Morphology-Based Language Modeling for Arabic Speech Recognition

Dimitra Vergyri[1] *, Katrin Kirchhoff*[2] *, Kevin Duh*[2] *, Andreas Stolcke*[1]

[1]Speech Technology and Research Laboratory
SRI International
Menlo Park, CA, USA
{dverg,stolcke}@speech.sri.com

[2]Department of Electrical Engineering
University of Washington
Seattle, WA, USA
{katrin,duh}@ee.washington.edu

Abstract

Language modeling is a difficult problem for languages with rich morphology. In this paper we investigate the use of morphology-based language models at different stages in a speech recognition system for conversational Arabic. Class-based and single-stream factored language models using morphological word representations are applied within an N-best list rescoring framework. In addition, we explore the use of factored language models in first-pass recognition, which is facilitated by two novel procedures: the data-driven optimization of a multi-stream language model structure, and the conversion of a factored language model to a standard word-based model. We evaluate these techniques on a large-vocabulary recognition task and demonstrate that they lead to perplexity and word error rate reductions.

1. Introduction

A standard statistical language model (LM) computes the probability of a word sequence $W = w_1, w_2, ..., w_T$ as a product of the conditional probabilities of each word w_i given its history, which is typically approximated by the one or two most recent words. Even with this limitation, the estimation of LM probabilities is challenging since many word contexts are observed infrequently or not at all. This is particularly problematic for morphologically rich languages, e.g. Turkish, Russian, or Arabic. Such languages have a high vocabulary growth rate, which results in high language model perplexity and a large number of out-of-vocabulary (OOV) words (see e.g. [1, 2, 3, 4, 5]). Recently, *Factored Language Models (FLMs)* [5, 6] have been developed to address this problem. FLMs decompose words into a number of features and use the resulting representation in a generalized backoff scheme that improves the robustness of probability estimates for rarely observed word n-grams. A straightforward way to use FLMs and other morphology-based LMs in automatic speech recognition (ASR) is by rescoring N-best lists and combining scores from different models for final hypothesis selection. Here, we present results using this technique as well as two extensions to this approach: (a) the automatic optimization of FLM design parameters using a data-driven procedure, and (b) the use of FLMs in first-pass recognition rather than rescoring.

2. Factored Language Models

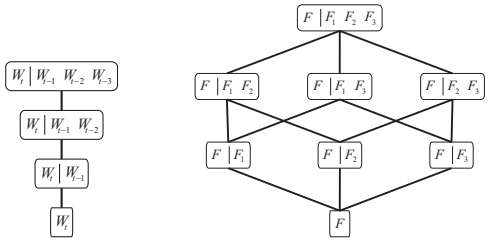
FLMs decompose each word w into a set of k features (or *factors*), i.e. $w \equiv f^{1:K}$. Factors represent morphological, syntactic, or semantic word information and can be e.g. stems, POS tags, etc. in addition to the words themselves. Probabilistic LMs are then constructed over (sub)sets of factors. Using a trigram approximation, this can be expressed as:

$$p(f_1^{1:K}, f_2^{1:K}, ..., f_T^{1:K}) \approx \prod_{t=3}^{T} p(f_t^{1:K} | f_{t-1}^{1:K}, f_{t-2}^{1:K}) \quad (1)$$

Each word is dependent not only on a single stream of temporally preceding word variables, but also on additional parallel streams of features. Such a representation can be used to back off to factors when the word n-gram has not been observed in the training data, thus improving probability estimates. For instance, a word trigram may not have any counts in the training set, but its corresponding factor combinations (e.g. stems and other morphological tags) may have been observed since they also occur in other words. This is achieved via a new *generalized parallel backoff* technique. In standard Katz-style backoff, the maximum-likelihood estimate of an n-gram with too few observations in the training data is replaced with a probability derived from the lower-order $(n-1)$-gram and a backoff weight as follows:

$$p_{BO}(w_t|w_{t-1}, w_{t-2}) \quad (2)$$
$$= \begin{cases} d_c p_{ML}(w_t|w_{t-1}, w_{t-2}) \text{ if } c > \tau_3 \\ \alpha(w_{t-1}, w_{t-2}) p_{BO}(w_t|w_{t-1}) \text{ otherwise} \end{cases}$$

where c is the count of (w_t, w_{t-1}, w_{t-2}), p_{ML} denotes the maximum-likelihood estimate, d_c is a discounting factor and $\alpha(w_{t-1}, w_{t-2})$ is a normalization factor. During standard backoff, the most distant conditioning variable (in this case w_{t-2}) is dropped first, followed by the second most distant variable etc., until the unigram is reached. This can be visualized as a backoff *path* (Figure 1(a)). If additional conditioning variables



(a) (b)

Figure 1: Standard backoff path for a 4-gram language model over words (left) and backoff graph for 4-gram over factors (right).

are used which do not form a temporal sequence, it is not immediately obvious in which order they should be dropped. In this case, several backoff paths are possible, which can be summarized in a backoff *graph* (Figure 1(b)). Paths in this graph can be chosen in advance based on linguistic knowledge, or at run-time based on statistical criteria such as counts in the training set. It is also possible to choose multiple paths and combine their probability estimates. The use of multiple conditioning factors is similar to the procedure described in [7] but is more general in that it allows arbitrary backoff paths instead of imposing an *a priori* ordering of more specific to more general probability distributions. Moreover, it provides different combination methods for probability estimates obtained from different paths. This is achieved by replacing the backed-off probability p_{BO} in Equation 2 by a general function g, which can be *any non-negative function* applied to the counts of the lower-order n-gram. Several different g functions can be chosen, e.g. the mean, weighted mean, product, minimum or maximum of the smoothed probability distributions over all subsets of conditioning factors [5]. In addition to different choices for g, different discounting parameters can be selected at different levels in the backoff graph. For instance, at the topmost node, Kneser-Ney discounting might be used whereas at a lower node Good-Turing might be applied. FLMs have been implemented as an add-on to the widely-used SRILM toolkit. Further details can be found in [5]. One difficulty in training FLMs is the choice of the best combination of design choices, in particular the conditioning factors, backoff path(s) and smoothing options. Since the space of different combinations is too large to be searched exhaustively, we have developed an automatic procedure to optimize FLMs, further described in Section 4.2.

3. Data and Baseline System

The experiments reported here were run on the LDC CallHome corpus of Egyptian Colloquial Arabic (ECA). The training set consists of the training, hub5_new and eval96 subsets and contains 120 conversations (˜180K words) in total. The development set (dev) has 32K words and the two test sets have 18K (eval97) and 11K (eval03) words, respectively. The recognizer was trained on the 'romanized' transcriptions of the data. 9% of all word tokens are disfluencies and 1.6% are foreign words. The recognition dictionary consisted of 18K words.

For recognition we use the SRI DECIPHER^{TM} system. The front-end consists of 52 mel-frequency cepstral coefficients (13 base coefficients + 1st + 2nd +3rd differences), reduced with HLDA to 39 dimensions. Mean and variance as well as vocal tract length normalization are performed for speaker clusters (the waves for each conversation side were clustered into an average of 3 speaker clusters). Continuous-density, genonic hidden Markov models [8] with 128 Gaussians per genone are used. The system contains approximately 220 genones. The decoder uses a multipass approach: In the first pass (Stage 1), N-best hypotheses are generated using phoneloop-adapted non-crossword models and a bigram LM. 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. The adapted models are used in the second pass to produce bigram lattices. The lattices are rescored with a trigram LM and are used as recognition networks for the following passes. Two more passes are performed, one using adapted non-crossword maximum-mutual-information (MMI) trained models, and one using adapted crossword maximum-likelihood trained models. Thus we obtain two sets of N-best hypotheses, each of which is rescored with additional morphology-based LMs as described below (Stages 2a and 2b). The final hypotheses are generated by 2-way N-best ROVER (Stage 3).

4. ASR Using Morphology-based LMs

Morphological information for language modeling is obtained by extracting the stem and the morphological class for each word from the CallHome ECA lexicon, and by using a morphological analyzer [9] to further decompose the stem into a root (typically a sequence of three consonants) and a pattern (a sequence of consonant and vowel slots) (cf. [5]). Root and pattern decomposition is noisy since the analyzer was developed for a different dialect of Arabic.

4.1. Fixed FLM Topologies

In the system submitted for the RT-03 benchmark evaluations [10], we used morphology-based language models to rescore the N-best lists prior to applying ROVER. The factors considered for LM training were: root, stem, and morphological class. For each of the two sets of N-best lists, a different combination of rescoring LMs was employed. The first used three class-based LMs, where the classes were defined based on each of the above-mentioned factors. The second used three FLMs, each with a fixed backoff path allowing backoff only to a single factor. This led to word error rate reductions on the eval03 set of 0.8% and 1.5% (absolute), respectively. In the final 2-way ROVER combination pass we obtained improvements of 1.3% and 0.8% on the dev and eval03 test sets, respectively (see "N-best" columns in Table 3).

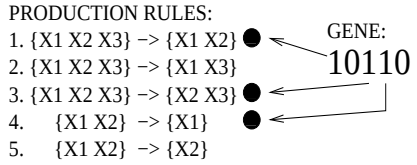
4.2. Automatic FLM Parameter Search

Since the space of possible FLM structures is very large we explored the use of Genetic Algorithms (GAs) to optimize the choice of conditioning factors, backoff paths, and smoothing options. GAs [11] encode problem solutions as strings (genes), and evolve successive populations of solutions through the use of genetic operators (e.g. selection, cross-over, mutation). The probability of each gene's survival is dependent on its fitness function, which represents the desired optimization criterion. In this case, each gene represents an FLM with a specific set of parameters, i.e. the initial conditioning factors, the backoff graph, and the smoothing options. The fitness function is the FLM's perplexity on the development set.

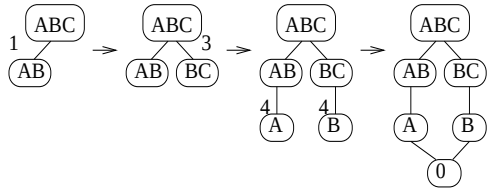
The main problem in applying GAs to our current task is to find a good encoding of the problem. The initial set of conditioning factors F is encoded as a binary string. For instance, a trigram for a word representation with three factors (A,B,C) has six potential conditioning variables: $\{A_{-1}, B_{-1}, C_{-1}, A_{-2}, B_{-2}, C_{-2}\}$ which can be represented as a 6-bit binary string, with a bit set to 1 indicating presence and 0 indicating absence of a factor in F. The string 10011 would correspond to $F = \{A_{-1}, B_{-2}, C_{-2}\}$. The backoff graph is encoded by means of graph grammar rules (similar to [12]), since a direct approach encoding every edge as a bit would result in overly long strings and inefficient GA search. (There are up to $m!$ backoff paths for a FLM with m initial factors). The grammar rules capture the regularity that a node with m factors can only back off to children nodes with $m-1$ factors. For instance, for $m = 3$, the choices for proceeding to the next-lower level in the backoff graph can be described by the following grammar rules:

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. 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 2. The smoothing options are encoded as tuples of integers, each specifying the discounting method and backoff threshold at a node in the graph. Finally, the GA operators are applied to concatenations of all three substrings describing the set of factors, backoff graph, and smoothing options, such that all parameters are optimized jointly. Table 1 lists the perplexities of the word-based n-grams, of the best FLMs obtained by a manual parameter search, random search, and the GA-based search. We observe that the GA procedure leads to 3% (bigram) and 6% (trigram) relative reductions in perplexity and performs better than either manual or random search. Models were optimized to reduce the perplexity on the known words, ignoring the probability given to OOV words. This constraint prevents the GA from minimizing perplexity by choosing models which assign high probability to OOV words rather than to words present in the recognition dictionary. The best-performing FLMs use all morphological factors (stems, morph classes, roots and patterns in addition to words) and parallel backoff with different smoothing options at different nodes in the backoff graph.



(a) Gene activates production rules



(b) Generation of Backoff Graph by rules 1, 3, and 4

Figure 2: Generation of Backoff Graph from production rules selected by the gene 10110.

Model	word	FLM manual	FLM rand	FLM GA
bigram	229.9	229.6	229.9	226.1
trigram	227.1	223.2	230.3	212.6

Table 1: Bigram and trigram perplexities on the CH dev set for word-based LMs and for FLMs obtained by manual, random (rand) and genetic search (GA).

Set	dev		eval97		eval03	
n	2	3	2	3	2	3
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 2: Bigram and trigram perplexities obtained by: the word-based baseline LM (I), the FLM (II), the baseline LM rescored with the FLM without adding additional n-grams (III), and with added n-grams (IV), on the different CH sets.

4.3. Converting FLMs to Word-based LMs

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. Better hypotheses at early passes can positively influence adaptation and re-recognition results at later 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. First, the entries in the word-based LM are converted to a factored representation based on a lexicon. They are then passed through the FLM trained on the decomposed training text and are assigned new probabilities from this FLM. 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 obtains 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 to the new 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 (cf. [13]). 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, we only search over those trigrams for which both word bigrams have already been added based on the above criterion. Table 2 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 to perplexity reductions on all sets with the exception of the bigram on the eval03 set. Since reductions are achieved on the eval97 set, it is unlikely that this is due to an overfitting to the development data by the GA search procedure; rather, it seems to be the case that the eval03 is very different in nature from the other two sets. This is confirmed by the much lower perplexities and typical word error rates (around 40%) obtained on this set. The

	dev			eval97			eval03		
Stage	baseline	FLM Nbest rescoring	FLM all passes	baseline	FLM Nbest rescoring	FLM all passes	baseline	FLM Nbest rescoring	FLM all passes
(1)	57.3		56.2	61.7		61.0	46.7		46.3
(2a)	54.8	53.4	52.7	58.2	56.9	56.5	40.8	39.9	40.2
(2b)	54.3	53.0	52.5	58.8	57.9	57.4	41.0	39.5	40.1
(3)	53.9	52.6	52.1	57.6	56.6	56.1	40.2	39.4	39.6

Table 3: Word error rates (in %) obtained by the baseline system , the system using morphology-based LMs for N-best list rescoring, and the system using morphology-based FLM in all recognition passes and the previous models for N-best rescoring, at different recognition stages as described in Section 3.

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. Since trigrams that are added in IV are dependent on previously added bigrams, perplexity does not decrease but increase in this case.

4.4. Recognition Experiments

In order to evaluate the total effect of the morphology-based LMs in the multipass system, we replaced the standard word-based LM used in the baseline with Model IV in Table 2. Recognition results (Table 3) show that the use of morphology-based LMs improves WER by 1.8% absolute on the dev set and 1.5% on the eval97 set. One third of that improvement (0.5%) is due to the use of morphological knowledge throughout all the recognition passes.

The last columns in Table 3 show the results on the eval03 set. Here, the application of morphological LMs in the rescoring pass leads to improvements comparable to those on the dev and eval97 sets; however, the use of the rescored LM from the first-pass slightly hurts rather than helps the performance. This is most likely due to the increase in perplexity of the rescored model on this set, as explained above.

5. Discussion

We have shown that the use of morphology-based LMs at different stages in an LVCSR system for Arabic leads to word error rate reductions. One drawback of the current approach is that the full potential of the FLM cannot be exploited directly since factored word representations are not supported by current decoders. Future work will focus on creating better interfaces between the decoder and factored language models, and on extending the current method by adding out-of-vocabulary words with probabilities assigned by morphological language models.

Acknowledgments

We would like to thank Jeff Bilmes for providing and supporting the FLM software, and Karim Darwish for providing the morphological analyzer.

This material is based on work funded 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.

6. References

[1] P. Geutner, "Using morphology towards better large-vocabulary speech recognition systems," in *Proceedings of ICASSP*, 1995, pp. 445–448.

[2] K. Çarki, P. Geutner, and T. Schultz, "Turkish LVCSR: towards better speech recognition for agglutinative languages," in *Proceedings of ICASSP*, 2000, pp. 3688–3691.

[3] E. Whittaker and P. Woodland, "Particle-based language modelling," in *Proceedings of ICSLP*, 2000.

[4] P. Ircing, P. Krebc, J. Hajic, S. Khudanpur, F. Jelinek, J. Psutka, and W. Byrne, "On large vocabulary continuous speech recognition of highly inflectional language - Czech," in *Proceedings of Eurospeech*, 2001, pp. 487–490.

[5] K. Kirchhoff et al., "Novel approaches to Arabic speech recognition - final report from the JHU summer workshop 2002," John-Hopkins University, Tech. Rep., 2002.

[6] J. A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proceedings of HLT/NACCL*, 2003, pp. 4–6.

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

[8] V. Digalakis and H. Murveit, "Genones: Optimizing the degree of mixture tying in a large vocabulary hidden markov model based speech recognizer," in *Proceedings IEEE ICASSP*, 1994, pp. I–537–540.

[9] K. Darwish, "Building a shallow Arabic morphological analyser in one day," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA, 2002.

[10] A. Stolcke et al., "Speech-to-text research at SRI-ICSI-UW," NIST RT-03 Spring Workshop, http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/sri+-rt03-stt.pdf, 2003.

[11] J. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[12] H. Kitano, "Designing neural networks using genetic algorithms with graph generation system," *Complex Systems*, pp. 461–476, 1990.

[13] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proceedings of the ARPA Workshop on Human Language Technology*, 1998, pp. 270–274.