Morpheme Level Feature-based 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

One of the challenges for Large Vocabulary Continuous Speech Recognition (LVCSR) of German is its complex morphology and high level of compounding. It leads to high Out-of-vocabulary (OOV) rates, and poor Language Model (LM) probabilities. In such cases, building LMs on morpheme level can be considered a better choice. Thereby, higher lexical coverage and lower LM perplexities are achieved. On the other side, a successful approach to improve the LM probability estimation is to incorporate features of words using feature-based LMs. In this paper, we use features derived for morphemes as well as 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). Relative reductions of around 1.5% in Word Error Rate (WER) are achieved compared to the best previous results obtained using FLMs.

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

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 causes data sparsity problems and leads to high OOV rates and poor LM probability estimates indicated by high perplexities. A traditional approach to overcome these problems is to use a very large vocabulary and more training data. Yet, still relatively high OOV rates are obtained. Moreover, the speech recognition system requires more resources (CPU/memory).

An alternative approach is to use morpheme-based LMs in order to lower the OOV rate and perplexity, reduce data sparsity, and thus achieve lower WERs. 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 [1, 2]. Some supervised methods rely on carefully built morphological analyzers based on lexical and syntactic knowledge like in [3, 4]. On the other hand, the unsupervised approaches are statistical data driven approaches like in [5, 6, 7]. Some unsupervised methods are based on the Minimum Description Length principle (MDL) [8]. In contrast to the supervised approaches, the unsupervised approaches do not require any language specific knowledge and can be applied to any language.

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 suitable features (also called classes or factors) to words and build LMs over those features. These yield better smoothing and, hopefully, better generalization to unseen word sequences. The features can also be generated based on linguistic methods as in [9], or via data driven approaches as in [10]. The approaches for incorporating word features into LMs are called *feature-based LMs*: like stream-based LMs [11], class-based LMs [12] and FLMs [13]. In a stream-based LM, 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. However, a class-based LM combines the N-gram model over classes with the probability distribution of words in classes so as to estimate better 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 probabilities. All these types of LMs can be used for rescoring N-best lists.

This paper presents an approach that attempts to gain the benefits of feature-based LMs, while at the same time retain the advantages of morpheme-based LMs. This is accomplished by generating features on the level of morphemes. In previous work [14], we investigated the use of morpheme level FLMs for German LVCSR. Here, we compare the performance of FLMs to stream-based and class-based LMs. Moreover, we examine the interpolation of N-gram LMs with class-based LMs, and the combination of N-best scores obtained from different LMs.

2. Methodology

2.1. Word decomposition and feature derivation

We perform morphological decomposition of words using a data driven tool called *Morfessor* [15]. It is a sta-

tistical 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 strongly [8]. In previous work [14, 16], Morfessor morphemes were successfully used to model some fraction of vocabulary words leading to significant improvement in WER for German LVCSR Therein, it is found that keeping 5k most frequent full-words without decomposition (out of 100k items) is quite helpful for the recognition process.

It is stated in [15] that ignoring word counts in a given corpus and using only the corpus vocabulary to train the Morfessor model produces segmentations that are closer to linguistic morphemes. Therefore, we train our Morfessor model using a vocabulary of distinct words that occur more than 5 times in the training corpus. This gives about 0.5 Million words. We do not include less frequent words in order to avoid irregularities that are harmful to the training process. In addition, the resulting segmentations are postprocessed to avoid very short and noisy morphemes. The final set of morphemes appears linguistically meaningful, where mainly the compound words are decomposed (giving valid smaller words) and meaningful morphemes are stripped out. An example of observed decompositions is: *eingeschlafen* $\rightarrow$ *ein+ geschlafen*.

Word features are generated using the *TreeTagger* developed by the University of Stuttgart [17]. It is a probabilistic tool that uses decision trees for annotating text with Part-of-speech *(POS)* and lemma information, where lemma is the canonical *baseform* of the word. The TreeTagger has been successfully used to tag words of many languages including German. One of the useful properties of the TreeTagger is that it operates successfully over morphemes as well as full-words provided that the input morphemes are linguistically meaningful which is true in our case. In addition to POS and baseform, we derive a third feature called *index*. This is a data driven class index assigned to every word or morpheme after performing a classification algorithm. First, all the discrete vocabulary items are converted into real valued vectors using word-pair co-occurrence matrix and Singular Value Decomposition (SVD) [18, 19], then these vectors are clustered into 250 clusters using k-means approach. A detailed description of the algorithm is found in [14].

Finally, The LM training corpus is preprocessed so that every word/morpheme is replaced by a vector of features: {*W-<word>:P-<pos>:B-<baseform>:I-<index>*}. A sequence of individual vector components defines a feature stream (class stream). A vector example in the case of words is: *eingeschlafen* $\rightarrow$ {*W-eingeschlafen:P-VVPP:B-einschlafen:I-224*}; where VVPP means past participle verb. However, in the case of morphemes: *eingeschlafen* $\rightarrow$ {*W-ein+:P-ART:B-ein:I-15*} {*W-geschlafen:P-VVPP:B-schlafen:I-192*}.

2.2. Stream-based language models (SLMs)

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

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

If this model is built over 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 baseforms, POSs or Indexes. Similar to Equation 1, given a sequence of classes $c_1 c_2, ..., c_M$, an N-gram stream-based model is:

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

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

2.3. Class-based language models (CLMs)

The class-based LMs are initially described in [12]. 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.

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

Normally, the standard word-based N-gram LMs perform 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 [20]. Here, we use linear interpolation with multiple class-based LMs expressed as:

$$p(W) = \lambda_0 p_w(W) + \sum_{i=1}^{k} \lambda_i p_c^i(W) \quad (4)$$

where W is the word sequence, $p_w(W)$ is the word-based probability, $p_c^i(W)$ is the class-based probability using the i^{th} model, λ_i are the interpolation weights optimized on some development data, such that $\sum_{i=0}^{k} \lambda_i = 1$, and k is the number of class-based models.

2.4. Factored language models (FLMs)

FLMs were first introduced in [13]. In an FLM, a word is viewed as a vector of K parallel factors (features), 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 model takes the form $p(f_t^{1:K}|f_{t-1}^{1:K}, f_{t-2}^{1:K}, ..., f_{t-n+1}^{1:K})$ [21]. This model represents the interdependencies among features of words both across position and within word. It uses a complex backoff mechanism across multiple features. The model backs off to other factor combinations 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 $P(W_t|W_{t-1}, B_{t-1}, P_{t-1}, W_{t-2}, B_{t-2}, P_{t-2})$, where W is word, B is baseform, and P is POS. It is worth noting that the Index factor I was found not helpful for the FLM. The details of how the model is created and optimized are found in our previous work in [14].

2.5. N-best 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 on some development data [11]. This is similar to the discriminative model combination described in [22]. For the weight optimization, we use *"Amoeba"* search implemented in SRILM toolkit [23].

3. Experimental setup

Our acoustic models are triphone models that are maximum likelihood trained using about 343h of audio material taken from Broadcast News (BN), European Parliament Plenary Sessions (EPPS), read articles, dialogs, and web data. The LM training corpus consists of around 188 Million running full-words including the official data of the Quaero project (mainly news data). The text corpus is used for vocabulary selection (M most frequent words) and to estimate LMs via the SRILM toolkit [23]. Our speech recognizer works in 2 passes. In the first pass, across-word acoustic models are used without speaker adaptation. A standard 3-gram back-off LM is used to construct the search space and to produce 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). A standard 3-gram LM is used to generate N-best lists, then N-best rescoring is performed using different types of LMs. 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.

4. Experiments

In Table 1, the column labeled *"fw"* shows the WERs of a 100k full-words system. While, the column labeled *"mrf"* shows the WERs of a 100k morpheme-based system that uses *5k full-words + 95k morphemes*. The first row of the table presents the WERs using the 3-gram LM before any rescoring. The second column presents the WERs after lattice rescoring using a normal 4-gram LM. The rest of the table gives the WERs after N-best rescoring using different types of feature-based LMs. N-best sentences with *N = 10 to 200* are generated and processed as illustrated in Section 2.1 so as to produce a representation suitable for the rescoring LM. We can see that the class-based LMs perform almost similar to the optimized FLM. On the other hand, they perform better than stream-based LMs. In addition, the interpolation of the class-based LMs with the normal N-gram model helps reducing the WERs a bit further. The interpolation weights are optimized on dev09 corpus. Moreover, the best interpolated model is created by interpolating all the 3 class-based LMs together with the N-gram model. This model achieves the best performance for both full-words and morphemes. In the case of full-words, relative WER reductions of [dev09: 0.9%; eval09: 1.8%] are achieved compared to the traditional 4-gram lattice rescoring. While, in the case of morphemes, relative WER reductions of [dev09: 1.5%; eval09: 1.8%] are achieved. At the end, we make an attempt to perform score combination of the FLM and the 3 interpolations of the class-based LMs with the 4-gram LM. Combination weights are optimized on dev09 corpus. Unfortunately, this could not further reduce the WERs.

5. Conclusions

We investigated the use of morpheme level feature-based LMs. The performance of stream-based, class-based and factored LMs was compared for a German LVCSR task. Different types of morphological and data-driven features are examined. We showed that the feature-based modeling 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 effect of data sparsity. While the feature-based models try to achieve better generalization to unseen word sequences. The best performance is obtained by interpolating all the class-based models with the traditional word/morpheme N-gram models. Relative WER reductions of around 3.0% are achieved compared to the conventional word-based approach. Moreover, relative WER reductions of around 1.5% are achieved compared to the use of FLMs. Using the bootstrap method of significance analysis described in [24], the WER reductions are proved statistically significant with a probability of improvement around 95%.

Table 1: *WERs[%] after 2^{nd} pass rescoring for [**fw**: 100k full-word system, OOV rate = (dev09: 4.6%, eval09: 4.5%); **mrf**: 100k morpheme based system (5k full-words + 95k morphemes), OOV rate = (dev09: 4.1%, eval09: 3.9%)]. **ftr**: features used in LM; **SLM**: stream-based LM; **CLM**: class-based LM; **FLM**: factored LM; **B**: baseform; **P**: POS; **I**: index; **SC**: score combination of FLM and 3 interpolations of CLMs with 4-gram LM.*

2^{nd} **pass**		**fw**		**mrf**	
LM	**ftr**	**dev09**	**eval09**	**dev09**	**eval09**
3-gram	-	33.0	28.5	32.5	28.0
4-gram	-	**32.8**	**28.4**	**32.3**	**28.0**
FLM	*B,P*	32.9	28.2	32.2	27.9
SLM	*B*	32.9	28.3	32.4	27.9
	P	32.9	28.3	32.4	27.9
	I	33.0	28.3	32.4	28.0
CLM	*B*	32.8	28.2	32.2	27.8
	P	32.8	28.2	32.2	27.9
	I	32.9	28.3	32.3	28.0
4-gram + *CLM*	*B*	32.6	28.1	32.2	27.7
	P	32.5	28.0	31.9	27.5
	I	32.8	28.2	31.9	27.5
	B,P,I	**32.5**	**27.9**	**31.8**	**27.5**
SC	-	32.5	28.0	31.9	27.5

6. Acknowledgments

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.

7. References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] A. El-Desoky, M. Shaik, R. Schlüter, and H. Ney, "Morpheme Based Factored Language Models for German LVCSR," in *Interspeech*, Florence, Italy, Aug. 2011, pp. 1445 – 1448.

[15] 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.

[16] 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.

[17] H. Schmid, "Improvements In part-of-speech tagging with an application to German," in *Proc. of the the ACL SIGDAT-Workshop*, Dublin, Ireland, Mar. 1995, pp. 47 – 50.

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

[19] 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.

[20] C. Samuelsson and W. Reichl, "A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Phoenix, AZ, USA, Mar. 1999, pp. 537 – 540.

[21] 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.

[22] P. Beyerlein, "Discriminative model combination," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Seattle, WA, USA, May 1998, pp. 481 – 484.

[23] 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.

[24] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Montreal, Canada, May 2004, pp. 409 – 412.