Morpheme Based Factored Language Models for German LVCSR

Amr El-Desoky Mousa, M. Ali Basha Shaik, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition – Computer Science Department
RWTH Aachen University, 52056 Aachen, Germany
{desoky,shaik,schlueter,ney}@cs.rwth-aachen.de

Abstract

German is a highly inflectional language, where a large number of words can be generated from the same root. It makes a liberal use of compounding leading to high Out-of-vocabulary (OOV) rates, and poor Language Model (LM) probability estimates. Therefore, the use of morphemes for language modeling is considered a better choice for Large Vocabulary Continuous Speech Recognition (LVCSR) than the full-words. Thereby, better lexical coverage and less LM perplexities are achieved. On the other side, the use of Factored Language Models (FLMs) is considered a successful approach that allows the integration of many information sources to get better LM probability estimates. In this paper, we try a combined methodology for language modeling where both morphological decomposition and factored language modeling are used in one model called *morpheme based FLM*. Finally, we obtain around 2.5% relative reduction in Word Error Rate (WER) with respect to a traditional full-words system.

Index Terms: morpheme, factored language model, German

1. Introduction

German is characterized by a complex morphological structure, as a large number of distinct lexical forms can be generated from the same root due to word compounding, inflection, and derivation. This huge lexical variety leads to data sparsity resulting in poor language model probability estimates, and thus high perplexities. This normally causes problems in LVCSR. One successful approach to deal with these problems is to build LMs on morphemic sub-words (morphemes) rather than full-words. Another approach to improve the LM probability estimates is to use the factored language models which are powerful models that combine multiple sources of information and efficiently integrate them via a complex backoff mechanism [1].

Normally, morphemes are generated from the full-words by applying word decomposition based on supervised or unsupervised approaches. Both approaches are successfully used for German as well as for other languages. The supervised approaches make use of linguistic knowledge like in [2], where a set of manual rules is developed for German word decomposition. However, in [3], a manually decomposed lexicon is used for recognition. Other supervised methods rely on carefully built morphological analyzers based on lexical and syntactic knowledge like in [4, 5, 6]. Although the supervised decomposition is normally optimized for high performance, it requires labor-intensive work and still suffer from the so-called *unknown word problem*, that is, words that are not coded into the system. On the other hand, the unsupervised approaches are statistical based data driven approaches like in [7, 8, 9]. In [10], an algorithm is proposed that decomposes words according to the statistical relevance of the resulting constituents. Other unsupervised methods are based on the Minimum Description Length principle (MDL) like in [11, 12]. On the contrary, the unsupervised approaches do not require any language specific knowledge and can be applied to any language.

A relatively unexplored form of LMs is the factored language model (FLM). Although the FLM is first introduced in [1, 13] for incorporating various morphological information in Arabic LMs, its applicability is in fact more general. In a FLM, a word is viewed as a collection or 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. Hence, the 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: $p(f_t^{1:K}|f_{t-1}^{1:K}, f_{t-2}^{1:K}, ..., f_{t-n+1}^{1:K})$, which could be reformed as a product of probabilities of the form $p(f|f_1, f_2, ..., f_N)$ [14]. This model represents the interdependencies among features of words both across time and within word. The main idea of the model is to backoff to other factors when some word n-gram is not sufficiently observed in the training data, thus improving the probability estimates. For detailed description of the FLMs refer to [14].

Although many previous publications investigate both morpheme based LMs and FLMs as reviewed above, no attempt is made to combine both approaches in a single combined methodology. An exception to this is our previous work in [15], where significant improvements in WERs are achieved by using morphologically decomposed FLMs for Arabic LVCSR. While, in [16], a set of decompositional factors are used for building FLMs for Amharic language and compared with standard morpheme based LMs. In this paper, we combine the strengths of morpheme based approach and factored language modeling approach. Therefore, *morpheme based FLMs* estimated over factored morphemes are used for German LVCSR. We compare our approach to the standard full-word, standard morpheme based, and factored full-word n-gram approaches. As per our knowledge, the application of this methodology to German LVCSR is not previously explored.

2. Methodology

2.1. Morphological decomposition

We perform morphological decomposition of German words using a data driven tool called *Morfessor* [17]. It is a statistical tool that can automatically discover the optimal decomposition for words of a text corpus based on the MDL principle. It is mainly designed to cope with languages having rich morphology, where the number of morphemes per word is varying so

Copyright © 2011 ISCA

much and not known in advance [12]. In our previous publication [18], Morfessor is successfully used to model some fraction of in-vocabulary words leading to significant improvement in WER for a German LVCSR task compared to a traditional full-words system. Therein, it is found after a series of optimization experiments over the development corpus that keeping 5k most frequent full-words without decomposition (out of 100k vocabulary) is quite helpful for the recognition process.

We train our decomposition model using a list of unique words that occur more than 5 times in the LM training data; this gives about 0.5 Million words. We do not include other words in order to avoid irregular words that are harmful to the training process. Nevertheless, the model is still capable of decomposing unseen words. In addition, the resulting decompositions are modified by merging irregular very short fragments so as to produce a clean set of morphemes. The final set of morphemes appears linguistically meaningful, where mainly the compound words are decomposed and meaningful morphemes are stripped of the full-words.

2.2. Preparing factors for FLM

The first issue regarding the construction of the FLM is to choose an appropriate set of features (factors). This can be done using linguistic knowledge or based on data driven techniques. Here, both approaches are used to define our set of factors.

2.2.1. Deriving linguistic based factors

To derive our linguistic based factors, we use the *TreeTagger* developed at the Institute of Computational Linguistics at the University of Stuttgart. It is a probabilistic tool that uses decision trees for annotating text with part-of-speech and lemma information, where lemma is the canonical baseform of the word [19]. The TreeTagger has been successfully used to tag words of many languages including German. It is also adaptable to other languages if a lexicon and a manually tagged training corpus are available. Moreover, it is found that the tags given by the TreeTagger are also valid when linguistically meaningful morphemes are provided as input instead of normal full-words. Using the information generated by the TreeTagger, we could define two different factors for German words and morphemes so as to be used as a part of the FLM conditioning factors, namely: the word baseform, and the part-of-speech tag.

2.2.2. Deriving data driven factors

Additionally, we derive one data driven factor called *data driven class* defined for words or morphemes. For generality of the notation, we use the term *word* to refer to word or morpheme. To generate this factor, we first map discrete words of the vocabulary into a continuous parameter space in the form of vectors of real numbers [20]. Then, this continuous space of vectors is clustered into a fixed number of clusters via standard *K-means* clustering, and every word is assigned a cluster index which acts as the data driven class of the underlying word.

In order to map discrete words into a continuous space, we follow an approach inspired from Latent Semantic Analysis (LSA) [21]. We begin with the creation of word-pair co-occurrence matrix based on bigram counts. Where, all the word bigrams are accumulated for the entire text corpus to fill in the entries of a co-occurrence matrix C, where $C(w_i, w_j)$ denotes the counts for which word w_i follows w_j in the text corpus. This forms a large, but very sparse matrix, since typically a small number of words follow a given word. The matrix dimension is $M \times M$, where M is the number of the vocabulary entries. Because of its large size and sparsity, Singular Value Decomposition (SVD) is a good choice to produce a reduced-rank approximation of this matrix. The co-occurrence matrix typically contains few high frequency events and many low frequency events. Since SVD derives a compact approximation of the co-occurrence matrix that is optimum in the least-square sense, it is normally over-fitted to the high frequency events which may not be the most informative. Therefore, the entries of the word-pair co-occurrence matrix are log-smoothed according to Equation 1. Then, following the same approach described in [22, 20], SVD is performed as in Equation 2.

$$\hat{C}(w_i, w_j) = log(C(w_i, w_j) + 1) \tag{1}$$

$$\hat{C} \approx USV^T \tag{2}$$

Assuming that we use an order of decomposition $R \ll M$, then U is a left singular matrix with dimension $M \times R$. S is a diagonal matrix of singular values with dimension $R \times R$. V is a right singular matrix with dimension $M \times R$. The continuous space for the words is defined as the space spanned by the column vectors of $A_{M\times R} = US$. Now, assuming that a word w_i is represented by an indication vector $\vec{w_i}$ of dimension $M \times 1$, where the i^{th} entry of $\vec{w_i}$ is 1 and all the remaining entries are zeros. Then, this indication vector $\vec{w_i}$ is mapped to a lower dimensional vector $\hat{w}$ of dimension $R \times 1$, using the formula:

$$\hat{w_i} = A^T \vec{w_i} \tag{3}$$

In other words, the above Equation 3 represents a word w_i by the i^{th} row vector of matrix A. These row vectors are called *latent word vectors* which define the continuous space of the original discrete words. In order avoid zero variance in word mapping into continuous space, all latent word vectors are added a small amount of white noise. Using a vocabulary of size $M = 100k$, and considering an order of decomposition $R = 100$, we generate $100k$ vectors each of R real values. Those $100k$ vectors are clustered into 250 classes (the number is empirically chosen), then each word vector is assigned a class index from 0 to 249. Thereby, we attach a data driven class index to each word in vocabulary.

2.2.3. Defining the set of factors

Having the above factors generated, we define the following set of factors to be used as the FLM conditioning factors for both word and morpheme based FLMs:

- **W** : word/morpheme surface form.
- **B** : word/morpheme baseform.
- **P** : part-of-speech tag of word/morpheme.
- **I** : data driven class index of word/morpheme.

Both the word and morpheme based versions of the LM training data are processed so as to produce the factored representation as required by SRILM-FLM extensions [14].

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) presented in [23] which seeks to minimize the perplexity of the FLM over some held-out text. Furthermore, we apply some manual optimization to fine tune the FLM parameters [1]. For memory limitations, we only use factors up to two previous time slots (trigram like models). Finally, we come up with a set of competing FLMs. In Table 1, we record the perplexities measured for a held-out text. The first column gives the combination of the parent factors. So that, our baseline FLM_1 corresponds to the model: $P(W_t|W_{t-1}, W_{t-2})$, which is the FLM equivalent of the standard trigram LM. While, $FLM_{2:4}$ correspond to the model:

$$P(W_t|W_{t-1}, B_{t-1}, I_{t-1}, P_{t-1}, W_{t-2}, B_{t-2}, I_{t-2}, P_{t-2})$$

However, $FLM_{5,6}$ correspond to the model:

$$P(W_t|W_{t-1}, B_{t-1}, P_{t-1}, W_{t-2}, B_{t-2}, P_{t-2})$$

The FLM_7 corresponds to the model:

$$P(W_t|W_{t-1}, I_{t-1}, P_{t-1}, W_{t-2}, I_{t-2}, P_{t-2})$$

The differences among the models with the same parent factors come into the structure of the backoff path and the choice of the smoothing options.

From Table 1, comparing perplexities of our proposed FLMs to the baseline perplexity, we see that using more factors along with the normal word helps decreasing the perplexity. This is true for both word and morpheme based FLMs. Moreover, the morpheme based FLMs achieve lower perplexities than the word based FLMs. Nevertheless, we know from our previous experience in Arabic [15], that the FLM which achieves the best WER may not correspond to the one with the least perplexity. It is also worth noting that no normalization takes place during the computation of the FLM perplexities.

Table 1: *Perplexities of different FLM topologies based on full-words or morphemes (WB: word based, MB: morpheme based).*

FLM_x: **W | parent factors**	**WB**	**MB**
1: W | W1 W2 (baseline)	349.7	311.0
2: W | W1,B1,I1,P1,W2,B2,I2,P2	311.7	280.6
3:	314.8	283.8
4:	330.4	296.4
5: W | W1,B1,P1,W2,B2,P2	342.9	306.0
6:	384.7	343.0
7: W | W1,I1,P1,W2,I2,P2	326.2	294.7

4. Experimental Setup

Our acoustic models are triphone models trained using about 343h of audio material taken from Broadcast News (BN), European Parliament Plenary Sessions (EPPS), read articles, dialogs, and some web data. The acoustic models are trained based on Maximum Likelihood (ML) method. While, our LM training corpus consists of around 188 Million running full-words including the official data provided for the Quaero project (mainly news data). The text corpus is used for vocabulary selection (M most frequent words) and to estimate back-off N-gram LMs as well as FLMs by the SRILM toolkit [24].

Our speech recognizer works in 2 passes. In the first pass, across-word acoustic models are used with no speaker adaptation. A standard 3-gram back-off LM is used to construct the search space and to produce recognition lattices, then lattices are rescored with a 4-gram LM. The second pass performs speaker adaptation based on both Constrained Maximum Likelihood Linear Regression (CMLLR), and Maximum Likelihood Linear Regression (MLLR). Here, a standard 3-gram LM is used to generate N-best lists, then N-best list rescoring is performed using the different FLM topologies shown in Table 1.

To evaluate the recognition performance, we use the Quaero 2009 development and evaluation corpora (dev09: 7.5h; eval09: 3.8h). Each corpus consists of audio material from EPPS sessions and web sources. Additionally, eval09 has some BN data.

5. Experiments

5.1. Word based system

Table 2 summarizes the recognition results of a 100k word based system running on both dev09 and eval09 corpora. Here, a standard word based 3-gram LM is used in the first pass to get lattices. Then, lattices are rescored with a standard word based 4-gram LM. In the second pass, a standard word based 3-gram LM is used to generate N-best sentences *(N = 5 to 30)*. The sentences are processed in a similar way as the training data (refer to Section 2) so as to produce a factored word representation suitable for word based FLM rescoring. Then, the N-best lists are rescored with $FLM_{2:7}$ introduced in Section 3. The WERs after the second pass rescoring are presented in Table 2. We see that the best WER is obtained by using FLM_5 for N-best list rescoring. We obtain WER reductions of [dev09: 0.3% relative (0.1% absolute); eval09: 1.1% relative (0.3% absolute)] over the standard word based 3-gram LM.

Table 2: *Second pass recognition results for a 100k word based system (OOV rate = [dev09: 4.6%, eval09: 4.5%]).*

	WER [%]	
2^{nd} **pass**	**Dev09**	**Eval09**
3-gram (baseline)	33.0	28.5
N-best FLM rescoring:		
+ FLM_2	33.2	28.3
+ FLM_3	33.1	28.4
+ FLM_4	33.1	28.4
+ FLM_5	**32.9**	**28.2**
+ FLM_6	33.0	28.3
+ FLM_7	33.1	28.4

5.2. Morpheme based system

Table 3 summarizes the recognition results of a 100k morpheme based system running on both dev09 and eval09 corpora. Herein, a 5k most frequent full-words are kept without decomposition. This is previously found to be helpful in order to prevent the most frequent words from being mixed-up with other morphemes in the search space [18]. In a similar way as in the previous section, a standard morpheme based 3-gram LM is used in the first pass to get lattices. Then, lattices are rescored with a standard morpheme based 4-gram LM. In the second pass, a standard morpheme based 3-gram LM is used to generate N-best sentences ($N = 5to30$). The sentences are processed similarly as the case of the training data (see Section 2) so as to produce a factored morpheme representation suitable for morpheme based FLM rescoring. Then, the N-best lists are rescored with $FLM_{2:7}$ introduced in Section 3. The WERs after the second pass rescoring are shown in Table 3. We see that the best WER is obtained Also by using the same FLM_5 for N-best list rescoring. We obtain WER reductions of [dev09: 0.9% relative (0.3% absolute); eval09: 0.4% relative (0.1% absolute)] over the standard morpheme based 3-gram LM. On the other hand, we obtain WER reductions of [dev09: 2.4% relative

[1] for exact topologies of all models contact the first author.

(0.8% absolute); eval09: 2.1% relative (0.6% absolute)] over the standard word based 3-gram LM of Table 2.

Table 3: *Second pass recognition results for a 100k morpheme based system having 5k full-words + 95k morphemes (OOV rate = [dev09: 4.1%, eval09: 3.9%]).*

	WER [%]	
2^{nd} **pass**	**Dev09**	**Eval09**
3-gram (baseline)	32.5	28.0
N-best FLM rescoring:		
+ FLM_2	32.6	28.0
+ FLM_3	32.5	27.9
+ FLM_4	32.5	28.0
+ FLM_5	**32.2**	**27.9**
+ FLM_6	32.3	27.9
+ FLM_7	32.5	27.9

6. Conclusions

We introduced a morpheme based factored language modeling approach for German LVCSR. Our approach combines the strengths of both morpheme based and factored language modeling. Thus, we used language models with factored morphemes. We compared our approach to the traditional approaches like: standard word based n-grams, standard morpheme based n-grams, and word based factored language models. Moreover, we tested several FLM structures during N-best list rescoring. Finally, we could achieve WER improvements over all the traditional approaches. We believe that there is a further possibility for more improvement, provided that better morphological features are available.

7. Acknowledgements

This work was partly funded by the European Community's 7^{th} Framework Programme under the project SCALE (FP7-213850), and partly realized under the Quaero Programme, funded by OSEO, French State agency for innovation.

8. References

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

[2] M. Adda-Decker and G. Adda, "Morphological decomposition for ASR in German," in *Workshop on Phonetics and Phonology in ASR*, Saarbrücken, Germany, Mar. 2000, pp. 129 – 143.

[3] A. Berton, P. Fetter, and P. Regal-Brietzmann, "Compound words in large-vocabulary German speech recognition systems," in *Proc. Int. Conf. on Spoken Language Processing*, vol. 2, Philadelphia, PA, USA, Oct. 1996, pp. 1165 – 1168.

[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] W. Byrne, J. Hajič, P. Ircing, P. Krbec, and J. Psutka, "Morpheme based language models for speech recognition of Czech," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, 2000, vol. 1902, pp. 139 –162.

[6] J. Kneissler and D. Klakow, "Speech recognition for huge vocabularies by using optimized sub-word units," in *Proc. European Conf. on Speech Communication and Technology*, vol. 1, Aalborg, Denmark, Sep. 2001, pp. 69 – 72.

[7] M. Adda-Decker, "A corpus-based decompounding algorithm for German lexical modeling in LVCSR," in *Proc. European Conf. on Speech Communication and Technology*, Geneva, Switzerland, Sep. 2003, pp. 257 – 260.

[8] R. Ordelman, A. V. Hassen, and F. D. Jong, "Compound decomposition in Dutch large vocabulary speech recognition," in *Proc. European Conf. on Speech Communication and Technology*, Geneva, Switzerland, Sep. 2003, pp. 225 – 228.

[9] T. Rotovnik, M. S. Maučec, and Z. Kačič, "Large vocabulary continuous speech recognition of an inflected language using stems and endings," *Speech Communication*, vol. 49, no. 6, pp. 537 – 452, Jun. 2007.

[10] M. Larson, D. Willett, J. Köhler, and R. Rigoll, "Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches," in *Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, Oct. 2000.

[11] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraclar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Transactions on Speech and Language Processing*, vol. 5, no. 1, Dec. 2007.

[12] M. Creutz, "Induction of the morphology of natural language: Unsupervised morpheme segmentation with application to automatic speech recognition," Ph.D. dissertation, Helsinki University of Technology, Finland, 2006.

[13] K. Kirchhoff, J. Bilmes, J. Henderson, R. Schwartz, M. Noamany, P. Schone, G. Ji, S. Das, M. Egan, F. He, D. Vergyri, D. Liu, and N. Duta, "Novel speech recognition models for Arabic," Johns-Hopkins University Summer Research Workshop, Baltimore, Maryland, USA, Tech. Rep., Jul. 2002.

[14] 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.

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

[16] M. Tachbelie, S. Abate, and W. Menzel, "Morpheme-based and factored language modeling for Amharic speech recognition," in *Human Language Technology. Challenges for Computer Science and Linguistics*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2011, vol. 6562, pp. 82 – 93.

[17] M. Creutz and K. Lagus, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0," Computer and Information Science Helsinki University of Technology, Finland, Tech. Rep., Mar. 2005.

[18] A. El-Desoky, M. Shaik, R. Schlüter, and H. Ney, "Sub-lexical language models for German LVCSR," in *IEEE Workshop on Spoken Language Technology*, Berkeley, CA, USA, Dec. 2010, pp. 159 – 164.

[19] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proc. of the International Conference on New Methods in Language Processing*, Manchester, UK, Sep. 1994, pp. 44 – 49.

[20] R. Sarikaya, M. Afify, and B. Kingsbury, "Tied-mixture language modeling in continuous space," in *Proc. Human Language Technology Conf. of the North American Chapter of the ACL*, Boulder, CO, USA, Jun. 2009, pp. 459 – 467.

[21] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[22] J. Bellegarda, "Large vocabulary speech recognition with multispan language models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 76 – 84, 2000.

[23] 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.

[24] 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.