A Hybrid Morphologically Decomposed Factored Language Models for Arabic LVCSR

Amr El-Desoky, Ralf Schlüter, Hermann Ney
Lehrstuhl für Informatik 6 – Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{desoky,schluter,ney}@cs.rwth-aachen.de

Abstract

In this work, we try a hybrid methodology for language modeling where both morphological decomposition and factored language modeling (FLM) are exploited to deal with the complex morphology of Arabic language. At the end, we are able to obtain from 3.5% to 7.0% relative reduction in word error rate (WER) with respect to a traditional full-words system, and from 1.0% to 2.0% relative WER reduction with respect to a non-factored decomposed system.

1 Introduction

Arabic language is characterized by a complex morphological structure where different kinds of prefixes and suffixes are appended to the word stems producing a very large number of inflectional forms. This leads to poor language model (LM) probability estimates, and thus high LM perplexities (PPLs) causing problems in large vocabulary continuous speech recognition (LVCSR). One successful approach to deal with this problem is to consider LMs including morphologically decomposed words. Another approach is to use the factored language models (FLMs) which are powerful models that combine multiple sources of information and efficiently integrate them via a complex backoff mechanism (Bilmes and Kirchhoff, 2003).

Morphological decomposition is successfully used for Arabic LMs in several previous works. Some are based on linguistic knowledge, and others are based on unsupervised methods. Some of the linguistic methods are based on the Buckwalter Arabic Morphological Analyzer (BAMA) like in (Lamel et al., 2008). Alternatively, in our previous work (El-Desoky et al., 2009), we use the Morphological Analyzer and Disambiguator for Arabic (MADA) (Habash and Rambow, 2007). On the other side, most of the unsupervised methods are based on the minimum description length principle (MDL) like in (Creutz et al., 2007).

Another type of models is the FLM, in which words are viewed as vectors of K factors, so that $w_t := \{f_t^{1:K}\}$. A factor could be any feature of the word such as morphological class, stem, root or even a semantic feature. An FLM is a model over factors, i.e., $p(f_t^{1:K}|f_{t-1}^{1:K}, f_{t-2}^{1:K}, ..., f_{t-n+1}^{1:K})$, which could be reformed as a product of probabilities of the form $p(f|f_1, f_2, ..., f_N)$. The main idea of the model is to backoff to other factors when some word n-gram is not observed in the training data, thus improving the probability estimates.

In this work we try to combine the strengths of morphological decomposition and factored language modeling. Therefore, language models with factored morphemes are used. For this purpose, the LM training data are processed such that full-words are decomposed into prefix-stem-suffix format with different added features. We compare our approach with the standard full-word, decomposed word, and factored full-word n-gram approaches.

2 Factorization and Decomposition

We use MADA 2.0 in order to perform morphological analysis and attach a complete set of morphological tags to Arabic words in context. From those tags

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 701–704,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

we derive three different features. Moreover, we derive a fourth feature based on the root of the word generated by ”Sebawai” (Darwish, 2002). The list of features is:

- **”W”** (Word): word surface form.
- **”L”** (Lexeme): word lexeme.
- **”M”** (Morph): morphological description.
- **”P”** (Pattern): word after subtracting root.

The LM training corpora are processed so that words are replaced by the factored representation as required by SRILM-FLM extensions (Kirchhoff et al., 2008). Then, word decomposition is performed based on MADA as described in our previous publication (El-Desoky et al., 2009).

3 FLM topologies

In order to obtain a good performance via FLMs, we need to optimize the FLM parameters: the combination of the conditioning factors, backoff path, and smoothing options. For this purpose, we use a Genetic Algorithm based FLM optimization tool (GA-FLM) developed by Kirchhoff (2006) which seeks to minimize the PPL over some held-out text. Furthermore, we apply some manual optimization to fine tune the FLM parameters. For memory limitations, we only use factors up to 2 previous time slots (tri-gram like models). Finally, we come up with a set of competing FLMs with rather close PPLs. In Table 1, we record the PPLs measured for some held-out text. The first column gives the combination of the parent factors. So that, FLM_1 corresponds to the model $P(W_t|W_{t-1}, W_{t-2})$, which is the FLM equivalent of the standard tri-gram LM (our baseline model), while FLM_2 & FLM_3 correspond to the model $P(W_t|W_{t-1}, M_{t-1}, L_{t-1}, P_{t-1}, W_{t-2})$, however FLM_4 & FLM_5 correspond to the model $P(W_t|W_{t-1}, M_{t-1}, L_{t-1}, W_{t-2}, M_{t-2}, L_{t-2})$. The ”gtmin” refers to the count threshold that is sufficient to have a language model hit at some node of the the backoff graph (for exact topologies, contact the first author). From Table 1, comparing PPLs (non-normalized) across factored and non-factored LMs, we see that using more factors other than the normal word could help decreasing the PPL. This is true for all the used types of vocabulary units.

	vocabulary		
FLM_x **parent factors**	**FW**	**PD**	**FD**
1: W1 W2 (baseline)	302.6	284.1	82.7
W1 M1 L1 P1 W2			
2: gtmin = 1	306.2	296.9	83.2
3: gtmin = 2-4	290.9	279.1	79.8
W1 M1 L1 W2 M2 L2			
4: gtmin = 1	300.2	291.1	83.6
5: gtmin = 2-4	294.5	283.7	81.1

Table 1: perplexities of the FLMs using vocabularies: (FW: 70k full-words; PD: partially decomposed with 20k ful-words + 50k morphemes; FD: 70k fully decomposed).

FLM_x **parent factors**	**WER [%]**
1: W1 W2 (baseline)	20.4
W1 M1 L1 P1 W2	
2: gtmin = 1	20.2
3: gtmin = 2-4	20.4
W1 M1 L1 W2 M2 L2	
4: gtmin = 1	**19.9**
5: gtmin = 2-4	20.3

Table 2: WERs using FLMs based on 70k full-words.

In order to select the best FLM topology, we run a simple one pass recognition for a small internal dev corpus derived from GALE data sets, consists of 40 minutes of audio data recorded during January to March 2007. The acoustic models are within-word tri-phone models trained using 1100h of audio material. The basic acoustic models are trained based on Maximum Likelihood (ML) method. Then, a discriminative training based on Minimum Phone Error (MPE) criterion is performed to enhance the models. A 70k full-words lexicon is used. The FLM training data consists of 206 Million running full-words. A standard bi-gram LM based on full-words is used to generate N-best lists, then N-best list rescoring is performed using the different FLM topologies shown in Table 1. We start by N = 1000-best down to 3-best sentences. Using N = 10 always gives the best results. The recognition WERs are recorded in Table 2. The least WER is obtained with FLM_4. We note that the best FLM does not correspond to the least PPL. This is because a higher ”gtmin” value causes more backoff in cases of insufficient data leading to better estimates. Therefore, we select FLM_4 for the coming experiments.

4 Experimental Setup

Our recognition system is close to the one described in section 3. However, we use within and across-word models at different recognition passes. In addition, we use 70k or 256k lexicon of full-words or partially decomposed words. Alternatively, we evaluate the results on the GALE 2007 development and evaluation sets (dev07: 2.5h; eval07: 4h). Our recognizer works in 3 passes. In the first pass, within-word acoustic models are used with no adaptation, along with a standard bi-gram LM to generate lattices, followed by a standard tri-gram or 4-gram LM rescoring of lattices. The second pass does the same, but it uses across-word models with Constrained Maximum Likelihood Linear Regression (CMLLR) adaptation. Then, a third pass with additional Maximum Likelihood Linear Regression (MLLR) adaptation is performed, using a standard bi-gram LM to generate lattices or N-best lists. Then, one of the following is performed: **1)** lattice rescoring using standard tri-gram or 4-gram LM, **2)** N-best list rescoring using FLMs based on full-words, partially or fully decomposed words.

5 Experiments

In this section, we record our recognition results for: **1)** systems based on full-words, and **2)** systems based on decomposed words. Also, we introduce additional results for larger lexicon sizes.

5.1 Systems Based on Full-words

In this section, we present the WERs of our recognition systems based on full-words. Where, during the search, we use a lexicon of 70K full-words. In the first 2 passes, we use a standard bi-gram LM to generate lattices, followed by a standard tri-gram LM rescoring of lattices. However, in the third pass, we generate both lattices and N-best lists based on the same bi-gram LM. The final lattices and N-best lists are rescored using different LMs as shown in Table 3. In case we perform N-best list rescoring with a FLM, the N-best lists are processed to produce factored representation, followed by partial or full decomposition as previously described in section 2.

It is clear from Table 3 that the least WER is achieved when using N-best list rescoring using a full-words based FLM. This gives an absolute improvement of 0.8% (about 4.8% relative) compared to the standard tri-gram lattice rescoring. On the other hand, we have 0.6% absolute improvement (about 3.7% relative) compared to the standard 4-gram lattice rescoring. Decomposition does not help in this case. This is because the original N-best lists are generated in full-words format, whose decomposition might not lead to better LM scores. For this reason, we expect that it is better to start with a decomposed LM for lattice and N-best generation.

LM rescoring (3^{rd} **pass)**	**Dev07 [%]**
tri-gram lattice resc. (baseline)	16.5
4-gram lattice resc.	16.3
N-best FLM resc.:	
+ FW (original N-best)	**15.7**
+ PD (decomposed N-best)	15.8
+ FD (decomposed N-best)	16.0

Table 3: WERs for 70k full-words systems.

LM rescoring (3^{rd} **pass)**	**Dev07 [%]**
tri-gram lattice resc. (baseline)	14.7
4-gram lattice resc.	14.5
N-best FLM resc.:	
+ FW (re-joint N-best)	14.6
+ PD (original N-best)	**14.3**
+ FD (decomposed N-best)	14.4

Table 4: WERs for 70k partially decomposed systems (20k full-words + 50k morphemes).

5.2 Systems Based on Decomposed Words

This section introduces the WERs of our systems based on decomposed words. We use a similar setup as in section 5.1. However, we use a lexicon and a bi-gram LM based on a 70k partially decomposed words (20k full-words + 50k morphemes). Table 4 presents the results. As expected, we get the best WER when using N-best list rescoring with a FLM based on partially decomposed words. An absolute improvement of 0.4% (2.7% relative) is achieved compared to the new baseline. Compared to the old baseline of Table 3, we get an absolute improvement of 2.2% (13.3% relative).

5.3 Larger Lexicon Sizes

Now, we increase the size of our lexicon to 256k partially decomposed words (20k full-words + 236k

System	Dev07 [%]	Eval07 [%]
traditional full-words	14.9	16.5
partially decomposed		
+ 4-gram lat. resc. (baseline)	14.2	16.1
+ N-best FLM resc.:		
+ FW (re-joint N-best)	14.1	-
+ PD (original N-best)	**13.9**	**15.9**
+ FD (decomposed N-best)	14.0	-

Table 5: WERs for 256k full-words, and partially decomposed systems (20k full-words + 236k morphemes).

	70k vocabularies			256k vocabularies		
Corpus	**FW**	**PD**	**FD**	**FW**	**PD**	**FD**
Dev07	3.65	1.33	0.75	1.36	0.51	0.24
Eval07	4.82	1.94	1.13	1.85	0.64	0.41

Table 6: OOVs [%] of the used vocabularies.

morphemes). In addition, we use a standard 4-gram LM for rescoring the bi-gram lattices in the first 2 passes. To complete our comparisons, we record the WERs using traditional 256k full-words lexicon, standard bi-gram search, and standard 4-gram LM for lattice rescoring, with no decomposition or factorization. In Table 5, we see that the improvement persists for the larger lexicon. Compared to the new baseline, the 256k decomposed system achieves WER reductions of [dev07: 0.3% absolute (2.1% relative); eval07: 0.2% absolute (1.2% relative)] when using N-best list rescoring with a FLM based on partially decomposed words. Moreover, it improves over the traditional full-words by [dev07: 1.0% absolute (6.7% relative); eval07: 0.6% absolute (3.6% relative)]. The out-of-vocabulary rates (OOVs) are given in Table 6. It is worth noting that using fully decomposed lexicons as well as higher order LMs could not improve WERs, this we previously proved in (El-Desoky et al., 2009).

6 Conclusions

We have introduced a hybrid approach to Arabic language modeling. Our approach combines the strengths of both morphological decomposition and factored language modeling. Thus, we have used language models with factored morphemes. We have compared our approach to traditional approaches like: standard full-word n-grams, standard decomposed n-grams, and full-word based factored language models. Finally, we could achieve some improvements over all the traditional approaches. Nevertheless, we have only considered the use of factored language models in the rescoring phase.

Acknowledgments

This material is based upon work supported by the DARPA under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

J. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proc. Human Language Technology Conf. of the North American Chapter of the ACL*, volume 2, pages 4 – 6, Edmonton, Canada, May.

M. Creutz, T. Hirsimki, M. Kurimo, A. Puurula, J. Pylkknen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraclar, and A. Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1), December.

K. Darwish. 2002. Building a shallow Arabic morphological analyzer in one day. In *ACL workshop on Computational approaches to semitic languages*, Philadelphia, PA, USA, July.

A. El-Desoky, C. Gollan, D. Rybach, R. Schlüter, and H. Ney. 2009. Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR. In *Interspeech*, pages 2679 – 2682, Brighton, UK, September.

N. Habash and O. Rambow. 2007. Arabic diacritization through full morphological tagging. In *Proc. Human Language Technology Conf. of the North American Chapter of the ACL*, volume Companion, pages 53 – 56, Rochester, NY, USA, April.

K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke. 2006. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech and Language*, 20(4):589 – 608, October.

K. Kirchhoff, J. Bilmes, and K. Duh. 2008. Factored language model tutorial. Technical report, Department of Electrical Engineering, University of Washington, Seattle, Washington, USA, February.

L. Lamel, A. Messaoudi, and J.L Gauvain. 2008. Investigating morphological decomposition for transcription of Arabic broadcast news and broadcast conversation data. In *Interspeech*, volume 1, pages 1429 – 1432, Brisbane, Australia, September.