Stolcke).

speaker adaptation) are largely language-independent (Schwartz et al., 2004). However, porting automatic speech recognition (ASR) systems to new languages may also highlight unsolved modeling problems. One such problem is posed by complex morphology, i.e. productive word formation processes (inflection, derivation, compounding) which create a large number of possible word forms in a given language. As a result, statistical language modeling becomes increasingly difficult, due to high out-of-vocabulary (OOV) rates and high perplexity.

Arabic is one of the languages that are often described as morphologically complex. In addition, the problem of language modeling for Arabic is compounded by extreme dialectal variation and significant differences between the spoken and the written language. In particular, most varieties of Arabic are essentially spoken dialects for which generally accepted written standards do not exist. This makes it difficult to collect large amounts of text data to improve language models for conversational Arabic speech. In this paper we describe a range of morphology-based language models for Arabic that exploit sparse training data in a more efficient way and offer the potential of data-sharing across different dialects. All of these models are based on the decomposition of word forms into smaller morphological components. This includes a standard linear decomposition into morphs or particles, as used previously in other languages (e.g. Whittaker (2000)), but also a new type of parallel decomposition resulting in a so-called *factored language model*. The latter provides more robust probability estimates for word n-grams by employing a backoff procedure which utilizes information from additional word features, such as morphological tags. It is thus particularly suited to Arabic and other morphologically rich languages, but it can also be used as a more general framework for incorporating additional information sources into statistical language modeling. We explore the use of these models in both first-pass recognition and rescoring experiments and report speech recognition results on Egyptian Colloquial Arabic.

The remainder of this paper is structured as follows: Section 2 describes the linguistic properties of Arabic and the resulting problems for ASR in greater detail. Section 3 describes the corpus used for the present study. Section 4 explains the various morphological modeling approaches. The recognition system and recognition results obtained by using morphological language models in rescoring experiments are described in Sections 5 and 6, respectively. Sections 7 and 8 describe results obtained by more advanced modeling techniques, in particular automatic parameter search in factored language models, and the use of these models in first-pass recognition. Section 9 provides a discussion of the results.

2 Linguistic Properties of Arabic

Arabic is part of the Semitic language family and serves as the official language in more than 22 countries. Rather than being a single homogeneous language, however, it is more properly described as a collection of different dialects or varieties. The most widely encountered variety is Modern Standard Arabic (MSA), which is used for written as well as formal oral communication (e.g. in news broadcasts, official speeches, etc.) and is understood by educated speakers throughout the Arabic-speaking world. Everyday informal communication, by contrast, is carried out in a local dialect. The differences among local dialects are considerable and affect pronunciation, phonology, vocabulary, morphology, and syntax. Widely differing dialects (e.g. Moroccan Arabic and the Iraqi dialect) may hinder communication to the extent that speakers choose to use MSA as a common language. Table 1 lists examples of some differences between Egyptian Colloquial Arabic (ECA) and MSA.

Change	MSA	ECA	Gloss
/*θ*/ → /s/,/t/	/*θ*ala:*θ*a/	/tala:ta/	ثلَاث *three*
/ð/ → /z/,/d/	/ðahab/	/dahab/	ذهب *gold*
/ay/ → /e:/	/s̶aif/	/s̶e:f/	صيف *summer*
inflections	yatakallam(u)	yitkallim	يتكلم ’he speaks’
vocabulary	Tawila	tarabeeza	*table*
word order	VSO	SVO	

Table 1
Linguistic differences between MSA and ECA.

Only MSA has a universally agreed-upon writing standard; Arabic dialects are spoken rather than written varieties. If speakers do attempt to write dialectal speech, the MSA writing system is typically used, which consists of twenty-eight letters (twenty-five consonants and three long vowels). A distinguishing feature of this system is that short vowels are not represented by the letters of the alphabet but by *diacritics*, short strokes placed either above or below the preceding consonant. Several other phenomena are marked by diacritics, such as consonant doubling and word-final adverbial markers. Arabic texts are usually not fully diacritized, which leads to considerable lexical ambiguity and, as a consequence, increased language model perplexity. In ASR, this problem can be circumvented by using grapheme-based acoustic models (Billa et al., 2002) or automatic diacritization (Kirchhoff and Vergyri, 2004). For the present study we use a fully transcribed corpus which includes diacritic information; this problem is therefore not further investigated here.

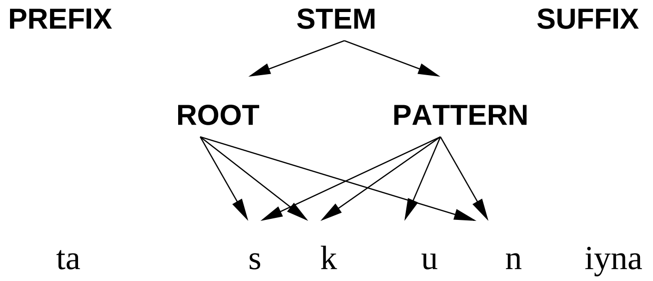

Fig. 1. Morphological structure for *taskuniyna (you (f.sg.) live).*

The morphological structure of Arabic open-class words is that of a stem surrounded by affixes signaling grammatical categories such as person, number and gender. The stem can further be decomposed into a *root* (sequence of three consonants) and a *pattern* of vowels and, possibly, additional consonants. The root and the pattern are interspersed according to the possible consonantal slots specified in the pattern. The root assigns the basic meaning to a word, e.g. the root *d-r-s*, indicates the basic meaning of 'study', *k-t-b* has the basic meaning of 'write', etc. It never occurs on its own but only in combination with the pattern, whose vowel and consonant slots indicate secondary grammatical features like voice, tense, or causality. Table 2 lists examples of words derived from the same root and different patterns.[1] Arabic has approximately 5000 roots, several hundred patterns, and dozens of affixes. Although not all of them can combine freely, the resulting number of possible word forms is enormous. In addition, a small number of particles can attach to the beginning of the word, thus increasing the number of word forms even further. Compared to MSA, dialectal Arabic exhibits some morphological simplifications, but word formation is still very complex, as the following analysis demonstrates.

Root KTB	Root DRS
ka**t**a**b**a - he wrote	**d**a**r**a**s**a - he studied
ki**t**aa**b** - book	**d**a**rs** - lesson
ku**t**u**b** - books	**d**u**r**uu**s** - lesson
'a**kt**u**b**u - I write	'a**dr**u**s**u - I study
ma**kt**a**b** - desk, office	ma**dr**a**s**a - school
ma**kt**a**b**a - library	mu**d**a**rr**i**s** - teacher
kaa**t**i**b** - writer	**d**a**rr**a**s**a - he taught

Table 2
Words derived from the roots KTB ('write') and DRS ('study'). Root consonants are marked in boldface.

Highly-inflected languages typically exhibit a large number of word types rel-

[1] For a more extensive overview of Arabic morphology, see e.g. (Schulz et al., 2000).

ative to the number of tokens in a given text. Figure 2 shows a comparison of the vocabulary growth rates (the increase in number of word types vs. number of word tokens for a given text) for English and ECA, which were calculated in each case from 100K words of the English and ECA CallHome corpora available from the Linguistic Data Consortium (LDC). The vocabulary growth rate of ECA exceeds that of English significantly. Figure 3 shows the vocabulary growth rates for the stemmed versions of the same texts. The Arabic text was stemmed by looking up the stem for each word in the CallHome ECA lexicon distributed with the corpus; the English text was stemmed by the Porter stemmer (Porter, 1980). In both cases a reduction in vocabulary growth can be observed. In Arabic, however, the reduction is much greater than in English. This demonstrates that the vocabulary growth rate in Arabic is indeed primarily caused by the multiplication of word forms through morphological affixation. The large number of actual and possible word forms makes it difficult to robustly estimate statistical languages models: many word combinations are observed only infrequently or not at all, leading to high perplexity and a large out-of-vocabulary rate. This is a particularly severe problem for language modeling of Arabic conversational (i.e. dialectal) speech. Since Arabic dialects are essentially spoken languages, very few dialectal resources exist; the only corpus currently available is the LDC ECA corpus, though further data collection efforts are under way. It was shown in (Kirchhoff et al., 2002) that adding MSA data to language model training data for ECA did not improve the ECA language model, due to the considerable differences between the two varieties. This includes the linguistic differences described above as well differences in topic structure and style, which are caused by the fact that

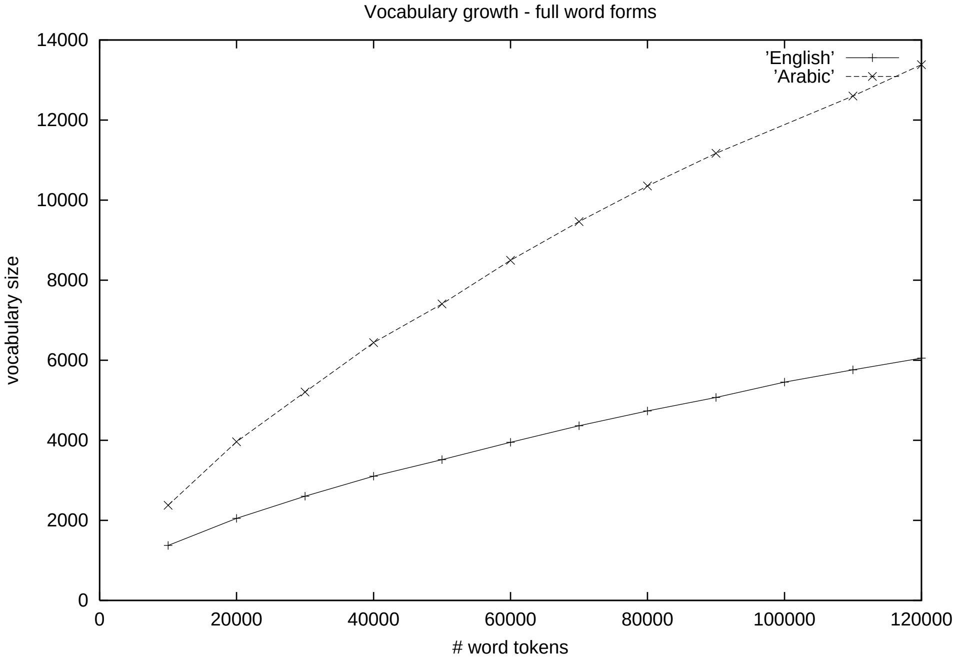

Fig. 2. Vocabulary growth rates for CallHome data in English and ECA.

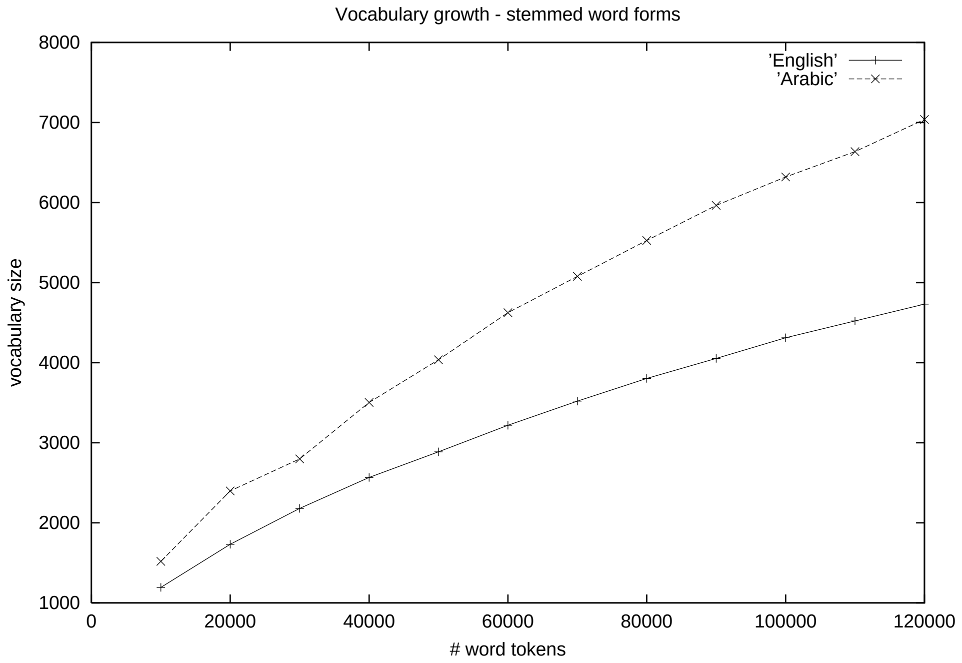

Fig. 3. Vocabulary growth rates for stemmed CallHome data in English and ECA.

MSA and ECA are used in almost complementary situations. Although some gain could potentially be obtained from first transforming MSA text to make it more similar in style to ECA speech, such a technique has not yet been tested, leaving transcriptions of actual recorded speech as the only available training data. For this reason it would be desirable to find a way of exploiting language modeling training data in a more efficient way, i.e. by decomposing word forms into their morphological components and sharing training data across words.

3 Data

For the present study we used the LDC CallHome (CH) corpus of Egyptian Colloquial Arabic. This corpus is a collection of informal phone conversations between close friends or family members, with one speaker being located in the U.S. and the other being located in Egypt. The majority of speakers come from the Cairene dialect region. The corpus consists of the data sets shown in Table 3. Two different sets of transcriptions are provided: the script form and a "romanized" form containing the phonetic information absent from the script form. Two lexicons of around 16K and 61K, respectively, are available for this corpus. They contain information about the script and romanized form, the pronunciation, the stem and morphological class of the word, and word frequency information. The data is characterized by a relatively high degree of disfluencies (9% of all word tokens), foreign (in particular English) words

Set	# conversations	# words	# hrs
train	80	146,298	14
dev	20	32,148	3.5
eval96	20	15,584	1.5
eval97	20	17,670	1.8
h5_new	20	16,752	1.8
eval03	10	11,015	1.9

Table 3
LDC ECA CH data sets.

(1.6%), and noise events. Table 3 lists the standard division of the corpus into subsets. For our recognition experiments we use the combined training, h5_new and eval96 sets for training, the dev set for development and the eval97 and eval03 sets for evaluation. The rate of out-of-vocabulary words on these sets is around 5%. It should be noted that while the eval97 set is similar to the dev set, the eval03 differs from both. In addition to its smaller size and vocabulary, it contains 30% more word fragments and overlaps much more with the training set in terms of n-gram coverage – this is explained further in Sections 4 and 8 below.

4 Morphology-Based Language Models

Standard statistical language models compute the probability of a word sequence $W = w_w, w_2, ..., w_N$ as a product of the conditional probabilities of individual words given their histories. Typically, histories are approximated by one or two preceding words, resulting in bigrams or trigrams, respectively. A trigram is expressed as:

$$p(w_1, ..., w_N) \approx \prod_{i=3}^{N} p(w_i|w_{i-1}, w_{i-2}) \tag{1}$$

The quality of a language model is usually measured as its perplexity with respect to an evaluation text. The definition of perplexity is given below for the trigram case:

$$PP(w_1, ..., w_N) = 2^{\frac{1}{N}\sum_{i=3}^{N} log(p(w_i|w_{i-1},w_{i-2}))} \tag{2}$$

Morphology-based language models typically do not compute probabilities over words but over some decomposed word representation. Various morphology-based language models have been explored in the past for languages other than Arabic, e.g. German (Geutner, 1995), Turkish (Çarki et al., 2000), Korean Kiecza et al. (1999), Russian (Whittaker, 2000) and Czech (Byrne et al., 2001). We briefly review these approaches before presenting the different language models developed for our task.

4.1 Previous Work

Early approaches to exploiting morphological information in language model are reported in (El-Bèze and Derouault, 1990; Cerf-Dannon and El-Bèze, 1991), where a class-based model using parts-of-speech and word stems as classes was described for French. Initial experiments on 115K-word language modeling task yielded a perplexity reduction from 280 to 239; a word recognition experiment on a 20K-word recognition task showed a word error rate reduction from 5% to 4.6%. In (Geutner, 1995), language models for German were constructed by decomposing full word forms into root forms. When using the resulting roots in the language model, a reduction in trigram perplexity from 67 to 59 but an decrease in root form accuracy (compared to the word-based baseline) from 66.2% to 63.5% was observed. In (Kiecza et al., 1999), word decomposition for Korean ASR was investigated. Elementary syllable units were combined in a data-driven way to form units intermediate between syllables and words. The OOV rate was reduced from 41% to less than 1%, syllable accuracy was improved from 63% to 69%, and the relative improvement in lattice word accuracy of from 75% to 83%, compared to a syllable-based baseline. However, no improvement was obtained in actual one-best recognition. In (Çarki et al., 2000) a recognizer for Turkish was built by automatically decomposing words into morpheme-sized "chunks". Chunks were then clustered according to their position in the word, and a 4-gram model over chunks was used as a language model. The chunk units were used both in the acoustic component and in the language modeling component. Although the OOV rate showed a decrease from 15% to 7%, there was an increase in word error rate from 34.1% to 37%, for a 30K-word recognition task. Whittaker (2000) developed a so-called particle model which was tested on Russian and English. This model assumes that a word is unambiguously decomposed into a number of particles. An n-gram model over particles then computes the probability of a particle given its history of $n-1$ particles, which may span word boundaries. Three different ways of decomposing words were compared, including both knowledge-based and data-driven methods. In all cases, n-gram models up to an order of $n = 6$ were used. Perplexity reductions of 5.4% (for English) and 7.5% (for Russian) were obtained but no ASR results were reported. In (Byrne et al., 2001) a morphologically-based language model for Czech was used in a large-vocabulary continuous speech recognition system. Words were decomposed into stems and affixes, and a morpheme bigram model was used. No word error rate improvements were obtained compared to the word-based baseline system. Some rudimentary morphological processing was also applied

	dev	eval97	eval03
bigram coverage	48.1%	49.4%	58.6%
bigram ppl	230	227	132
trigram-I coverage	16.9%	17.2%	39.0%
trigram-I ppl	227	222	123
trigram-II coverage	22.8%	23.3%	52.2%
trigram-II ppl	179	156	128

Table 4
Perplexities (ppl) obtained by word-based models and n-gram token coverage rates on the CH development and evaluation sets. Trigram-I refers to a trigram trained on data processed with method I; trigram-II is a trigram trained on data processed by method II.

to Arabic in (Billa et al., 1997). The definite article in ECA, *il*, which always attaches to the following noun, was separated from its successor and treated as a separate word, both for acoustic modeling and for language modeling. This reduced the vocabulary size by 7% (relative) and the word error rate by 1% (absolute), for a baseline word error rate of 73%.

4.2 Models for Arabic

In order to be able to compare the properties of morphology-based language models with those of standard word-based models, we first describe the baseline language models trained for our task. Word-based bigrams and trigrams were trained on differently pre-processed versions of the training data. Processing method I preserves all foreign words, hesitations and fragments as individual items in the language models since these are also individual entries in the dictionary used for first-pass recognition and lattice generation. Bigrams and trigrams trained on this representation are used for the generation of first-pass hypotheses and word lattices generation (see further Section 5). Processing method II conflates all foreign words, hesitations, and fragments in to three broad classes (FOREIGN, FILLER, FRAGMENT). Models trained on this representation are used at later stages in the recognition system (in particular N-best rescoring), where a match with acoustic models is not required. The perplexities of these models on the CH development and evaluation sets are shown in Table 4. We also show the percentage of bigram and trigram tokens covered by the language model and notice that the eval03 set has a higher degree of overlap with the training set, which is responsible for the lower perplexity.

Morphology-based language models for Arabic can be developed by exploiting

linguistic knowledge about different word components, i.e. stems, affixes, roots and patterns. We have investigated four different ways of using this information. These include
1. particle models similar to the work of Whittaker (2000) described above;
2. class-based models where classes are defined by morphological components;
3. single-stream models where sequences of stems, morph tags, etc. are considered individually;
4. a new type of language model called *factored language model*, which uses morphological information in a novel backoff procedure.

Affix	Function	Affix	Function
-i	1st sg poss	-ni	1st sg dir
-li	1st sg ind	-na	1st pl poss/dir/ind
-ik	2nd sg fem poss	-lik	2nd sg fem dir/ind
-ak	2nd sg masc poss	-lak	2nd sg masc dir/ind
-ha	3rd sg fem poss	-l(a)ha	3rd sg fem dir/ind
-hu	3rd sg masc poss	-lhu	3rd sg masc dir/ind
-ku(m)	2nd pl poss/dir/ind	-hum	3rd pl poss/dir/ind
Il-	definite article	bi-	preposition *in*
fi-	preposition *in*	li-	preposition *in order to, for*
fa-	conjunction *so, and*	ka-	preposition *like*
ma-	negation particle	Ha-	future tense particle

Table 5
Affixes used for word decomposition. Sg = singular, pl = plural, poss = possessive, dir = direct object, ind = indirect object.

4.2.1 Particle model

Since a rudimentary morphological decomposition (i.e. separation of the definite article *il* from the following noun) showed promising results in earlier work on Arabic ASR (Billa et al., 1997), we assumed that a higher degree of decomposition would be beneficial. Thus, in addition to separating the definite article from its successor, several other particles and affixes were separated from their word stems. We used the information in the CH ECA lexicon combined with knowledge of Egyptian Arabic morphology (see e.g. (Abdel-Massih, 1975)) to identify recurrent affixes. Those affixes which were used in the final decomposition are the possessive and object pronoun suffixes, negation and future tense markers, and prepositional particles (see Table 5). The following are examples of decomposed sequences:

(1) Hayibqu yacni Hatibqa il+nAs kullaha

	# unique items	2-gram	3-gram	4-gram
particle model	49,256	227	215	217

Table 6
Number of unique lexical items and perplexities on the CH dev set obtained by particle-based language models.

⇒ Ha+ yibqu yacni Ha+ tibqa il+ nAs kulla +ha

(2) akallimak yOm il+Hadd fa+ana baqa baHAwil a$raH+lu
⇒ akallim +ak yOm il+ Hadd fa+ ana baqa baHAwil a$raH +lu

Word and particle representations can be converted to each other unambiguously. Table 6 shows the number of unique lexical items obtained by decomposing the word-based lexicon into particles. Our baseline lexicon consists of a combination of the word forms in the two different CH lexicons mentioned above and contains 54,545 entries.

A language model trained on this representation models statistical regularities governing sequences of stems and affixes rather than sequences of words. N-grams up to an order of 4 were trained on this representation. Their perplexities were measured as shown in Equation 3 (for trigrams)

$$PP(w_1, ..., w_N) = 2^{-\frac{1}{N}\sum_{i=1}^{M} log(P(part_i|part_{i-1},part_{i-2}))} \quad (3)$$

where N is the number of words and M is the number of particles into which the word stream has been decomposed. Note that the log probability is accumulated over particles but the normalization factor is still the number of words, not the number of particles. This is done in order to compensate for the effect that perplexity tends to be lower for a text containing more individual units, since the sum of log probabilities is divided by a larger denominator.

N-gram models were trained on text preprocessed according to method II, using modified Kneser-Ney smoothing with interpolation of higher-order and lower-order probabilities. The perplexities with respect to the development set are shown in Table 6. Although we see a significant perplexity increase compared to the corresponding word-based trigram (see Table 4), the model may provide complementary information and is therefore used in rescoring experiments described below in Section 6.

4.2.2 Class-based models

The third type of morphology-based language model is a class-based model of the type initially described by (Brown et al., 1992) and shown below in Equation 4. The class c is defined by either the stem, root, pattern, or morph

	stems	morphs	roots	patterns
PP	159.1	275.8	265.6	302.1

Table 7
Perplexities of class-based models on the CH dev set. Classes are defined by stems, morph tags, roots, and patterns, respectively.

class obtained from the morphological decomposition as described above.

$$p(w_i|w_{i-1}) = \sum_{c_i,c_{i-1}} p(w_i|c_i)p(c_i|c_{i-1})p(c_{i-1}|w_{i-1}) \tag{4}$$

The perplexities associated with the various class-based models are shown in Table 7.

4.2.3 Stream models

Instead of splitting words into sequences of morphs or particles, words can also be conceived of as bundles of *simultaneous* morphological features. For example, the word *calls* could be analyzed as consisting of a stem *CALL* plus the morphological feature [noun plural] or [verb 3rd person singular], depending on the interpretation. The structure of Arabic words is even richer (cf. Section 2), suggesting a decomposition into stems, morphological affixes, roots, and patterns. A sequence of words can thus be represented as a sequence of word feature vectors; a sequence of individual vector components defines a feature stream. These individual streams can be used as alternative information sources in language modeling. Streams can either be used separately, or variables in one stream can be predicted from variables in another stream, thus taking into account cross-stream dependencies. Our approach is to train standard trigram models for each stream. For instance, given a sequence of stems $s_1, ..., s_N$, the corresponding trigram model is

$$p(s_1, ..., s_N) \approx \prod_{i=3}^{N} p(s_i|s_{i-1}, s_{i-2}) \tag{5}$$

In order to obtain multiple morphological streams, words were decomposed based on information provided in the CH ECA lexicon, which provides the stem and the morphological class for each word. The latter is defined by a combination of grammatical features such as part-of-speech, number, gender, tense, etc. The stem can be further decomposed into the root and the pattern. Since root and pattern information was not included in the lexicon, we used an automatic morphological analyzer (Darwish, 2002), which extracts roots from stems. The pattern was then obtained by subtracting the root provided by the automatic analyzer from the stem. It should be noted that the analyzer was developed for Modern Standard Arabic; it therefore does not provide accurate stem-to-root conversions for all ECA forms, such that both the root and the

	Word	Stems	Morph Classes	Roots	Patterns
# types	54,545	37,325	1,360	7,294	4,724
PP	179	125.3	50.0	76.8	52.8

Table 8
Number of unique words and factors in the CH lexicon and perplexities of the corresponding trigram models on the CH dev set.

pattern information are noisy.[2] This decomposition produces four streams:

$S = s_1, s_2, s_3, ..., s_N$ (stems)
$R = r_1, r_2, r_3, ..., r_N$ (roots)
$P = p_1, p_2, p_3, ..., p_N$ (patterns)
$M = m_1, m_2, m_3, ..., m_N$ (morph classes)

An example of a decomposed word is shown below:

Word:	akallimak (*I talk to you (masc.sg.)*)
Stem:	kallim
Morph:	verb+subj-1st-sg+DO-2nd-masc-sg
Root:	klm
Pattern:	CACCiC

Note that the order of the substrings in the 'Morph' tags is not required to match the order of different affixes in the word – the label simply represents a bag of morphological features. Table 8 lists the number of different types for both words and morphological components, based on the converted CH dictionary. Trigram models were built for each of the streams, using Kneser-Ney smoothing and interpolation of bigram and trigram probabilities. For language model rescoring, the hypothesized word sequence is mapped to the corresponding sequences of stems, roots, etc., and each sequence is rescored with the corresponding trigram. The final language model score for each hypothesis can be computed as a log-linear combination of the different stream scores. The weights of this combination can be optimized to directly minimize word error rate, similar to the framework of discriminative model combination (Beyerlein, 1998). The trigram perplexities for individual streams are listed in Table 8. These are not directly comparable to the perplexities of language models predicting words, but they provide an impression of the complexity of the prediction task within each stream.

[2] An exact quantification of the analyzer's error rate on dialectal data is not possible due to the lack of hand-annotated reference material.

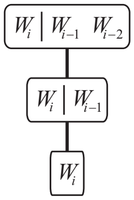

Fig. 4. Backoff path in a standard word-based language model.

4.2.4 Factored language models

A novel type of model, termed *factored language model (FLM)*, is used to explicitly represent interdependencies among the morphological components of words both across time and within a word, thus generalizing the stream models described above. We present a brief summary of this approach here; a more detailed description can be found in (Kirchhoff et al., 2002; Bilmes and Kirchhoff, 2003). We assume that each word corresponds to a bundle of k features or *factors*, such that $w_i \equiv \{f_i^1, f_i^2, \ldots, f_i^K\}$. In general, factors can be any features relevant to the word, e.g. part-of-speech tags, semantic features, or, as in this case, morphological components. A word sequence of length N can thus be converted to K parallel sequences of factors, denoted as $f_1^{1:K}, f_2^{1:K}, ..., f_N^{1:K}$. A statistical trigram model over this representation would be defined as follows:

$$p(f_1^{1:K}, f_2^{1:K}, ..., f_N^{1:K}) \approx \prod_{i=3}^{N} p(f_i^{1:K} | f_{i-1}^{1:K}, f_{i-2}^{1:K}) \tag{6}$$

The factored word representation can be exploited during language model backoff, in order to estimate word n-gram probabilities more robustly. To this end, we have developed a novel *generalized parallel backoff (GBP)* procedure. Standard Katz-style backoff (Katz, 1987) is defined as

$$p_{BO}(w_i|w_{i-1}, w_{i-2}) = \begin{cases} d_{N(w_i,w_{i-1},w_{i-2})} p_{ML}(w_i|w_{i-1}, w_{i-2}) & \text{if } c > \tau_3 \\ \alpha(w_{i-1}, w_{i-2}) p_{BO}(w_i|w_{i-1}) & otherwise \end{cases} \tag{7}$$

where p_{ML} denotes a maximum-likelihood estimate, c denotes the count of the triple (w_i, w_{i-1}, w_{i-2}) in the training data, τ_3 is the count threshold above which the maximum-likelihood estimate is retained, and $d_{N(w_i,w_{i-1},w_{i-2})}$ is a discounting factor (generally between 0 and 1) that is applied to the higher-order distribution. The normalization factor $\alpha(w_{i-1}, w_{i-2})$ ensures that the distribution sums to unity. In standard word-based LMs, zero probabilities for unseen trigrams are often prevented by backing off to the next lower-order probability distribution, proceeding from a trigram to a bigram to a unigram. This can be visualized as a backoff *path* (Figure 4). In a factored representation, where temporally synchronous as well as temporally successive elements are present, it is less obvious in which order the conditioning variables should

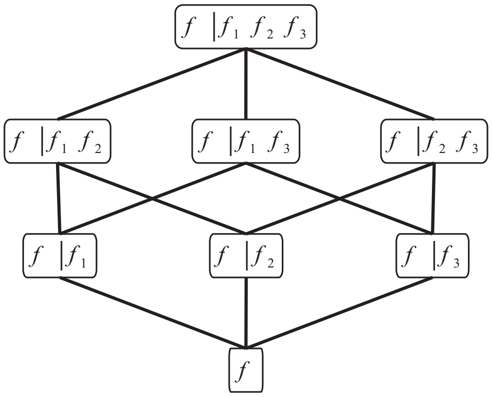

Fig. 5. Backoff graph for a factored language model (4-gram).

be dropped. The different possibilities for backing off from a higher-order to lower-order distribution can be summarized in a backoff *graph*, exemplified in Figure 5 for a 4-gram. There are several different ways of choosing among different paths in this graph:

1. Choose a fixed, predetermined backoff path based on linguistic knowledge, e.g. always drop syntactic before morphological variables.
2. Choose the path at run-time based on statistical criteria.
3. Choose multiple paths and combine their probability estimates.

Together, the last two options define the generalized parallel backoff method, which is implemented via a new backoff function (here shown for a 4-gram):

$$p_{GBO}(f|f_1, f_2, f_3) = \begin{cases} d_c p_{ML}(f|f_1, f_2, f_3) \ \ if \ \ c > \tau_4 \\ \alpha(f_1, f_2, f_3) g(f, f_1, f_2, f_3) \ \ otherwise \end{cases}$$

where c is the count of (f, f_1, f_2, f_3), $p_{ML}(f|f_1, f_2, f_3)$ is the maximum likelihood distribution, τ_4 is the count threshold, and $\alpha(f_1, f_2, f_3)$ is the normalization factor. The function $g(f, f_1, f_2, f_3)$ determines the backoff strategy. In a traditional backoff procedure $g(f, f_1, f_2, f_3)$ equals $p_{BO}(f|f_1, f_2)$. In generalized parallel backoff, however, g can be *any non-negative function* of f, f_1, f_2, f_3. In our implementation of FLMs we consider several different g functions, including the mean, weighted mean, product, and maximum of the smoothed probability distributions over all subsets of the conditioning factors. In addition to different choices for g, different discounting parameters can be chosen at different levels in the backoff graph. For instance, at the topmost node, Kneser-Ney discounting might be chosen whereas at a lower node Good-Turing might be applied.

Certain aspects of the generalized parallel backoff technique in FLMs are similar to related approaches that have used multiple heterogeneous conditioning variables in discrete conditional probability models. Dupont and Rosenfeld (1997) developed an approach called lattice-based language modeling where sets of conditioning variables are ordered not only with respect to their temporal precedence but also by an inclusion hierarchy defined by increasingly more fine-grained word classes. Smoothing is done by 'two-dimensional backoff', i.e. both the 'vertical' and the 'horizontal' backoff possibilities are pursued and combined by linear interpolation. Similar procedures are reported in (Gildea, 2001; Wang, 2003; Zitouni et al., 2003), which all use multiple classes and a fixed pre-determined backoff path from more general to more specific classes. Wang and Harper (2002) use multiple syntactic-semantic features from an 'almost-parsing' model as additional conditioning variables in a statistical LM, together with a fixed backoff path determined empirically. FLMs are a generalization of these techniques in that they allow fixed backoff paths, but also the choice of paths at run-time depending on the particular n-gram under consideration. Moreover, different methods for combining probability estimates from multiple backoff paths are available (geometric mean, product, min, max, etc. in addition to linear interpolation), and different smoothing techniques can be used at different nodes in the backoff graph.

FLMs were implemented as an extension to the SRILM toolkit (Stolcke, 2002) and have been released as part of the standard distribution; a more detailed description of the implementation can be found in (Kirchhoff et al., 2002). Since their inception at the Johns-Hopkins University Summer Workshop in 2002, FLMs have been used successfully for various language modeling and speech processing tasks (Bilmes and Kirchhoff, 2003; Parandekar and Kirchhoff, 2003; Ji and Bilmes, 2004). For the present task we tested several FLM structures (different sets of conditioning factors and various backoff paths) manually to optimize the perplexity on the development set. The best model, shown in Figure 6, obtained a perplexity of 169, a reduction of 6% relative to the comparable word trigram (see Table 4).

5 Recognition System

Recognition experiments are carried out with the SRI DECIPHER$^{\mathrm{TM}}$speaker-independent continuous speech recognition system. The front-end consists of 52 mel-frequency cepstral coefficients (13 base coefficients plus first, second and third derivatives), which are subsequently reduced to 39-dimensional feature vectors by Heteroscedastic Linear Discriminant Analysis (HLDA). Mean and variance normalization as well as vocal tract length normalization (VTN) are performed. VTN is applied to speaker clusters that are obtained by clustering the acoustic signals for each conversation side into three clusters on average. The acoustic models are genonic HMMs (Digalakis and Murveit, 1994) with approximately 220 genones and 128 Gaussians per genone. Decoding is performed using a multi-pass recognition strategy (Murveit et al., 1993). In the first pass (Stage 1), N-best hypotheses are generated using phoneloop-adapted

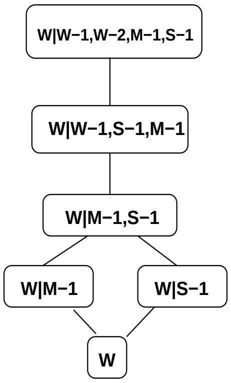

Fig. 6. FLM bigram for CH ECA. The model attempts to predict the current word based on the previous two words, the previous morph class and the previous stem. When this combination is not found, the most distant word is dropped, followed by the word at the previous time step. When backing off from the combination of previous morph and stems, both estimates obtained by just using either the previous morph or the previous stem are computed, and the larger of them is utilized. The final backoff node is always the word unigram probability.

non-crossword triphone models, a recognition lexicon of 18,352 words, and a bigram language model. Maximum word posterior hypotheses are obtained using N-best ROVER, which are then used to train speaker-adaptive training (SAT) and maximum-likelihood linear regression (MLLR) transforms for each speaker. These adapted models are then used to produce bigram lattices. The lattices are rescored with a trigram LM (trigram-I, based on preprocessing method I as described in Section 4.2) and are used as recognition networks for the following two recognition passes, one of which uses adapted non-crossword maximum-mutual-information (MMI) trained acoustic models, the other uses adapted crossword maximum-likelihood trained models. Each pass generates a set of N-best hypotheses, which is then rescored with additional language models (including the word-based trigram II, see Section 4.2) as described below (Stages 2a and 2b). The final hypotheses are obtained by using two-way N-best ROVER (Stolcke et al., 2000) (Stage 3), i.e. the hypotheses from the separate N-best lists are combined into one confusion network before rescoring. For language modeling training we used the SRILM language modeling toolkit (Stolcke, 2002) with FLM extensions as implemented and described in (Kirchhoff et al., 2002).

The utterance scores used for reranking the N-best hypotheses are a weighted combination of several component scores. In the baseline system these consist of the acoustic score, the language model scores from trigram-I and trigram-II, and the number of words. In our experimental systems, scores from various morphology-based language models are added. For score combination we use discriminative score combination framework (Beyerlein, 1998; Ostendorf et al., 1991). This approach aims at an optimal integration of independent sources of information in a log-linear model. The parameters to be trained discriminatively are the weights of the log-linear combination; these are optimized to minimize the word error rate on a held-out set. This method has been helpful in a variety of score combination tasks (Byrne et al., 2000; Vergyri et al., 2000; Glotin et al., 2001). For weight optimization we use a simplex downhill method known as amoeba search (Nelder and Mead, 1965), which is available as part of the SRILM toolkit.

6 Rescoring Experiments

In our initial experiments we investigated several different combinations of morphological language models for N-best list rescoring. Preliminary experiments were conducted at the Johns-Hopkins Summer Research Workshop 2002; these are described in (Kirchhoff et al., 2002). Here, those experiments were repeated and extended using a different baseline system. We had previously observed that morphological patterns did not contribute any information and usually received negative weights during weight optimization, possibly due to the errorful extraction mechanism. The models used in the following experiments were therefore restricted to either stream or class-based models for stems, morphs, and roots. In addition, the particle model and the FLM described in the previous section were used.

In a first step, the effect of different models and model combinations was investigated using a simpler baseline system than the one presented above which yielded a baseline word error rate of 56.1% on the development set during the second stage (only one set of N-best lists was used in this system). Table 9 shows the word error rates obtained with different LM combinations after weight optimization and language model rescoring. We observe that most morphology-based language models improve the word error rate marginally. Combinations of more than one model usually have a bigger effect; the largest reduction (0.5% absolute) is obtained by combining stream models with a class-based model, involving all three morphological components (roots, stems, and morph classes). The combination with an FLM did not yield an improvement on this set despite the improvement in perplexity observed before. Further analysis revealed that the perplexity obtained by the FLM on the actual N-best hypotheses (as opposed to the true reference transcriptions) was higher than that of the word-based n-gram.

We subsequently applied the two best combinations of language models to the evaluation system described in the previous section. Each of the two sets of N-best hypotheses was rescored with a different combination of models. The first system used the class-based models for each of the three factors *stem*, *morph*, and *root*. The second system used three stream models for same factors. The recognition results are shown in Table 10. Compared to the baseline results we

System	WER
Baseline	56.1
Baseline + particle model	56.0
Baseline + FLM	56.6
Baseline + added stream/class models	
morph stream	55.8
stem stream	56.0
root stream	56.0
morph class	55.9
stem class	55.9
root class	56.0
Baseline + added LM combinations	
morph stream, stem stream	56.7
morph stream, root stream	55.8
morph stream, stem stream, root stream	55.7
root stream, stem stream, morph stream, morph class	55.6
root stream, stem stream, morph stream, stem class	55.8
root class, stem class	55.9
morph class, stem class	55.8
morph class, stem class, root class	57.2
stem stream, morph stream, stem class	56.1

Table 9
Word error rates (%) on the CH dev set obtain by various combinations of the baseline language models with morphology-based language model scores.

see an absolute improvement of 1.3% othe development set and improvements of 1.0% and 0.6% on the eval97 and eval03 sets, respectively. Although these improvements are modest, they are consistent across all sets. Moreover, an analysis of the combination weights showed that consistently high weights were given to morphology-based models, demonstrating that they contribute useful information. For instance, the combination of the baseline models with various class-based morphological models yielded the weights 2.7 (trigram I), 1.2 (trigram II), 2.17 (stem class model), 3.52 (morph class model), and 0.27

	word-based LMs			morph-based LMs		
Stage	dev	eval97	eval03	dev	eval97	eval03
1	57.3	61.7	46.7	57.3	61.7	46.7
2a	54.8	58.2	40.8	53.4	56.9	39.9
2b	54.3	58.8	41.0	53.0	57.9	39.5
3	53.9	57.6	40.2	52.6	56.6	39.4

Table 10
Word error rates (%) obtained by N-best list rescoring with morphology-based language models. The leftmost column refers to the different recognition passes as described in Section 5. Word error rates for the first pass are the same for both systems since the the morphology-based language models are only used during rescoring.

(root class model). [3]

Larger improvements might be obtained by using morphological information in the first recognition pass: better hypotheses obtained at earlier stages in the recognition system can significantly affect adaptation and lattice generation at later stages. To this end we need to use a language model that makes use of morphological information but directly predicts words (rather than stems, roots, etc.). Of the models presented above, only the FLM meets these requirements: it predicts words but uses morphological factors during backoff.

We investigated two techniques to enable us to use FLMs for first-pass recognition. First, the choice of the best set of FLM parameters (i.e. the combination of conditioning factors, backoff path, and smoothing options) is important for good performance but is difficult to optimize by hand since the space of possible parameter combinations is very large. For this reason, we have developed an automatic optimization technique based on Genetic Algorithms (GAs), which is described in the following section. This procedure optimizes the model automatically and, furthermore, seeks to minimize perplexity only on those words present in the recognition lexicon. Second, first-pass recognition requires interfacing a standard word-based decoder with a language model based on a factored word representation. This is described in Section 8.

7 Automatic Parameter Search in Factored Language Models

Three types of parameters define an FLM: the set of initial conditioning factors, the backoff graph, and the smoothing options. The search space defined by these parameters is extremely large: Given a factored word representation

[3] These are normed weights such that the weight for the acoustic model is 1.

with a total of k factors, there are $\sum_{n=1}^{k} \binom{k}{n} = 2^k$ possible subsets of initial conditioning factors. Further, for a set of m conditioning factors, there are up to $m!$ backoff paths, each with its own smoothing options. In addition, the search space is complex: nonlinear interactions between parameters make it difficult to guide the search into a particular direction. For instance, a particular backoff path that works well with Kneser-Ney smoothing may perform poorly when a different smoothing method is chosen. For these reasons, we optimize parameters using Genetic Algorithms (Holland, 1975), which typically perform well on problems with large and poorly understood search spaces.

7.1 Parameter Search Using GAs

GAs work by encoding problem solutions as (mostly binary) strings and by evolving successive populations of solutions via genetic operators applied to these strings. Each string can be evaluated by a so-called "fitness function", which represents the desired optimization criterion. In each iteration of the algorithm, a new population with a higher average fitness is created. This is achieved by applying the genetic operators selection, crossover, and mutation. The selection operator selects particular strings from the general pool with a probability proportional to their fitness. Crossover creates new strings by splitting existing strings in random positions and recombining their constituent parts. Mutation randomly replaces bits in existing strings. Both crossover and mutation are applied with a fixed small probability. In our case, strings describe individual FLMs, and the fitness function is the perplexity of the FLM represented by a string. The encoding scheme that defines a mapping between a string and a particular FLM consists of three subparts representing the factors, the backoff graph, and the smoothing options, respectively.

Factor Encoding. The initial factors F are encoded as binary strings, with 0 representing the absence and 1 representing the presence of a factor. For example, a FLM trigram might have three factors (A,B,C) per word. Then the set of all potential conditioning factors S is: $\{A_{-1}, B_{-1}, C_{-1}, A_{-2}, B_{-2}, C_{-2}\}$ where the subscript indicates the time position of each factor. A particular set of initial factors F is a subset of S and can be represented as a 6 bit binary string, where 1 at a position indicates the inclusion of that factor in F. The string `100011`, for instance, means that F is: $\{A_{-1}, B_{-2}, C_{-2}\}$

Backoff Graph Encoding The backoff graph is encoded by means of graph grammar rules (similar to Kitano (1990)), since a direct approach encoding every edge as a bit would result in overly long strings and inefficient GA search. Grammar rules indicate which factor to drop and capture the graph regularity that a node with m factors can only back off to children nodes with $m - 1$ factors. For instance, for $m = 3$, three rules describe the choices for backing off to nodes with $m = 2$:

RULE 1:$\{X_1, X_2, X_3\} \rightarrow \{X_1, X_2\}$
RULE 2:$\{X_1, X_2, X_3\} \rightarrow \{X_1, X_3\}$
RULE 3:$\{X_1, X_2, X_3\} \rightarrow \{X_2, X_3\}$

Here X_i corresponds to the factor at the ith position in the parent node. Rule 1 indicates a backoff that drops the third factor, Rule 2 drops the second factor, etc. To describe the backoff from $\{A_{-1}, B_{-2}, C_{-2}\}$ to $\{A_{-1}, C_{-2}\}$ we would indicate that Rule 2 was activated (the second factor was dropped). To describe a parallel backoff from $\{A_{-1}, B_{-2}, C_{-2}\}$ to both $\{A_{-1}, C_{-2}\}$ and $\{B_{-2}, C_{-2}\}$ we would indicate that *both* Rule 2 and Rule 3 are activated.

The choice of rules used to generate the backoff graph is encoded in a binary string, with 1 indicating the use and 0 indicating the non-use of a rule. The backoff graph grows according to the rules specified by the gene, as shown schematically in Figure 7.

Encoding of Smoothing Options. Smoothing options are encoded as tuples of integers, each specifying the discounting method (e.g. Kneser-Ney, Witten-Bell smoothing) and backoff threshold (e.g. $\tau = \{1, 2, 3, \ldots\}$) at a node in the graph. A gene consists of the concatenation of the three strings representing the initial factors, the backoff graph, and the smoothing options. Genetic operators thus optimize all FLM parameters jointly.

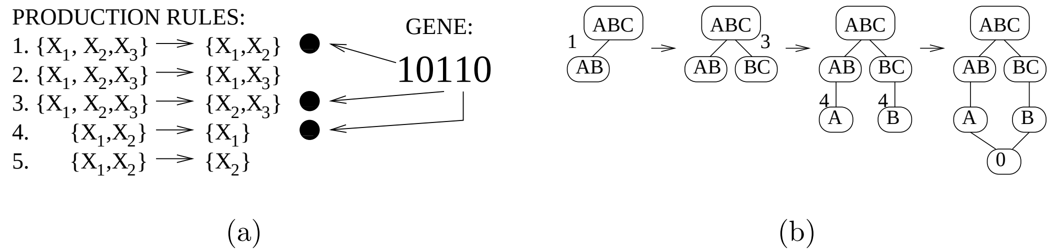

Fig. 7. Gene activates graph grammar production rules (a); Generation of Backoff graph by activated rules 1, 3, 4 (b).

7.2 GA Experiments and Parameter Search Results

In our application of GAs to FLM parameter search, the perplexity of models with respect to the development data was used as an optimization criterion. The perplexity of the best models found by the GA were compared to the best models identified by a lengthy manual search procedure using linguistic knowledge about dependencies between the word factors involved, and to a random search procedure which evaluated the same number of strings as the GA.

Several GA options were investigated to examine the impact on convergence speed and final result. Empirically, the crossover operator impacts performance the most, due to the specific encoding method employed. The differences among other GA options were less clear, and different options appear to give equally good results. This robustness of option choices is most likely due to the inherent robustness of the genetic algorithm. As a rule of thumb, the following options yielded good results: population size 30-50, crossover probability 0.9, mutation probability 0.01, Stochastic Universal Sampling as the selection operator, and 2-point crossover. We also experimented with re-initializing the GA search with the best model found in previous runs. This method consistently improved the performance of normal GA search and we used it as the basis for the results reported below.

n-gram	Word	Hand	Rand	GA	ΔPP (%)
	Development Set				
2-gram	230.3	228.4	232.1	223.9	-2.0
3-gram	227.1	226.9	231.7	212.6	-6.3
	Evaluation Set (eval97)				
2-gram	227.9	226.2	230.5	222.4	-1.7
3-gram	222.3	226.6	228.1	208.4	-6.3

Table 11
Perplexity for word-based LMs and FLMs with parameters optimized using manual, random, and GA search.

Table 11 compares the best perplexity results for standard word-based models and for FLMs obtained using manual search (Hand), random search (Rand), and GA search (GA). The last column shows the relative change in perplexity for the GA compared to the best of the word, manual or random search models. Results are shown for the CH dev and eval97 sets. These perplexities are evaluated *without* consideration of out-of-vocabulary (OOV) words, since the speech recognizer has a fixed vocabulary and cannot recognize OOV words. If this constraint were not used, the GA might attempt to minimize perplexity on OOV tokens rather than on n-grams known to the decoder.

The results show that GA search yielded the lowest perplexity on both the development and evaluation set. In general, the best FLMs found by GA search used all available conditioning factors (word, stem, root, pattern, and morph class) and parallel backoff with different smoothing options at different nodes in the backoff graph. A graphical representation of the best bigram model can be found in the Appendix. The FLMs with the best perplexity on the development set were used for the first-pass recognition experiments, described in the following sections.

8 First-pass recognition with factored language models

Since promising results were obtained by applying morphological knowledge during rescoring, we expect to gain a further improvement when applying it at earlier recognition passes. However, the use of FLMs in first-pass recognition is problematic because standard word-based decoders cannot process the decomposed word representations required by FLMs. For this reason we use a novel feature of the SRILM toolkit that allows us to 'rescore' a word-based language model with an FLM using the following steps:

(1) The entries in the word-based LM are converted to a factored word representation, based on a lexicon.
(2) The factored representations are then passed through the FLM trained on the decomposed training text and are assigned new probabilities from this FLM.
(3) After renormalization, the entries are converted back to words and written out as a new LM in standard ARPA format for use with a word-based decoder.

When applied to a development or test set, the rescored word LM typically yields a higher perplexity than the corresponding FLM. This is because unseen word n-grams in the new text can be assigned probabilities in the FLM by backing off to previously encountered factor combinations (e.g. morph class or stem n-grams); however, if the corresponding word n-grams are not present in the original word-based LM, they will not be present in the rescored LM. For this reason, additional word n-grams need to be added prior to rescoring in order to derive the maximum benefit from the FLM. Adding all possible bigrams and trigrams is clearly infeasible. We select bigrams which do not exist in the original training data by searching over all possible bigrams and retaining those for which

$$p_{FLM}(w,h)(log(p_{FLM}(w|h)) - log(p_{word}(w|h))) > \epsilon$$

where h is the word history, p_{FLM} is the probability obtained by the original FLM and p_{word} is the probability obtained by the word LM. This criterion is derived from previous experiments on language model pruning (cf. Stolcke et al. (2000)) and approximates the relative entropy between the FLM and the rescored word-based n-gram model. The value of ϵ was chosen such that p_{word} would be within 2% of that of the FLM. Since a comparable search over the entire trigram space is infeasible, the search is conducted only for those trigrams for which both component word bigrams have already been added based on the above criterion.

Table 12 compares the perplexities on the dev and eval sets obtained by different language models. The results show that the use of FLMs (line II) leads

	dev		eval97		eval03	
	2-gram	3-gram	2-gram	3-gram	2-gram	3-gram
I	230	227	227	222	132	123
II	223	213	222	209	136	89
III	250	227	249	225	145	141
IV	226	217	225	215	137	137

Table 12
Bigram and trigram perplexities obtained by: the word-based baseline model (I), the FLM (II), the baseline model rescored with the FLM without adding additional n-grams (III), and with added n-grams (IV), on the different CH sets.

to perplexity reductions on all sets, with the exception of the bigram on the eval03 set. The slight increase in bigram perplexity and the significant reduction in trigram perplexity on eval03 is a combination of the nature of the data set as well as the smoothed probability estimates provided by the FLM. As explained in Section 4.2, the eval03 set has a much higher n-gram coverage than the dev and eval97 sets. For bigrams, the standard word-based model may therefore already provide reasonable probability estimates, while the highly smoothed estimates from the FLM (resulting from the combination of a large number of component models) actually lead to a higher perplexity. In the trigram context, by contrast, the highly smoothed estimates are beneficial and reduce the perplexity considerably.

The differences between rows II and III/IV demonstrate the loss in performance due to the rescoring procedure described above, which prevents us from exploiting the benefits of FLMs to the full extent. This is particularly obvious for the trigram applied to the eval03 set. The trigrams that are added in IV depend on previously added bigrams; however, the FLM bigram model itself already leads to a worse perplexity than the word-based language model, which is why not much improvement can be expected in this case.

Table 13 shows the word error rates obtained by applying the rescored language models in the first recognition pass in addition to using stream and class-based models during rescoring. Additional absolute improvements of 0.5% and 0.4% were obtained on the dev and eval97 sets, whereas the eval03 set showed a 0.2% absolute degradation. The overall improvements compared to a baseline system that does not make use of morphological information thus are 1.8% (dev), 1.5% (eval97) and 0.6% (eval03).

	Baseline, word-based LMs			N-best lists rescored with morph-based LMs			Rescoring plust 1st pass pass recognition w/ FLM		
Stage	dev	eval97	eval03	dev	eval97	eval03	dev	eval97	eval03
1	57.3	61.7	46.7	57.3	61.7	46.7	56.2	61.0	46.3
2a	54.8	58.2	40.8	53.4	56.9	39.9	52.7	56.5	40.2
2b	54.3	58.8	41.0	53.0	57.9	39.5	52.3	57.4	40.1
3	53.9	57.6	40.2	52.6	56.6	39.4	52.1	56.1	39.6

Table 13
Word error rates (%) obtained by using approximations of FLMs during the first recognition pass, in addition to morphological stream and class models during rescoring.

9 Conclusions

We have presented an overview of different approaches to morphology-based language modeling for Arabic LVCSR. The models we have developed include particle-based models, morphological stream models, class-based models where classes are defined by morphs, and a new type of language model called factored language model. The latter uses a backoff procedure where both words and/or additional morphological features can be used in combination. We tested these models in N-best list rescoring experiments; in addition, language models derived from FLMs were used in first-pass recognition. This was facilitated by the automatic optimization of FLM structure and parameters, and by a language model approximation procedure which allows probability estimates from a FLM to replace those in a word-based language model which can be used with a standard decoder. The combined use of these procedures led to significant word error rate reductions on one evaluation set and to non-significant but consistent improvements on the other. A drawback of the FLM approximation procedure is that not all word combinations which are implicitly represented in the FLM can be represented explicitly in the approximating word-based language model; the search space and memory requirements would be too large. Future work will focus on improving the interface between decoder and FLMs and on alternative approximation methods, e.g. techniques based on frequent occurrences of morphological factor combinations rather than word combinations in the training data.

The main objective of this study was to determine the relevance of morphology-based language models in ASR, i.e. its potential for reducing word error rate given perfect knowledge of the morphological word structure. However, the methods presented here also lend themselves to modeling automatically learned morphological classes, or other factors obtained by data-driven word decomposition or clustering. A limited investigation of such data-driven techniques has been reported in (Kirchhoff et al., 2002). We are currently exploring additional ways of learning factors automatically. Another direction currently under investigation is the use of the FLM framework for sharing training data across different languages or different varieties of a language, such as different dialects. This should prove particularly useful for a language like Arabic, where little training data for individual dialects is available.

Acknowledgements

This material is based upon work funded by the NSF under grant No. 0121285, by DARPA under contract No. MDA972-02-C-0038, and by the NSF and the CIA under NSF Grant No. IIS-0326276. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of these agencies. We would like to thank Karim Darwish for providing the morphological analyzer.

APPENDIX

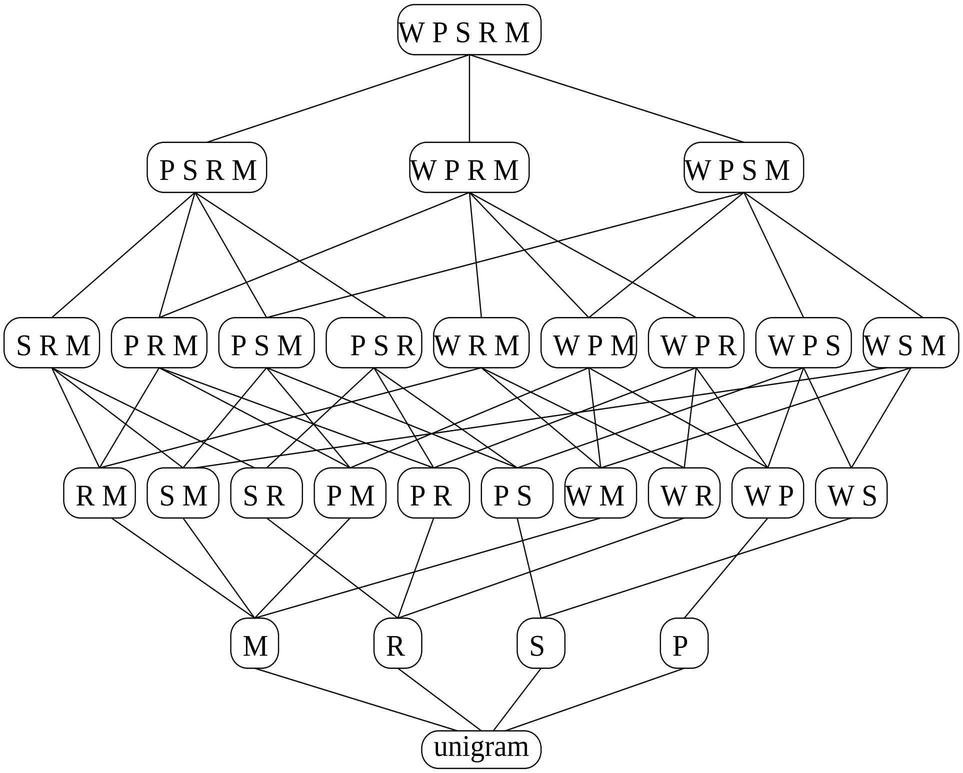

Fig. 8. Backoff graph of the Arabic bigram FLM used for rescoring and first-pass recognition. The characters W, P, S, R, M represent the previous word, pattern, stem, root, and morphological class factors, respectively. The top node thus stands for the probability distribution $P(W_i|W_{i-1}, P_{i-1}, S_{i-1}, R_{i-1}, M_{i-1})$. At each of the lower level nodes, one of the conditioning factors is dropped. Multiple paths entering one node indicate the weighted mean combination of the corresponding probability estimates. Note that although we have more than one conditioning variable, we retain the term "bigram" for a model of this type, to incidate that only factors pertaining to the preceding word are required. This allows us in principle to use the model in a bigram decoding framwork.

References

Abdel-Massih, E., 1975. An Introduction to Egyptian Arabic. The University of Michigan Press, Ann Arbor.

Beyerlein, P., 1998. Discriminative model combination. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 481–484.

Billa, J., Ma, K., McDonough, J., Zavaliagkos, G., Miller, D., Ross, K., El-Jaroudi, A., 1997. Multilingual speech recognition: The 1996 Byblos Callhome system. In: Proceedings of Eurospeech. pp. 363–366.

Billa, J., Noamany, M., Srivastava, A., Liu, D., Stone, R., Zu, J., Makhoul, J., Kubala, F., 2002. Audio indexing of broadcast news. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 5–8.

Bilmes, J. A., Kirchhoff, K., 2003. Factored language models and generalized parallel backoff. In: Proceedings of HLT/NACCL. pp. 4–6.

Brown, P., et al., 1992. Class-based n-gram models of natural language. Computational Linguistics 18 (4), 467–479.

Byrne, B., Hajic, J., Ircing, P., Jelinek, F., Khudanpur, S., Krbec, P., Psutka, J., 2001. On large vocabulary continuous speech recognition of highly inflectional language - Czech. In: Proceedings of Eurospeech. pp. 487–490.

Byrne, W., Beyerlein, P., Huerta, J., Khudanpur, S., Marthi, B., Morgan, J., Peterek, N., Picone, J., Vergyri, D., Wang, W., 2000. Towards language independent acoustic modeling. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 1029–1032.

Çarki, K., Geutner, P., Schultz, T., 2000. Turkish LVCSR: towards better speech recognition for agglutinative languages. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 134–137.

Cerf-Dannon, H., El-Bèze, M., 1991. Three different probabilistic language models: comparison and combination. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 297–300.

Darwish, K., 2002. Building a shallow Arabic morphological analyser in one day. In: Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages. Philadelphia, PA.

Digalakis, V., Murveit, H., 1994. Genones: Optimizing the degree of mixture tying in a large vocabulary hidden Markov model based speech recognizer. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 537–540.

Dupont, P., Rosenfeld, R., 1997. Lattice based language models. Tech. Rep. CMU-CS-97-173, Carnegie-Mellon University, Department of Computer Science.

El-Bèze, M., Derouault, A., 1990. A morphological model for large vocabulary speech recognition. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 577–580.

Geutner, P., 1995. Using morphology towards better large-vocabulary speech recognition systems. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 445–448.

Gildea, D., 2001. Statistical language understanding using frame semantics. Ph.D. thesis, University of California, Berkeley.

Glotin, H., Vergyri, D., Neti, C., Potamianos, G., Luettin, J., 2001. Weighting schemes for audio-visual fusion in speech recognition. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 173–176.

Holland, J., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

Ji, G., Bilmes, J., 2004. Multi-speaker language modeling. In: Proceedings of HLT/NAACL. pp. 137–140.

Katz, S. M., March 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. Proceedings of the International Conference on Acoustics, Speech and Signal Processing 35 (3), 400–401.

Kiecza, D., Schultz, T., Waibel, A., 1999. Data-driven determination of appropriate dictionary units for Korean LVCSR. In: Proceedings of the International Conference on Speech Processing. pp. 323–327.

Kirchhoff, K., Bilmes, J., Henderson, J., Schwartz, R., Noamany, M., Schone, P., Ji, G., Das, S., Egan, M., He, F., Vergyri, D., Liu, D., Duta, N., 2002. Novel speech recognition models for Arabic. Tech. rep., Johns Hopkins University.

Kirchhoff, K., Vergyri, D., 2004. Cross-dialectal acoustic data sharing for Arabic speech recognition. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. Montreal, Canada, pp. 765–768.

Kitano, H., 1990. Designing neural networks using genetic algorithms with graph generation system. Complex Systems, 461–476.

Murveit, H., Butzberger, J., Digalakis, V., Weintraub, M., 1993. Large-vocabulary dictation using SRI's DECIPHER(TM) speech recognition system: Progressive-search techniques. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 319–322.

Nelder, J., Mead, R., 1965. A simplex method for function minimization. Computing Journal 7(4), 308–313.

Ostendorf, M., Kannan, A., Austin, S., Kimball, O., Schwartz, R., Rohlicek, J. R., 1991. Integration of diverse recognition methodologies through reevaluation of N-best sentence hypotheses. In: Proceedings of the DARPA Speech and Language Workshop. pp. 83–87.

Parandekar, S., Kirchhoff, K., 2003. Multi-stream language identification using data-driven dependency selection. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 28–31.

Porter, M., 1980. An algorithm for suffix stripping. Program 14(3), 130–137.

Schulz, E., Krahl, G., Reuschel, W., 2000. Standard Arabic - An Elementary-Intermediate Course. Cambridge University Press.

Schwartz, R., Colthurst, T., Duta, N., Gish, H., Iyer, R., Kao, C.-L., Liu, D., Kimball, O., Ma, J., Makhoul, J., Matsoukas, S., Nguyen, L., Noamany, M., Prasad, R., Xiang, B., Xu, D., Gauvain, J.-L., Lamel, L., Schwenk, H., Adda, G., Chen, L., 2004. Speech recognition in multiple languages and domains: The 2003 BBN/LIMSI EARS system. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. pp. 753–756.

Stolcke, A., 2002. SRILM- an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing. pp. 901–904.

Stolcke, A., Bratt, H., Butzberger, J., Franco, H., Gadde, V. R. R., Plauche, M., Richey, C., Shriberg, E., Sonmez, K., Weng, F., Zheng, J., 2000. The SRI March 2000 Hub-5 conversational speech transcription system. In: Proceedings of the NIST Speech Transcription Workshop. College Park, MD.

Vergyri, D., Tsakalidis, S., Byrne, W., 2000. Minimum risk acoustic clustering for multilingual acoustic model combination. In: Proceedings of the International Conference on Spoken Language Processing.

Wang, W., 2003. Factorization of language models through backing off lattices. Computation and Language E-print Archive, oai:arXiv.org/cs/0305041.

Wang, W., Harper, M., 2002. The SuperARV language model: investigating the effectiveness of tightly integrating multiple knowledge sources. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 238–247.

Whittaker, E., 2000. Statistical language modelling for automatic speech recognition of Russian and English. Ph.D. thesis, Department of Engineering, Cambridge University, Cambridge, UK.

Zitouni, I., Siohan, O., Lee, C.-H., 2003. Hierarchical class n-gram language models: towards better estimation of unseen events in speech recognition. In: Proceedings of Eurospeech - Interspeech. pp. 237–240.