INVESTIGATIONS ON THE USE OF MORPHEME LEVEL FEATURES IN LANGUAGE MODELS FOR ARABIC LVCSR

Amr El-Desoky Mousa, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition – Computer Science Department
RWTH Aachen University, 52056 Aachen, Germany

ABSTRACT

A major challenge for Arabic Large Vocabulary Continuous Speech Recognition (LVCSR) is the rich morphology of Arabic, which leads to high Out-of-vocabulary (OOV) rates, and poor Language Model (LM) probabilities. In such cases, the use of morphemes rather than full-words is considered a better choice for LMs. Thereby, higher lexical coverage and less LM perplexities are achieved. On the other side, an effective way to increase the robustness of LMs is to incorporate features of words into LMs. In this paper, we investigate the use of features derived for morphemes rather than words. Thus, we combine the benefits of both morpheme level and feature rich modeling. We compare the performance of stream-based, class-based and Factored LMs (FLMs) estimated over sequences of morphemes and their features for performing Arabic LVCSR. A relative reduction of 3.9% in Word Error Rate (WER) is achieved compared to a word-based system.

Index Terms— language model, morpheme, stream-based, class-based, factored

1. INTRODUCTION

Arabic is considered one of the morphologically complex languages. In fact, it is a highly inflected Semitic language. Arabic words are derived from roots which have, in most cases, three letters by applying templates to get stems and then attaching prefixes and suffixes to obtain a very large number of different surface forms. This huge lexical variety causes data sparsity problems and leads to high OOV rates and high LM perplexities. A traditional approach to overcome this problem is to use a very large recognition vocabulary. Yet, still relatively high OOV rates are obtained. Moreover, the speech recognition system suffers from high resource requirements such as CPU time and memory.

An alternative approach is to use morpheme-based LMs in order to lower the OOV rate and perplexity, reduce data sparsity, decrease resource requirements and achieve lower WERs. Normally, morphemes are generated by applying morphological decomposition to words. In some cases morphological decomposition is based on linguistic knowledge as in [1], and in other cases it is based on unsupervised approaches like in [2]. Some of the linguistic methods make use of the Buckwalter Arabic Morphological Analyzer (BAMA) like in [3]. Alternatively, in our previous work [4], we use the Morphological Analyzer and Disambiguator for Arabic (MADA) [5].

Another approach to overcome the data sparseness and to reduce the dependence of the traditional word-based LMs on the discourse domain, is to assign proper features (classes) to words and build LMs over those features. This yields better smoothing and, hopefully, better generalization to unseen word sequences. The features can be generated based on linguistic methods as in [6], or via data driven approaches as in [7]. Possible approaches for incorporating word features into LMs are: stream-based LMs [8], class-based LMs [9] and factored LMs [10]. In stream-based LMs, a normal back-off N-gram model is built over a stream of word classes, where the stream consists of sequences of a single class type called class stream. However, a class-based LM combines the N-gram model over classes with the probability distribution of words in classes in order to better estimate smoothed probabilities of word sequences. On the other side, an FLM uses a complex backoff mechanism across multiple features in the same model in order to obtain robust probability estimates. All these types of LMs can be used for LM rescoring of the hypothesized N-best lists.

This paper presents a technique that attempts to gain the benefits from the incorporation of features into LMs, while in the same time retain the advantages of using morpheme-based LMs. This is accomplished by generating features on the level of morphemes rather than full-words. In a previous work [11], we investigated the use of morpheme level FLMs for Arabic LVCSR. Here, we compare the performance of FLMs to stream-based and class-based LMs all estimated on morpheme level. Moreover, we examine the interpolation of normal N-gram LMs with class-based LMs, and the combination of different N-best scores obtained from different LMs.

2. METHODOLOGY

2.1. Data processing and feature derivation

Our LM training data is processed using MADA 2.0 tool. MADA is a morphological analyzer and disambiguator tool for Arabic, which is built over BAMA [5]. It is able to asso-

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

ciate a complete set of morphological tags with each word in context. These tags are used to produce robust diacritization and tokenization for the words. Based on this tokenization, we produce decomposed words in the form of *" prefix+ stem +suffix"*. The '+' sign is used as a marker for full-word recombination. For a detailed description of the decomposition process and constraints, see our previous publication [4].

Starting from the MADA morphological tags along with the decomposition, we derive two different features namely, *"Lexeme"* and *"Morph"*. Lexeme is an abstraction over the inflected words that groups together all word forms that differ only in one of the morphological categories such as number or gender. Morph is the morphological description of the word; it includes the word Part-of-speech (POS) and indicates whether a conjunction, particle, article or a clitic are agglutinated to the word. In addition, a third feature called *"Pattern"* is derived by subtracting *root* letters from the word. The root is generated by *"Sebawai"* tool [12]. Finally, The LM training corpus is re-written so that every word is replaced by a vector of features as in the form: {*W-<word>:L-<lexeme>:M-<morph>:P-<pattern>*}. The same features are similarly defined for morphemes as well as words. A sequence of individual vector components defines a feature stream (class stream). A vector example in the case of words is: *wAl$rqyp* → {*W-wAl$rqyp:M-conj+art+AJ-FEM-SG:L-$rqy:P-wAlCCCyp*}. However, in the case of morphemes: *wAl$rqyp* → {*W-wAl+:M-conj+art:L-wAl+:P-NUL*} {*W-$rqyp:M-AJ-FEM-SG:L-$rqy:P-CCCyp*}; given that *root(wAl$rqyp) = $rq*. From these examples we can see that a careful handling of word morphological features could help to produce valid features for morphemes, these are called *morpheme level features*.

2.2. Stream-based language models

Given a sequence of words $W = w_1, w_2, ..., w_M$, a standard N-gram LM is expressed as:

$$p(w_1, w_2, ..., w_M) \approx \prod_{i=1}^{M} p(w_i | w_{i-N+1}^{i-1}) \qquad (1)$$

If this model is built over decomposed words (morphemes), then it is called a *morpheme level model*. However, instead of building the N-gram LM over sequences of words or morphemes, we could build the model over sequences of some selected class stream defined for words or morphemes like sequences of lexemes, morphs or patterns. Similar to Equation 1, given a sequence of classes $c_1 c_2, ..., c_M$, an N-gram stream-based model is:

$$p(c_1, c_2, ..., c_M) \approx \prod_{i=1}^{M} p(c_i | c_{i-N+1}^{i-1}) \qquad (2)$$

Such models can be used for N-best list rescoring. Therefore, the hypothesized N-best sentences are mapped to the corresponding class stream suitable for the underlying model.

2.3. Class-based language models

The class-based LMs are initially described in [9]. Assuming multiple (ambiguous) class membership, where a word can be a member of multiple classes, an example bigram class-based LM is shown in Equation 3, where the word is denoted by w and c is the class. An analogous model could be estimated for morphemes and their features.

$$p(w_i | w_{i-1}) = \sum_{c_i, c_{i-1}} p(w_i | c_i) p(c_i | c_{i-1}) p(c_{i-1} | w_{i-1}) \quad (3)$$

Normally, the standard word-based LMs are performing better in capturing the relations between words for in-domain text. Thus, an effective way to retain the advantages of both word-based and class-based LMs is to combine them. the combination may rely on backing-off or linear interpolation. Here, we use linear interpolation expressed as:

$$p(W) = \lambda p_w(W) + (1 - \lambda) p_c(W) \qquad (4)$$

where W is the word sequence, $p_w(W)$ is the word-based probability, $p_c(W)$ is the class-based probability, and λ is the interpolation weight optimized on some development data.

2.4. Factored language models

FLMs were first introduced in [10]. In an FLM, a word is viewed as a vector of K parallel factors, so that $w_t := \{f_t^1, f_t^2, ..., f_t^K\}$. A factor could be the word itself or any feature of the word such as morphological class, stem, root or even a data driven class or a semantic feature. A probabilistic LM is estimated over both words and their factors. In other words, the objective of the FLM is to produce a statistical model over the individual factors, namely $p(f_{1:T}^{1:K})$. Using an N-gram-like formula, the goal is produce accurate models of the form $p(f_t^{1:K} | f_{t-1}^{1:K}, f_{t-2}^{1:K}, ..., f_{t-n+1}^{1:K})$ [13]. This model represents the interdependencies among features of words both across time and within word. It uses a complex backoff mechanism across multiple features. The model backs off to other factor combination when some word N-gram is not sufficiently observed in the training data, which improves the probability estimates. In our experiments, we use an FLM corresponding to the model $P(W_t | W_{t-1}, M_{t-1}, L_{t-1}, W_{t-2}, M_{t-2}, L_{t-2})$, where W is word, M is morph, L is lexeme. The details of how the model is created and optimized are found in our previous work [11].

2.5. Score combination

The score used for re-ranking the N-best hypotheses is normally a weighted combination of several components: the acoustic score, the LM score and the number of words. However, scores from various LMs can be added, such as the scores from various stream-based, class-based LMs and FLMs. The final score for each hypothesis can be computed as a log-linear combination of the invoked scores. The weights of this combination can be optimized to minimize the WER [8]. For the weight optimization, we use **"Amoeba"** search which is available in SRILM toolkit [14].

3. EXPERIMENTAL SETUP

Our acoustic models (AMs) are triphone models trained on 1100h of audio material taken from two domains: Broadcast News (BN) and Broadcast Conversation (BC). The basic AMs are trained using Maximum Likelihood (ML) method. Then, a discriminative training based on Minimum Phone Error (MPE) criterion is performed to enhance the models [15]. Our LM training corpora have around 206 Million running words including data from Agile Arab text, FBIS, TDT4 and GALE BN and BC data. For word level systems, a lexicon of 70k full-words is used. However, for morpheme-based systems, 70k or 256k lexicons are used while preserving the 20k most frequent full-words without decomposition [4]. Different types of LMs are estimated as described in Section 2. All models are smoothed via modified Kneser-Ney smoothing using SRILM toolkit [14]. Our speech recognizer works in 3 passes. In the first pass, within-word AMs are used without adaptation. The second pass uses across-word AMs with Constrained Maximum Likelihood Linear Regression (CMLLR) adaptation. Then, a third pass with additional Maximum Likelihood Linear Regression (MLLR) adaptation is performed. In each pass, a word-based or morpheme-based bigram LM is used to construct the search space and to produce lattices then these lattices are rescored using a word-based or morpheme-based 4-gram LM correspondingly. Additionally, in the third pass, we produce a set of N-best lists which are rescored with different LMs or a combination of them as described in Section 2. The recognition performance is evaluated on the GALE 2007 dev and eval sets [dev07: 2.5h; eval07: 4h]. During score combination, the weights of LMs are optimized over dev07 corpus.

4. EXPERIMENTS

In Table 1, the column labeled "WB" shows the WERs of a 70k word-based system for dev07 corpus after the third pass rescoring. N-best sentences with *N = 5 to 25* are generated and processed as illustrated in Section 2 so as to produce a representation suitable for the rescoring LM. Without score combination, the best WER is obtained using the FLM model previously given in Section 2.4 [11]. After the score combination (no. 11) of lexeme, morph and pattern class-based models each interpolated with a word model in addition to the FLM, a little more improvement of [dev07: 3.68% relative (0.6% absolute)] is achieved compared to the baseline lattice rescoring via a word-based LM. The column labeled "MB" shows the WERs of a 70k morpheme-based system for dev07 corpus. The performance of the morpheme-based system is better than the word-based system (WB column). We achieve a WER reduction of [dev07: 11% relative (1.78% absolute)] due to the use of morphemes. This is mainly caused by the better lexical coverage (OOV rate is 1.33% compared to 3.65%). Without score combination, the best improvement is obtained using a lexeme class-based LM interpolated with a morpheme-based LM. The WERs using the interpolated models (no. 7, 8, 9) and the FLM (no. 10) are almost equal. The score combination (no. 11) yields a little better WER reductions of [dev07: 1.86% relative (0.27% absolute)] compared to the baseline lattice rescoring via a morpheme-based LM. Also, generally, the class-based models perform better than the stream-based models.

Table 1. *WERs [%] for dev07 [WB: a 70k word-based system, OOV rate = 3.65%; MB: 70k morpheme-based system (20k full-words + 50k morphemes), OOV rate = 1.33%; w/m: word- or morpheme-based model].*

3^{rd} **pass**	**Dev07**	
	WB	**MB**
4-gram lattice rescoring (baseline)	16.30	14.52
N-best rescoring:		
1. stream-based: lexeme	15.99	14.54
2. stream-based: morph	16.43	14.99
3. stream-based: pattern	16.58	14.81
4. class-based: lexeme	16.12	14.49
5. class-based: morph	16.12	14.61
6. class-based: pattern	16.19	14.90
7. w/m + class-based: lexeme	15.92	**14.27**
8. w/m + class-based: morph	15.90	14.29
9. w/m + class-based: pattern	15.94	14.33
10. FLM: word, lexeme, morph	**15.74**	14.32
11. combination: 7 + 8 + 9 + 10	**15.70**	**14.25**

Table 2 shows the WERs of a 256k morpheme-based system on dev07 and eval07 corpora. For completeness, the first row of Table 2 shows the WERs of a 256k word-based system after a lattice rescoring via a word-based LM. Without score combination, the best WER is achieved using the FLM. Using score combination, WER reductions of [dev07: 2.11% relative (0.3% absolute); eval07: 1.43% relative (0.23% absolute)] are obtained compared to the baseline lattice rescoring via a morpheme-based LM. On the other side, this achieves WER reductions of [dev07: 6.71% relative (1% absolute); eval07: 3.94% relative (0.65% absolute)] compared to the standard word-based 256k system. The obtained performance improvements indicate an improvement in LM probability estimation due to the use of morpheme-level features.

5. CONCLUSIONS

We investigated the use of morpheme level features for Arabic LMs. We compared the performance of stream-based, class-based LMs and FLMs in an Arabic LVCSR task. We verified that those feature-based LM techniques could be used in morpheme domain as efficient as in word domain. Thereby, we retain the advantages of morpheme-based LMs in addition to the benefits of feature rich modeling. Morpheme-based LMs achieve better lexical coverage and reduce the problem of data sparsity. While the feature-based models try to achieve better generalization to unseen word sequences. We used different types of morphological features derived from MADA morphological analyzer. In most cases, FLMs provide better

Table 2. *WERs [%] for a 256k morpheme-based system (20k full-words + 236k morphemes), OOV rate = [dev07: 0.51%, eval07: 0.64%]; first row gives WER [%] for a 256k word-based system for completeness.*

3^{rd} **pass**	**Dev07**	**Eval07**
word-based 4-gram	14.90	16.50
morpheme-based 4-gram (baseline)	14.20	16.10
N-best rescoring:		
1. stream-based: lexeme	14.25	16.14
2. stream-based: morph	14.70	16.31
3. stream-based: pattern	14.53	16.37
4. class-based: lexeme	14.20	16.01
5. class-based: morph	14.27	16.08
6. class-based: pattern	14.56	16.34
7. morpheme + class-based: lexeme	13.93	15.89
8. morpheme + class-based: morph	13.94	15.96
9. morpheme + class-based: pattern	13.99	16.04
10. FLM: word, lexeme, morph	**13.90**	**15.87**
11. combination: 7 + 8 + 9 + 10	**13.90**	**15.85**

performance compared to other models. Moreover, using a combination of different LM scores during the N-best rescoring could improve the performance a little bit more.

6. ACKNOWLEDGMENTS

This work was partly realized under the Quaero Programme, funded by OSEO, French State agency for innovation.

7. REFERENCES

[1] G. Choueiter, D. Povey, S. Chen, and G. Zweig, "Morpheme-based language modeling for Arabic LVCSR," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Toulouse, France, May 2006, pp. 1053 – 1056.

[2] J. Goldsmith, "Unsupervised learning of the morphology of a natural language," *Computational linguistics*, vol. 27, no. 2, pp. 153 – 198, Jun. 2001.

[3] L. Lamel, A. Messaoudi, and J. Gauvain, "Investigating morphological decomposition for transcription of Arabic broadcast news and broadcast conversation data," in *Interspeech*, vol. 1, Brisbane, Australia, Sep. 2008, pp. 1429 – 1432.

[4] A. El-Desoky, C. Gollan, D. Rybach, R. Schlüter, and H. Ney, "Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR," in *Interspeech*, Brighton, UK, Sep. 2009, pp. 2679 – 2682.

[5] N. Habash and O. Rambow, "Arabic diacritization through full morphological tagging," in *Proc. Human Language Tech. Conf. of the North American Chapter of the ACL*, vol. Companion, Rochester, NY, USA, Apr. 2007, pp. 53 – 56.

[6] G. Maltese, P. Bravetti, H. Crépy, B. Grainger, M. Herzog, and F. Palou, "Combining word- and class-based language models: A comparative study in several languages using automatic and manual word-clustering techniques," in *Proc. European Conf. on Speech Communication and Technology*, Aalborg, Denmark, Sep. 2001, pp. 21 – 24.

[7] T. Matsuzaki, Y. Miyao, and J. Tsujii, "An efficient clustering algorithm for class-based language models," in *Proc. Human Language Tech. Conf. of the North American Chapter of the ACL*, vol. 4, Edmonton, Canada, May 2003, pp. 119 – 126.

[8] K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Computer Speech and Language*, vol. 20, no. 4, pp. 589 – 608, Oct. 2006.

[9] P. Brown, P. deSouza, R. Mercer, V. D. Pietra, and J. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, pp. 467 – 479, 1992.

[10] J. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proc. Human Language Tech. Conf. of the North American Chapter of the ACL*, vol. 2, Edmonton, Canada, May 2003, pp. 4 – 6.

[11] A. El-Desoky, R. Schlüter, and H. Ney, "A hybrid morphologically decomposed factored language models for Arabic LVCSR," in *Proc. Human Language Tech. Conf. of the North American Chapter of the ACL*, Los Angeles, CA, USA, Jun. 2010, pp. 701 – 704.

[12] K. Darwish, "Building a shallow Arabic morphological analyzer in one day," in *ACL workshop on Computational approaches to semitic languages*, Philadelphia, PA, USA, Jul. 2002.

[13] K. Kirchhoff, J. Bilmes, and K. Duh, "Factored language model tutorial," Department of Electrical Engineering, University of Washington, Seattle, Washington, USA, Tech. Rep., Feb. 2008.

[14] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. Int. Conf. on Spoken Language Processing*, vol. 2, Denver, Colorado, USA, Sep. 2002, pp. 901 – 904.

[15] D. Vergyri, A. Mandal, A. Stolcke, J. Zheng, M. Graciarena, D. Rybach, C. Gollan, R. Schlüter, K. Kirchhoff, A. Faria, and N. Morgan, "Development of the SRI/Nightingale Arabic ASR system," in *Interspeech*, vol. 1, Brisbane, Australia, Sep. 2008, pp. 1437 – 1440.