Morpheme Level Hierarchical Pitman-Yor Class-based Language Models for LVCSR of Morphologically Rich Languages

Amr El-Desoky Mousa[1], *M. Ali Basha Shaik*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition,
Computer Science Department, RWTH Aachen University, Aachen, Germany
[2]Spoken Language Processing Group, LIMSI CNRS, Paris, France
{desoky,shaik,schlueter,ney}@cs.rwth-aachen.de

Abstract

Performing large vocabulary continuous speech recognition (LVCSR) for morphologically rich languages is considered a challenging task. The morphological richness of such languages leads to high out-of-vocabulary (OOV) rates and poor language model (LM) probabilities. In this case, the use of morphemes has been shown to increase the lexical coverage and lower the LM perplexity. Another approach used to improve the LM probability estimates is to incorporate additional knowledge sources in the LM estimation process using class-based LMs (CLMs). Recently, the hierarchical Pitman-Yor LMs (HPYLMs) have shown superiority over the modified Kneser-Ney (MKN) smoothed N-gram LMs in terms of both perplexity (PPL) and word error rate (WER) on word-based LVCSR tasks. In this paper, hierarchical Pitman-Yor class-based LMs (HPYCLMs) are combined with morpheme level language modeling. This enables the application of the proposed models on top of morpheme-based systems. Experiments are conducted on Arabic and German LVCSR tasks. Consistent performance improvements are obtained for all the available corpora compared to the conventional morpheme-based and class-based LMs.

Index Terms: language model, morpheme-based, class-based, hierarchical Pitman-Yor, rich morphology

1. Introduction

Arabic and German are characterized by a complex morphological structure. Arabic belongs to the family of Semitic languages. It is in fact a highly inflected language having a large number of different surface forms. The Arabic words are derived from roots, by applying patterns to get stems and then attaching different affixes to obtain a large number of word forms. Thus, a stem can be thought of as being further decomposed into a root and a pattern [1]. On the other hand, German belongs to the family of Germanic languages. It is also cited as an outstanding example of highly inflected languages, as a large number of words can be derived from the same root. In addition, German makes a liberal use of noun compounding. Also, the meaning of German words can be expanded through the use of prefixes [2]. This huge lexical variety of Arabic and German causes data sparsity problems and leads to high OOV rates and poor LM probability estimates indicated by high LM perplexities. Normally, a conventional Arabic LVCSR system uses a very large LM training corpora and recognition vocabulary. Yet, still relatively high WERs are observed.

An alternative approach to deal with morphological richness is to use sub-lexical LMs [3, 4]. Typically, morpheme-based LMs are used to reduce data sparsity, lower the OOV rate and perplexity, and thereby achieve lower WERs. Morphemes are the smallest linguistic components of the word that hold semantic meanings. They are generated by applying morphological decomposition to words based on supervised or unsupervised approaches. The supervised approaches make use of linguistic knowledge like in [5]. Other supervised methods rely on carefully built morphological analyzers like in [6, 7, 8]. On the other hand, the unsupervised approaches are statistical data-driven approaches like in [9, 10]. Other unsupervised methods are based on the minimum description length (MDL) principle like in [11]. On the contrary, the unsupervised approaches do not require any language specific knowledge.

Another approach to overcome the data sparseness and reduce the dependence of the 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 [12], or via data-driven approaches [13]. One approach for incorporating word features into LMs is the class-based LM (CLM)[14]. It 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. This type of LM can be used to perform N-best list rescoring.

Recently, there has been a considerable amount of research aimed at improving the fundamental modeling of the N-gram LMs. Among this, hierarchical Bayesian LMs [15] have succeeded to achieve a comparable performance to the state-of-the art N-gram LMs smoothed with MKN smoothing. A hierarchical Pitman-Yor LM (HPYLM), initially introduced in [16], is a type of Bayesian LM based on the Pitman-Yor (PY) process that has been shown to improve the perplexity over the MKN smoothed N-gram LM. In [17], the HPYLM has been implemented as an extension to the SRILM toolkit [18] and WER improvements have been reported for typical LVCSR tasks.

This paper presents a novel approach that attempts to combine the benefits of all the aforementioned techniques. We make use of the HPYLM methodology to build CLMs using classes assigned on morpheme level. This is called *morpheme level HPYCLM*. Thereby, we gain the advantages of using morpheme-based LMs, along with the benefits of feature-rich modeling, in addition to the improvement from the HPYLM. Moreover, linear interpolation is performed to combine different types of LMs. The results are compared to our best previously published results in [19, 20]. Although little improvements are achieved over the best previous results, they are systematically consistent over all the available corpora.

2. Word decomposition and class derivation

2.1. Arabic

The Arabic LM training data is processed using MADA 2.0 tool [21]. MADA is a morphological analyzer and disambiguator tool developed for Arabic language. It is built on top of the Buckwalter Arabic morphological analyzer (BAMA) [22]. It is able to associate a complete set of morphological tags with each word in context. These tags are used to produce robust word diacritization and tokenization. Based on this tokenization, we produce decomposed words in the form of *"prefix+ stem +suffix"*, where the existence of the prefix and the suffix is optional. The '+' sign is used as a marker for full-word recombination. For a detailed description of the decomposition process and constraints, see our previous work [7]. Moreover, in [7], it was found out that having around 20k most frequent decomposable full-words without decomposition (out of 256k items) in the recognition vocabulary is quite helpful to achieve high recognition performance.

Starting from the MADA morphological tags along with the generated 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 category 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 the *"Sebawai"* tool [23]. Similarly, these features can be defined for morphemes as well as for full-words. Thus, after performing word decomposition, lexeme; morph; and pattern features are assigned to the resulting morphemes separately. Finally, the LM training corpus is re-written so that every word or morpheme is replaced by a vector of features as in the form: {*W-<word>:L-<lexeme>:M-<morph>:P-<pattern>*}. A sequence of individual vector components defines a feature stream (class stream). An example of a feature vector for the full-word والشرقية (*and the eastern*) using the well known Buckwalter transliteration is: *wAl\$rqyp* → {*W-wAl\$rqyp:M-conj+art+AJ-FEM-SG:L-\$rqy:P-wAlCCCyp*}. Given that the word *wAl\$rqyp* is decomposed into *wAl+ \$rqyp* and that *root(wAl\$rqyp) = \$rq*, then the morpheme features are written as: *wAl\$rqyp* → {*W-wAl+:M-conj+art:L-wAl+:P-NUL*} {*W-\$rqyp:M-AJ-FEM-SG:L-\$rqy:P-CCCyp*}. From these examples, it can be seen that a careful handling of word morphological features could help to produce valid features for morphemes, these are called *morpheme level classes*.

2.2. German

The decomposition of German words is performed using a data-driven tool called *Morfessor* [24]. It is a statistical tool that automatically discovers the optimal decompositions 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 [11]. In a previous work [25, 3], morphemes generated via the Morfessor tool were successfully used to model a fraction of the vocabulary words leading to a significant improvement in the WER for German LVCSR compared to a traditional full-word based system. Therein, it was found out that keeping 5k most frequent decomposable full-words without decomposition (out of 100k items) is quite helpful for recognition performance.

It is stated in [24] that ignoring word counts in a given corpus and using only the corpus vocabulary to train the Morfessor model produces decompositions that almost resemble the 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. Less frequent words are not included in training in order to avoid irregularities that are harmful to the training process. In addition, the resulting decompositions are modified to remove very short and noisy morphemes. The final set of morphemes appear linguistically meaningful, where the most dominant decompositions are mainly the decomposition of the compound words and the stripping off the common prefixes.

Word features are generated using the *TreeTagger* [26]. It is a probabilistic tool that uses decision trees for annotating text with part-of-speech (POS) and lemma information. Lemma is the canonical baseform of the word. The TreeTagger has been successfully used to tag words of many languages including German [26]. 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 *class-index*. This is a data-driven class-index assigned to every word or morpheme after running a data-driven 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) [27, 28], then these vectors are clustered into 250 clusters using a k-means approach. A detailed description of this algorithm is found in a previous publication [25].

Similar to Arabic, the LM training corpus is preprocessed so that every word or morpheme is replaced by a vector of features: {*W-<word>:P-<pos>:B-<baseform>:I-<class-index>*}. An example of a feature vector is: *eingeschlafen* → {*W-eingeschlafen:P-VVPP:B-einschlafen:I-224*}; where VVPP means past participle verb. Given that the word *eingeschlafen* (*fallen asleep*) is decomposed into *ein+ geschlafen*, then the morpheme level features are written as: *eingeschlafen* → {*W-ein+:P-ART:B-ein:I-15*} {*W-geschlafen:P-VVPP:B-schlafen:I-192*}.

3. Morpheme level class-based LMs

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[1], we could build the model over sequences of some selected class stream, like sequences of lexemes, morphs or patterns. The CLM, initially described in [14], aims at combining the N-gram model over classes with the probability distribution of words in classes in order to better estimate smoothed probabilities over word sequences. Assuming that *ambiguous* class membership is used (also called soft clustering), where a word can be a member of multiple classes, then a bigram CLM is given by Equation 2, where a word is denoted by w_i and c_i is the class assigned to the word w_i at time i.

$$p(w_i|w_{i-1}) = \sum_{c_i,c_{i-1}} p(w_i|c_i)p(c_i|c_{i-1})p(c_{i-1}|w_{i-1}) \qquad (2)$$

[1] Whatever stated for words is also valid for morphemes

An analogous model could be estimated for morphemes with properly assigned classes. In Equation 2, it can be seen that there are only two component distributions required to estimate the class-based probability. The first component is the probability distribution over sequences of classes (called *class N-gram*). The second component is the probability distribution of words given classes (called *class membership definition*).

Normally, the standard word-based N-gram LMs perform better in capturing the relations between words for in-domain text. Therefore, 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 [29]. Here, we use linear interpolation of multiple word-based and class-based LMs expressed as:

$$p(W) = \sum_{i=1}^{I} \lambda_i p_w^{(i)}(W) + \sum_{j=1}^{J} \lambda_j p_c^{(j)}(W) \qquad (3)$$

where W is the word sequence, $p_w^{(i)}(W)$ is i^{th} word-based probability, $p_c^{(j)}(W)$ is the j^{th} class-based probability, λ_i, λ_j are the interpolation weights optimized on the development corpus, such that $\sum_{i=1}^{I} \lambda_i + \sum_{j=1}^{J} \lambda_j = 1$, and $(I+J)$ is the total number of the interpolated models.

4. Hierarchical Pitman-Yor LMs

A HPYLM is a type of Bayesian LM based on a coherent Bayesian probabilistic model that explicitly declares prior assumptions over the LM parameters [30]. It is based on the Pitman-Yor (PY) process, a nonparametric generalization of the Dirichlet distribution [31]. The PY process produces a *power-law* distribution over word frequencies that is found to be one of the most striking statistical properties in natural language.

Using the context of a unigram LM as in [16], let W be a finite vocabulary of V words. Let $G(w)$ be the probability of each $w \in W$, and let $G = [G(w)]_{w \in W}$ be the vector of word probabilities. We place a PY process prior on G:

$$G \sim PY(d, \theta, G_0) \qquad (4)$$

where the parameters of the process are: a discount parameter $0 \le d < 1$, a strength parameter $\theta > -d$, and a mean vector $G_0 = [G_0(w)]_{w \in W}$. $G_0(w)$ is the prior probability of word w, usually uniformly distributed, thus $G_0(w) = 1/V$.

Let $[x_l] = x_1, x_2, ...$ be a sequence of words drawn from G. The PY process is described as a generative procedure that iteratively produces $[x_l]$ with G marginalized out. This can be achieved by relating $[x_l]$ to another separate sequence of draws $[y_k] = y_1, y_2, ...$ from the mean distribution G_0 as follows. The first word x_1 is assigned the value of the first draw y_1 from G_0. Let t be the current number of draws from G_0 (currently $t = 1$), c_k be the number of words assigned the value of draw y_k (currently $c_1 = 1$), and $c = \sum_{k=1}^{t} c_k$ be the current number of draws from G. For each subsequent word x_{c+1}, we either assign it the value of a previous draw y_k with probability $\frac{c_k - d}{\theta + c}$ (increment c_k; set $x_{c+1} \leftarrow y_k$), or we assign it the value of a new draw from G_0 with probability $\frac{\theta + dt}{\theta + c}$ (increment t; set $c_t = 1$; draw $y_t \sim G_0$; *set* $x_{c+1} \leftarrow y_t$).

The above procedure is often referred to as the Chinese restaurant process [32]. Imagine a sequence of customers (corresponding to the draws from G) visiting a Chinese restaurant with an infinite number of tables (corresponding to the draws from G_0), each of which can accommodate an infinite number of customers. The first customer sits at the first available table, and each subsequent customer either joins an already occupied table (assign the word to a previous draw from G_0), or sits at a new table (assign the word to a new draw from G_0).

Now, an N-gram LM can be described as a hierarchical extension of the PY process. An N-gram LM defines probabilities over words given $N-1$ context words. Given a context $\mathbf{u}$, let $G_{\mathbf{u}}(w)$ be the probability of the current word taking on value w. A PY process is used as a prior for $G_{\mathbf{u}} = [G_{\mathbf{u}}(w)]_{w \in W}$:

$$G_{\mathbf{u}} \sim PY(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \qquad (5)$$

where $\pi(\mathbf{u})$ is the suffix context of $\mathbf{u}$ after dropping the earliest word. We recursively place a prior over $G_{\pi(\mathbf{u})}$ using 5, but with parameters $\theta_{|\pi(\mathbf{u})|}, d_{|\pi(\mathbf{u})|}$, and a mean vector $G_{\pi(\pi(\mathbf{u}))}$. This is repeated until reaching G_ϕ with an empty context, then a global uniform prior G_0 is placed on G_ϕ, where $G_0(w) = 1/V$:

$$G_\phi \sim PY(d_0, \theta_0, G_0) \qquad (6)$$

Starting from a posterior distribution over seating arrangements in a hierarchical Chinese restaurant, the predictive probability of a word w after a context $\mathbf{u}$ can be inferred using Gibbs sampling. A detailed inference scheme is described in [30].

5. Experimental setup

5.1. Arabic system

Arabic 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. The LM training corpora have around 206 Million running words including data from Agile Arab text, FBIS, TDT4 and GALE BN and BC data. A morpheme-based system with a 256k vocabulary is used. The 20k most frequent full-words are preserved without decomposition [7]. The 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 maximum likelihood linear regression (MLLR) adaptation is performed. In each pass, a morpheme-based bigram LM is used to construct the search space and to produce lattices. Then, these lattices are rescored using a morpheme-based 4-gram LM. Both the bigram and 4-gram LMs are smoothed using MKN smoothing. Additionally, in the third pass, we produce a set of N-best lists (N=500) which are rescored with 4-gram conventional or trigram HPYCLMs using different classes as described in Section 2.1. All LMs are estimated using the SRILM toolkit [18] with Bayesian LM extensions [17]. The recognition performance is evaluated on the GALE 2007 dev and eval sets [dev07: 2.5h; eval07: 4h].

5.2. German system

German AMs are also triphone models that are ML trained using 343h of audio material taken from 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. A morpheme-based system with a 100k vocabulary is used. The 5k most frequent full-words are preserved without decomposition [25]. The speech recognizer works in 2 passes. In the first pass, across-word AMs are used without speaker adaptation. A morpheme-based trigram LM is used to construct the search space and to produce lattices, then lattices are rescored with a 4-gram LM. Both the bigram and trigram LMs are smoothed using

MKN smoothing via the SRILM toolkit. The second pass performs speaker adaptation based on both CMLLR and MLLR. A standard trigram LM is used to generate N-best lists (N=500), then N-best rescoring is performed using 4-gram conventional or trigram HPYCLMs using different classes as described in Section 2.2. The recognition performance is evaluated on the Quaero 2009 dev and eval corpora [dev09: 7.5h; eval09: 3.8h].

6. Experiments

Table 1 shows the baseline recognition results for word- and morpheme-based LMs on Arabic and German corpora after a conventional 4-gram LM lattice rescoring. It can be seen that significant improvements are achieved in WER using a morpheme-based LM compared to word-based LMs. On top of morpheme-based systems, Tables 2(a) and 2(b) present the recognition results after the final rescoring using MKN and HPY LMs built over different classes for both Arabic and German corpora. The column labeled "MKN" shows the results using a MKN smoothed N-gram LM interpolated with a MKN smoothed CLM using the available features as classes. In the last row of each table, the interpolation is extended to include all the CLMs built on all the available classes (1 N-gram LM + 3 CLMs). In a similar fashion, the column labeled "MKN+HPY" shows the results using an interpolation of: MKN smoothed LM, a MKN smoothed CLM, a HPYLM, and a HPYCLMs. Similarly, in the last row, the interpolation is extended to include all the CLMs built on all the available classes (2 N-gram LMs + 6 CLMs). We can see that the best results are obtained when using the HPYLMs with all the available classes. This means that both the use of multiple features and the application of HPY models are beneficial. Significant WER reductions of [dev07: 0.5% absolute (3.5% relative); eval07: 0.4% absolute (2.5% relative); dev09: 0.7% absolute (2.2% relative); eval09: 0.6% absolute (2.1% relative)] are achieved compared to the baseline morpheme-based systems in Table 1. Although little improvements in the recognition performance are achieved over the conventional CLMs (that do not use the HPY models), the improvements are quite persistent and systematically consistent over all the available corpora. The observed WER improvements are considered statistically significant using a bootstrap method of significance analysis described in [33], the probability of improvement (POI_{boot}) ranges between 95% and 98%. It is worth noting that the character error rate (CER) improvements are almost going in-line with the WER improvements.

Table 1: *WER[%], CER[%], OOV[%] & PPL for* **WB***: word-based system,* **MB***: morpheme-based system;* **CER***: character error rate.*

		Arabic		**German**	
voc/LM	**metric**	**dev07**	**eval07**	**dev09**	**eval09**
WB	WER	14.9	16.5	32.8	28.4
	CER	6.9	8.7	14.5	12.9
	OOV	1.36	1.85	4.6	4.5
	PPL	442	577	326	344
MB	WER	14.2	16.1	32.3	28.0
	CER	6.6	8.7	14.4	12.7
	OOV	0.51	0.64	4.1	3.9
	PPL	392	510	379	406

7. Conclusions

We have introduced a novel methodology that combines the benefits of: morpheme-based LMs, feature-rich CLMs, along with HPYLMs to perform LVCSR for Arabic and German as examples of morphologically rich languages. The morpheme-based modeling aims at increasing the lexical coverage and reducing the data sparseness, while the use of morpheme level features in CLMs attempts to achieve better generalization to unseen word sequences. At the same time, the use of HPYLMs improves the smoothness of the N-gram probabilities over the conventional MKN smoothing for both normal and class-based models. We used different types of morphological and data-driven features for building CLMs. The best results are achieved by interpolating all the normal and the class-based LMs together. Proper tests have shown the statistical significance of the obtained WER improvements compared to the conventional morpheme-based LMs alone. Little but systematically consistent improvements are achieved over the conventional CLMs.

Table 2: *WER[%], CER[%] & PPL after final pass rescoring using* **MKN***: MKN smoothed N-gram + MKN smoothed CLM,* **MKN+HPY***: MKN smoothed N-gram + MKN smoothed CLM + HPY N-gram + HPYCLM;* **CER***: character error rate.*

(a) Arabic: 256k morpheme-based system (20k full-words + 236k morphemes) (**L**: lexeme; **M**: morph; **P**: pattern).

		MKN		**MKN+HPY**	
class	**metric**	**dev07**	**eval07**	**dev07**	**eval07**
L	WER	13.9	15.9	13.7	15.8
	CER	6.6	8.7	6.6	8.6
	PPL	375	484	354	458
M	WER	13.9	16.0	13.8	15.8
	CER	6.6	8.7	6.6	8.7
	PPL	386	500	362	471
P	WER	14.0	16.0	13.9	15.8
	CER	6.6	8.7	6.6	8.7
	PPL	389	505	363	474
L,M,P	WER	13.8	15.8	**13.7**	**15.7**
	CER	6.6	8.6	**6.6**	**8.6**
	PPL	372	480	**353**	**456**

(b) German: 100k morpheme-based system (5k full-words + 95k morphemes) (**B**: baseform; **P**: POS-tag; **I**: class-index).

		MKN		**MKN+HPY**	
class	**metric**	**dev09**	**eval09**	**dev09**	**eval09**
B	WER	32.2	27.7	31.9	27.5
	CER	14.4	12.5	14.2	12.4
	PPL	357	379	342	363
P	WER	31.9	27.5	31.8	27.4
	CER	14.3	12.5	14.2	12.4
	PPL	355	378	343	365
I	WER	31.9	27.5	31.7	27.4
	CER	14.4	12.4	14.2	12.4
	PPL	354	375	342	362
B,P,I	WER	31.8	27.5	**31.6**	**27.4**
	CER	14.2	12.5	**14.2**	**12.4**
	PPL	343	362	**333**	**352**

8. 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. Hermann Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

9. References

[1] M. C. Bateson, *Arabic language handbook*, ser. Georgetown Classics in Arabic Language and Linguistics Series. Portland, OR, USA: Georgetown University Press, 2003.

[2] A. Fox, *The structure of German*. New York, NY, USA: Oxford University Press, 2005.

[3] 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.

[4] M. Shaik, A. El-Desoky, R. Schlüter, and H. Ney, "Using morpheme and syllable based sub-words for Polish LVCSR," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May 2011, pp. 4680 – 4683.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. Chapman and Hall/CRC, Jul. 2003.

[16] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," in *Proc. Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, Jul. 2006, pp. 985 – 992.

[17] S. Huang and S. Renals, "Hierarchical Pitman-Yor language models for ASR in meetings," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Kyoto, Japan, Dec. 2007, pp. 124 – 129.

[18] 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.

[19] A. El-Desoky, R. Schlüter, and H. Ney, "Investigations on the use of morpheme level features in language models for arabic LVCSR," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012, pp. 5021 – 5024.

[20] A. El-Desoky, M. Shaik, R. Schlüter, and H. Ney, "Morpheme level feature-based language models for German LVCSR," in *Interspeech*, Portland, OR, USA, Sep. 2012.

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

[22] T. Buckwalter, *Buckwalter Arabic Morphological Analyzer Version 2.0*. Linguistic Data Consortium (LDC) catalogue, 2004, no. LDC2004L02.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] Y. W. Teh, "A Bayesian interpretation of interpolated Kneser-Ney," School of Computing, National University of Singapore, Tech. Rep. TRA2/06, 2006.

[31] J. Pitman and M. Yor, "The Two-Parameter Poisson-Dirichlet distribution derived from a stable subordinator," *The Annals of Probability*, vol. 25, no. 2, pp. 855 – 900, 1997.

[32] J. Pitman, "Combinatorial stochastic processes," UC Berkeley Dept. of Statistics, Technical Report 621, 2002.

[33] 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.