ON THE USE OF MORPHOLOGICAL ANALYSIS FOR DIALECTAL ARABIC SPEECH RECOGNITION

Mohamed Afify, Ruhi Sarikaya, Hong-Kwang Jeff Kuo, Laurent Besacier, and Yuqing Gao

IBM T.J. Watson Research Center
1101 Old Kitchawan Road, Yorktown Heights, NY, 10598

ABSTRACT

Arabic has a large number of affixes that can modify a stem to form words. In automatic speech recognition (ASR) this leads to a high out-of-vocabulary (OOV) rate for typical lexicon size, and hence a potential increase in WER. This is even more pronounced for dialects of Arabic where additional affixes are often introduced and the available data is typically sparse. To address this problem we introduce a simple word decomposition algorithm which only requires a text corpus and a predefined list of affixes. Using this algorithm to create the lexicon for Iraqi Arabic ASR results in about 10% relative improvement in word error rate (WER). Also using the union of the segmented and unsegmented vocabularies and interpolating the corresponding language models results in further WER reduction. The net WER improvement is about 13% relative.

Index Terms: Speech recognition, language modeling, Dialectal Arabic, morphological analysis, prefixes and suffixes.

1. INTRODUCTION

Arabic is a morphologically rich language. Prefixes and suffixes, affixes for short, augment word stems to form words. Moreover, word stems are derived by applying some predefined patterns to roots. For example, the word stem lAEb[1]("player" in English) can be modified using the prefix Al and the suffixes An and wn to create the three words AllAEb (the player), AllAEbAn (the two players), and AllAEbwn (the players). The word stem lAEb itself is derived from the root lEb (to play) by applying a certain pattern. Thus, the root lEb in this example gives rise to four different words in addition to many others (not listed) by applying different prefixes and suffixes and also different patterns.

For automatic speech recognition (ASR) a word is defined as a string of characters separated by space. Hence, word definition is not aware of morphological relationships between different words. In practice this leads to a high out-of-vocabulary (OOV) rate. For example, 64K word lexicons which typically lead to around a 0.5% OOV rate for English result in about 5% OOV for Arabic and hence lead to higher word error rate during recognition. One direct solution is to increase the size of the recognition lexicon. For example, for modern standard Arabic (MSA) it was found that to achieve the typical 0.5% OOV rate a lexicon size around 500K word is needed; about an order of magnitude more than that required for English.

Arabic has several dialects which are used in every day communications in different countries. These dialects significantly differ from MSA which is used in newspapers and formal communications. The above problem is even more pronounced for dialectal Arabic due to the following reasons:

- Additional prefixes, and sometimes suffixes, are informally introduced during the everyday use of language.
- The amount of text data available for dialectal Arabic is usually much smaller than that for MSA, and hence it is not clear how to increase the vocabulary size to reduce OOV[2].
- Even if vocabulary is increased using some means the sparse text resources will lead to poor estimates of the language model probabilities, and hence may hurt performance on a different front.

In this paper we address the use of morphological segmentation for dialectal Arabic speech recognition. We only address this issue for prefixes and/or suffixes, but do not consider the problem of deriving stems from roots by applying patterns. Hence, in the above example, we would like to derive that the words lAEb and AllAEb have the same stem, but no attempt is done to infer that lAEb is derived from the root lEb by applying some pattern. The latter information will contribute to further reduction of the vocabulary size, but applying it needs significant deviation from typical ASR structure. It is worth noting that several languages, e.g. German, Dutch, Hebrew, and Turkish, share, to certain extents, this morphological richness. Hence, the same principle can be applied to these languages. However, the practice may be different and is very likely to be language dependent.

There were several previous attempts to use morphological processing for ASR of morphologically rich languages. In [1, 2, 3, 4] similar morphological decomposition, sometimes decomposition followed by composition, is used to define the vocabulary for ASR systems in German and Dutch. In all cases either a degradation in performance or a very small improvement is observed. In contrast, we show significant improvement by using morphological decomposition. This might be due to the difference in the nature of the language and the details of the decomposition process. On the other hand, the works [5, 6, 7, 8] use morphological processing in dialectal Arabic for better smoothing of the language model probabilities instead of redefining the vocabulary. These techniques while theoretically interesting in some cases, either lead to small improvement or degradation in performance. The recent work [9] uses morphological segmentation and obtains significant improvement for MSA. In particular, it was shown that using morphological segmentation it is possible to obtain results

[1] We use Buckwalter transliteration to represent Arabic words.

[2] Usually vocabulary is constructed from the unique words in the corpus.

similar to a 300K vocabulary using a 64K lexicon. Our work is similar in spirit to the latter except for the following differences:

- We work on dialectal (Iraqi) Arabic and hence use a modified list of affixes, especially prefixes.
- Our decomposition algorithm is considerably simpler.
- We show that we obtain significant improvement over our full unsegmented vocabulary (around 90K) using morphological processing. This is due to the sparse amount of data available for dialect which does not allow defining larger vocabularies.
- We perform experiments, in addition to vocabulary decomposition, on language model smoothing using the decomposed vocabulary. Nice improvements are also seen in these experiments.

Also another interesting recent work [10] uses a statistical segmentation algorithm [11] to perform word decomposition (possibly with multiple prefixes and/or suffixes per word) and use a graph to encode permissible word structure in Arabic that is composed with the original graph. Nice improvements were shown for vocabulary around 64K. The major difference between this approach and our work is in the segmentation algorithm (statistical vs knowledge based) and in allowing multiple prefixes and suffixes. We will give brief results on using a similar approach in Section 4.

The paper is organized as follows. In Section 2 we describe the word segmentation algorithm and show how it is applied to vocabulary selection. The structure of our speech recognition system and a description of both the acoustic and language model training data are given in Section 3. Section 4 gives the experimental results, where it is shown that using morphological segmentation leads to significant improvement over conventional word definition. Finally, conclusions are drawn in Section 5.

2. WORD SEGMENTATION ALGORITHM

Starting from given lists of prefixes and suffixes the goal of the word segmentation algorithm is to decompose each word in the vocabulary into one of the three forms: prefix-stem, stem-suffix, prefix-stem-suffix, or to leave it unchanged. Prefixes and suffixes in Arabic are composite, i.e. word beginnings(endings) can admit multiple prefixes(suffixes). For example, the word syktbh (meaning "he will write it") has multiple prefixes and one suffix and can be ultimately decomposed into s-y-ktb-h. However, initial experimentation with this approach led to poor recognition performance. One explanation is that allowing multiple prefixes and/or suffixes in the resulting words will give rise to sequences of prefixes and suffixes in the language model data and will accordingly lead to large insertion rate in the resulting decoded output. For this reason we decided to use single prefixes (suffixes) in our list. In the case we feel that a compound prefix (suffix) is useful we add it as a separate entry in the corresponding list.

The difficulty about blind segmentation is that sometimes the beginning (ending) part of a word agrees with a prefix(suffix), and hence blind segmentation will lead to illegitimate Arabic stems. For example, the word AlqY (threw in English) has its initial part agreeing with the popular prefix Al, and thus blind segmentation will lead to the segmentation Al-qY and hence to the invalid stem qY. To try to avoid this situation we employed the following segmentation algorithm. Using the given set of prefixes and suffixes, a word is first blindly chopped to one of the three forms prefix-stem, stem-suffix, or prefix-stem-suffix. This segmentation is accepted if the following three rules apply:

- The resulting stem is longer than two characters in length.
- The resulting stem is accepted by the Buckwalter morphological analyzer.
- The resulting stem exists in the original dictionary.

The first rule is very simple to apply and eliminates many of the illegitimate segmentations. The second rule simply means that the word is a valid Arabic stem, given that the Buckwalter morphological analyzer covers all words in the Arabic language. Unfortunately, the fact that the stem is a valid Arabic stem does not always imply that the segmentation is valid. This is especially true for unvowelized text. For example, for the word AlgyA ("both canceled") the segmentation Al-gyA is not valid but the stem will be accepted by the Buckwalter morphological analyzer. The third rule, while still not offering such guarantee, simply prefers keeping the word intact if its stem does not occur in the lexicon.

In our implementation we used a set of prefixes and suffixes for dialectal Iraqi. This list is given below:

- Prefix list: {chAl,bhAl,lhAl,whAl,wbAl,wAl,bAl,hAl,EAl, fAl,Al,cd,ll,b,f,c,d,w}.
- Suffix list: {thmA,tynA,hmA,thA,thm,tkm,tnA,tny,whA,whm wkm,wnA,wny,An,hA,hm,hn,km,kn,nA,ny,tm,wA,wh,wk, wn,yn,tk,th,h,k,t,y}.

There was no optimality criterion or even frequency considerations considered when selecting these affixes, only our intuition of their adequacy for dialectal Iraqi Arabic. These lists differ from those for MSA [9, 8] by adding prefixes and suffixes that are particular to Iraqi Arabic. In addition, we found in preliminary experiments that keeping the top-N frequent decomposable words intact led to better performance. A value of N=5000 was experimentally found to work well in practice.

By applying the above algorithm to an initial lexicon we can derive a map between each original word and its segmentation. This map can, in turn, be used to tokenize the existing corpus. Accordingly, a new lexicon consisting of affixes, stems, and unsegmented words can be obtained. This lexicon can be used in the speech recognition system in the same way as the original lexicon. In initial experiments we attempted applying the same type of tokenization to the acoustic training data and re-training the acoustic model but no significant performance difference was noticed from using the original acoustic model. Hence, this idea was not further pursued.

Using the above tokenization it should be pointed out that the ASR decoded output will be segmented in the sense that the output will contain prefixes and suffixes that should be glued to the following or previous word to form meaningful words. To facilitate such gluing we marked each prefix and suffix with a -, e.g. we have prefix Al- or suffix -yn. We used two gluing schemes. The first is very simple and just sticks any word that starts(ends) with a - to the previous(following) word. The second tries to apply some constraints to prevent sequences of affixes and to ensure that these affixes are not attached to words that start(end) with a prefix(suffix). No noticeable difference was seen between the two approaches.

3. SYSTEM ARCHITECTURE

This section presents the architecture of our speech recognition system including acoustic and language model training, and the

decoding strategy. It also gives a brief description of the acoustic and language model training data.

The speech data for recognition is sampled at 16kHz or 22kHz, and all the data is down-sampled to 16 kHz. Feature vectors are computed every 10 ms for the bandwidth between 300Hz and 7.6kHz. First, 24 dimensional mel frequency cepstrum coefficients (MFCC) are calculated. The MFCC features are then mean normalized, and 9 vectors are stacked leading to a 216-dimensional parameter space. The feature space is finally reduced to 40 dimensions using a combination of linear discriminant analysis (LDA), and maximum likelihood linear transformation (MLLT) [15]. This 40-dimensional space is used for both training and decoding.

During acoustic model training the feature vectors are first aligned to arcs (HMM states). A decision tree is then grown for each arc using a set of questions about context. The feature vectors corresponding to each leaf are used to build a Gaussian mixture model for this leaf. The set of leaf Gaussian mixture models are then re-estimated using several iterations of forward-backward training. The Gaussian prototypes are finally refined using minimum phone error (MPE) training [16]. After estimating the Gaussian distributions, rank distributions for each leaf are calculated using the resulting Gaussian mixture models [17]. These rank distributions are used to calculate acoustic scores in the decoding stage.

The acoustic model uses graphemes as the basic acoustic-phonetic units. There are 33 graphemes representing speech and silence. (To alleviate data sparseness, some acoustically similar graphemes have been classed together.) All words in the training and decoding lexicons are transcribed in terms of these graphemes. For MSA it was shown that, e.g. [13], using phonetic models obtained by adding the short vowels to the acoustic transcripts and decoding lexicon lead to improved performance over grapheme models. We were not able at present to observe similar improvements for dialectal Arabic, and hence use grapheme models. The training data consists of about 200 hours of dialectal Iraqi Arabic collected in the context of a speech to speech translation project. Applying the training algorithm outlined above results in about 2K leaves and 60K Gaussians.

The language model uses standard trigrams. The models are trained using deleted interpolation[12]. N-gram models are interpolated with lower order distributions, where interpolation weights are estimated using held-out data comprising about 10% of the training corpus. The training corpus consists of about 2M words also collected in the context of speech to speech translation of Iraqi Arabic.

The search uses a stack decoder [18]. The acoustic processor computes the rank probabilities of each feature vector. These probabilities together with the language model probabilities are employed in the search to find the most likely word sequence. The stack decoder makes use of a fast match. This fast match selects a set of words that are passed to the detailed match. The use of both the fast match, and a quantization-based fast labeling scheme lead to a very efficient search algorithm.

It is worth pointing out that the system acts as the speech recognition module in our speech to speech translation system [14] which operates on both laptop and hand-held devices. In the case of hand-held devices a slightly scaled down version of the acoustic models and language models are used but keeping the same architecture.

4. EXPERIMENTAL RESULTS

In this section we give experimental results of using the segmentation algorithm described above for dialectal Iraqi speech recognition. The LM training data has about 90K unique words. The segmentation algorithm of section 2 is applied to this lexicon. This leads to about a 60K lexicon consisting of prefixes, suffixes, stems and unsegmented words. In addition, we constructed two additional lexicons by keeping only the words that have counts two or higher, i.e. excluding singletons. The sizes of these lexicons for the original (unsegmented) and segmented data are about 60K, and 35K respectively. In preliminary experiments with the original lexicon we found that removing singleton words results in a slightly better performance, hence we present results for the original (unsegmented) lexicon only when discarding singletons. We refer to this lexicon as ULEX (Unsegmented-LEXicon). In addition, we test the segmented lexicon with and without discarding singletons. We refer to these as SLEX1 and SLEX2, respectively.

The test data consists of 19 subsets corresponding to different test scenarios in Iraqi Arabic. The test set comprises about 15K words, and about 1.5 hours of speech. Using morphological segmentation reduces the OOV. Indeed, the OOV rate on a development set that we use for this purpose is reduced to 1.0% and 0.7% for SLEX1, and SLEX2, respectively. This is compared to about 3.0% OOV for the original ULEX. On the other hand segmenting the vocabulary will reduce the scope of the N-gram language model. More study of this problem is interesting for future work. In the present paper, we adopted a quick fix. We created a new vocabulary by taking the union of SLEX1 and ULEX. The resulting lexicon is referred to as CLEX. We also interpolated the trigram LMs built using SLEX1 and ULEX using fixed weights that were manually adjusted on a development set. We refer to this LM as the interpolated model. It is worth noting that in this case the vocabularies of the two LMs will be different from CLEX, and when doing the interpolation any word in CLEX that does not exist in any one of the LM vocabularies is considered as the "UNK" word in the corresponding LM.

The results using different lexicons are shown in Table 1. In the table it is clear that using word segmentation improves the WER where about a 4% absolute gain can be observed. This is mainly due to the reduction in OOV. Interestingly, removing singletons also helps a little bit as in the case of the unsegmented vocabulary. This might be attributed to the fact that reducing the vocabulary size could help in obtaining better estimates of the language model probabilities. Finally, using the interpolated language model and the composite vocabulary further improves performance by about 1% absolute, resulting in a final 5.3% absolute gain over ULEX. It is worth mentioning that in the table the 3-gram in the column labelled LM is different for the first three rows. In the first row it is a 3-gram built using the unsegmented vocabulary, while in the second and third rows it is built using the segmented vocabularies with and without discarding singletons.

In parallel, we have also developed a finite state transducer based automatic word segmenter. In this model, Arabic characters, spaces and the automatically inserted prefix and suffix markers appear on the arcs of the finite state machine. The language model is conditioned to insert prefix and suffix markers based upon the frequency of their appearance in relation to the adjacent n-gram character contexts that appear in the training data (we used 5-grams). Our word segmenter was trained on tokenized data provided by LDC and contained about 100K words. The accuracy of

Lexicon	LM	WER
ULEX	3-gram	36.3
SLEX1	3-gram	32.1
SLEX2	3-gram	32.6
CLEX	Interpolated	31.0

Table 1: Word error rate for different lexicons and language models for Iraqi Arabic speech recognition.

this model, computed as the percentage of words in a final segmentation that is in agreement with the words provided in the manually segmented reference, is 95.4%. Then this automatic tokenizer was used instead of a fixed set of affixes, to create the lexicon. The best configuration of this approach led to 34.8% WER on the same Iraqi test set. While it performs better than an unsegmented lexicon, it is not as good as the approach using a fixed list of affixes. This may be due to the automatic segmenter errors that remain too frequent at this point.

5. CONCLUSION

In this paper we present a word decomposition algorithm, that uses popular Arabic affixes, for constructing the lexicon in Iraqi Arabic speech recognition. The algorithm is very simple and works starting from a text corpus and a given set of prefixes and suffixes. In addition to reducing the OOV, and hence reducing the WER, it was shown that further improvements can be obtained by providing better smoothing of the LM by interpolating both the segmented and original models. The net effect is about 13% relative improvement in WER.

While word decomposition definitely helps in reducing the OOV it also shortens the LM scope. Balancing this trade-off is probably an important consideration in the design of word segmentation schemes. Interpolating the segmented and unsegmented models partially addresses this issue. In future work we will study the effect of increasing the LM scope through integrated decoding and/or re-scoring.

Finally, it is worth mentioning that the simple morphological algorithm presented in this paper has also shown a good potential for machine translation (MT). In the machine translation part of our speech to speech translation system, we found that using a similar decomposition scheme helped in improving the BLEU score. This work will be reported in detail elsewhere.

6. REFERENCES

[1] P. Geutner, "Using morphology towards better large vocabulary speech recognition systems," in Proc. ICASSP'95, pp. 445-448, 1995.

[2] A. Berton, P. Fetter, and P. Regel-Brietzmann, "Compound words in large vocabulary German speech recognition system," in Proc. ICSLP'96, 1996.

[3] R. Ordelman, A. Hessen, and F. Jong, "Compound decomposition in Dutch large vocabulary speech recognition," in Proc. EUROSPEECH'03, Geneva, Switzerland, 2003.

[4] M. Larson, D. Willet, J. Kohler, and G. Rigoll, "Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speech," in Proc. ICSLP'00, 2000.

[5] K. Kirchoff et al, "Novel approaches to Arabic speech recognition: report from the 2002 Johns-Hopkins summer workshop," in Proc. ICASSP'03, pp. 344-347, 2003.

[6] D. Vergyri, K. Kirchoff, K. Duh, and A. Stolcke, "Morphology based language modeling for Arabic speech recognition," in Proc. ICSLP'04, Jeju, Korea, 2004.

[7] D. Vergyri, K. Kirchoff, R. Gadde, A. Stolcke, and J. Zheng, "Development of conversational telephone speech recognition for Levantine Arabic," in Proc. EUROSPEECH'05 Lisbon, Portugal, pp. 1613-1616, 2005.

[8] A. Ghaoui, F. Yvon, C. Mokbel, and G. Chollet, "On the use of morphological constraints in N-gram statistical language model," in Proc. EUROSPEECH'05 Lisbon, Portugal, 2005.

[9] B. Xiang, K. Nguyen, L. Nguyen, R. Schwartz, J. Makhoul, "Morphological decomposition for Arabic broadcast news transcription," in Proc. ICASSP'06, Toulouse, France, 2006.

[10] G. Choueiter, D. Povey, S.F. Chen, and G. Zweig, "Morpheme-based language modeling for Arabic LVCSR," Proc. ICASSP'06, Tolouse, France, 2006.

[11] Y. Lee et al., "Language model based Arabic word segmentation," in ACL'03, 2003, pp. 399-406.

[12] F. Jelinek, and R.L. Mercer, "Interpolated estimation of Markov source paramters from sparse data," in Pattern recognition in practice ed. by E.S. Gelsema and L.N. Kanal, pp. 381-402, North-Hollan publishing company, 1980.

[13] M. Afify, L. Nguyen, B. Xiang, S. Abdou, and J. Makhoul, "Recent progress in Arabic broadcast news transcription at BBN," in Proc. EUROSPEECH'05 Lisbon, Portugal, 2005.

[14] Y. Gao, B. Zhou, L. Gu, R. Sarikaya, H.-K. Kuo. A.-V.I. Rosti, M. Afify, and W. Zhu, "IBM MASTOR: Multilingual automatic speech-to-speech translator," in Proc. ICASSP'06, Toulouse, France, 2006.

[15] R. A. Gopinath, "Maxmimum Likelihood Modeling with Gaussian Distributions for Classification," in Proc. ICASSP '98, Seattle, USA.

[16] D. Povey, and P.C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in Proceedings ICASSP'02, Orlando, FL, May 2002.

[17] L.R. Bahl, P.V. de Souza, P.S. Gopalakrishnan, D. Nahamoo, and M.A. Picheny, "Robust methods for using context-dependent features and models in a continuous speech recognizer," in Proc. ICASSP'94, Adelaide, Australia, 1994.

[18] P.S. Gopalakrishnan, L.R. Bahl, R.L. Mercer, "A tree search strategy for large-vocabulary continuous speech recognition," in Proc. ICASSP'95, Detroit, Michigan 1995.