Exponential Language Modeling using Morphological Features and Multi-task Learning

Hao Fang, Mari Ostendorf, *Fellow, IEEE,* Peter Baumann, and Janet Pierrehumbert

Abstract

For languages with fast vocabulary growth and limited resources, data sparsity leads to challenges in training a language model. One strategy for addressing this problem is to leverage morphological structure as features in the model. This paper explores different uses of unsupervised morphological features in both the history and prediction space for three word-based exponential models (maximum entropy, logbilinear, and recurrent neural net (RNN)). Multi-task training is introduced as a regularizing mechanism to improve performance in the continuous-space approaches. The models are compared to non-parametric baselines. From using the RNN with morphological features and multi-task learning, experiments with conversational speech from four languages show we can obtain consistent gains of 7–11% in perplexity reduction in a limited-resource scenario (10 hrs speech), and 12–18% when the training size is increased (80 hrs). Results are mixed for all other approaches, compared to a modified Kneser-Ney baseline, but morphology is useful in continuous-space models compared to their word-only baseline. Multi-task learning improves both continuous-space models.

Index Terms

language model, neural network, morphology, limited resources

Manuscript received May 6, 2015; revised August 5, 2015; accepted September 10, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marcello Federico.

H. Fang and M. Ostendorf are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA.

P. Baumann is with the Department of Linguistics, Northwestern University, Evanston, IL, USA.

J. Pierrehumbert is with Oxford e-Research Centre, University of Oxford, Oxford, UK.

I. Introduction

Languages with rich morphological structure have high vocabulary growth which leads to data sparsity and makes it difficult to train a high-order n-gram language model (LM). The problem is particularly challenging for low-resource languages. The resulting weak LMs pose challenges for applications such as automatic speech recognition and machine translation, where LMs have a significant impact on performance.

In speech recognition, one approach to deal with n-gram sparsity in morphologically rich languages is to use subword vocabularies, such as syllables, morphologically-based subwords, or data-driven units such as graphones [1]. While some work has obtained good results by basing the vocabulary entirely on morphs (see [2]–[5] and references therein), in most scenarios better transcription performance is obtained with word-based vocabularies or with a combination of words and subwords, e.g. [6]–[10], since the shorter subwords are more acoustically confusable. In this work, we focus on word-based vocabularies, but the findings are relevant for mixed word/subword systems.

Given a word-based vocabulary in a limited resource scenario, data sparsity is a major challenge for the language model. Different methods have been developed for addressing the problem of data sparsity, including class LMs, n-gram and grammar-based LMs using sublexical units, and feature-based LMs. Class-based LMs (e.g. [11]–[13]) reduce the number of parameters via grouping words into classes for representing the history. LMs that expand words into a sequence of sublexical units (e.g. [2], [14]) reduce the number of free parameters by effectively reducing the vocabulary size. Feature-based models, which is the approach taken in this work, reduce the number of free parameters by using a feature vector to characterize a word that leverages advantages of the sublexical representation (when the features of a word are indicators of its sublexical components) in a factored rather than sequential representation. The features effectively associate a word with multiple classes. There are several variants of feature-based models, as described in the next section. For all three types of models, the classes/units can be defined in terms of linguistic categories or derived using data-driven unsupervised learning. For the limited resource scenario, our focus is on unsupervised learning.

This paper studies the use of unsupervised morphological features (or, "morphs") in the maximum-entropy (ME) LM [15], the log-bilinear (LBL) LM [16] and the recurrent neural network (RNN) LM [17], all of which share a similar exponential model form involving a mapping of the conditioning context (word history) to a vector space, either discrete or continuous. In addition to comparing these different model forms, we investigate the impact of using the morphological features only in the context

conditioning space vs. also in the word prediction (output) space as proposed in [18], [19], and the impact of using maximum log probability of the morph sequence together with word sequence log likelihood in the objective function (multi-task learning) as a form of regularization in training.

Experimental results are based on a standard limited resource training paradigm (transcripts for 10 hours of conversational speech), particularly aiming at languages with rich morphology and large vocabulary growth (Turkish, Bengali, Tamil and Zulu). We also show results on a less sparse data, i.e., transcripts with 80 hours of conversational speech. It is observed that while the morphological features do not improve the ME LM, they are beneficial to the LBL LM and the RNN LM, both of which use continuous-space context representations. Multi-task learning further improves the LBL LM and the RNN LM, and is shown to be more effective than output word factorization. In all four languages, the best performance is obtained using the RNN with morph features only for the context, in combination with multi-task learning. The various comparisons lead to insights into useful modeling assumptions for exponential LMs, particularly for the limited resource scenario.

The rest of the paper is organized as follows. In Section II, we review related work and outline the key differences in our approach. Details of the proposed neural network architecture with morphological features and the multi-task learning mechanism are provided in Section III. Experimental results and analysis are presented in Section IV, including details on the implementation of morphology learning and LM training. We conclude in Section V with a discussion of open questions and future directions for research.

II. Prior Work

In statistical language modeling, the morphological features of words can be leveraged in various ways to obtain more robust word predictions when n-gram coverage is sparse. The first and most widely used approach is the factored LM [20], [21], which addresses the problem of data sparsity by leveraging sublexical features of words in the backoff algorithm, providing more flexibility than the standard n-gram approach of sequential word backoff. Factored LMs are often used with morphological features, which have the potential advantage of introducing powerful constraints in language modeling [2], [20]. Stream-based, class-based and factored n-gram LMs are compared in [14] for Arabic speech recognition, where the best performance is obtained with the factored LM.

Features can also be incorporated in discriminative LMs, which use a perceptron or a ME framework in N-best rescoring. While the original work used only n-gram features [22], the framework has been used to combine morphological and syntactic features [5]. One can also think of word classes as features, which

have also been used with success in ME LMs [23], [24]. The ME framework is well-suited to incorporating different types of features and avoids the problems of more complex backoff strategies to leverage the features. A relative of the (loglinear) ME model is the logbilinear (LBL) model, introduced in [16] with a factored, low-rank representation of history and in [25], [26] with a combination of low-rank and sparse weights. A feature-based version of the factored LBL is described in [18] and shown to outperform a standard n-gram using the modified Kneser-Ney (mKN) backoff for six different languages on datasets with 1M tokens, including two high vocabulary growth languages (Russian, Czech). A feature-based version of the sparse-plus-low-rank model is explored for Turkish in [27], showing better performance than the factored n-gram.

The LBL LM can also be viewed as a neural network model but without a non-linear transformation; the low-rank transformation provides a continuous-space representation of words. Early work with continuous-space LMs was based on a feedforward neural network structure [28], [29], as are the feature-based models used in [30]. The factored (feature-based) neural model is shown to outperform the factored n-gram LM for Turkish in [30]. A word-character hybrid neural network LM outperforms the 4-gram LM for the Mandarin large vocabulary continuous speech recognition in [31]. More recently, deep [32] and recurrent [17] neural networks have been applied to language modeling with promising results. In particular, the RNN LM developed by Mikolov and colleagues has been shown to be the state-of-the-art LM by a large margin compared to other LMs when there are enough training data [17], [33]. However, as we shall see here, there is only a small gain from the RNN LM over the mKN smoothed n-gram LM in the low-resource scenario, which we hypothesize is due to the data sparsity problem. In [34], the authors propose an RNN LM using subword units, which addresses the data sparsity problem, but the subword-only vocabulary is not as effective for speech recognition. Sublexical features have been successfully used in word-based RNN models in [35] for a Dutch multi-style corpus (with part-of-speech tags and lemmas used as features in context words), in [36] for English news text (with part-of-speech tags, lemmas, and stems as features in context words), and in [19] for Chinese Twitter (with syllables used as features for both context and prediction words).

The use of features in [18], [19], [35], [36] all correspond to additive adjustments to the base continuous-space representation of a word (embedding). In [19], a word-dependent scaling factor is incorporated to weight the adjustments depending on the frequency of the word, further improving perplexity (PPL). Many of the models using features include them only in representations of context (word history), e.g. [30], [32], [35], [36]. Alternatively, the features may also be used in the word prediction space, as in [18], [19].

To summarize, in the various approaches to leveraging morphology features in word-based language modeling, a number of different dimensions have been explored, but most of the comparisons consider only a small number of options. In addition, the issue of sparse training data (for which the morphological features are particularly relevant) has not been a primary focus. In this work, we explore multiple dimensions within the general framework of exponential models, leveraging ME, LBL and RNN models, and comparing these to a non-parametric factored n-gram LM baseline. Specifically, we compare

- joint and discrete (ME) vs. additive and continuous (LBL, RNN) morphological adjustment terms;
- fixed (ME, LBL) vs. recurrent histories (RNN); and
- use of morphological features in the history (+c) vs. also in the prediction space (++).

Of particular importance, we further studied the use of combining the additive word representation (LBL, RNN) and multi-task learning, which we find has a greater impact than all these variants on successful use of morphological features. The focus on the limited resource scenario and conversational speech also differentiates our work from most prior continuous-space language modeling studies. For example, the work here leverages limited training sets of less than 100k words or full training sets of less than 600k words, compared to 1M words for the limited training case in [18], and 4M words in [19].

III. Model Description

A. Model Structures Without Morphological Features

We are interested in three different LM structures, i.e., the ME LM [15], the LBL LM [16], and the RNN LM [17]. All three kinds of LMs estimate the probability of a word w conditioned on the context words $\mathbf{h}$ as follows,

$$\Pr(w|\mathbf{h}) = \frac{\exp(\mathbf{r}_w^T \cdot \mathbf{q}_\mathbf{h} + b_w + b_0)}{\sum_{v\in\mathcal{V}} \exp(\mathbf{r}_v^T \cdot \mathbf{q}_\mathbf{h} + b_v + b_0)} \tag{1}$$

where $\mathcal{V}$ denotes the vocabulary, $\mathbf{r}_w^T \in \mathbb{R}^D$ is the weight vector for word w being predicted, $\mathbf{q}_\mathbf{h} \in \mathbb{R}^D$ is the feature vector for context words (or history) $\mathbf{h}$, b_w's are the bias terms encoding the prior unigram distribution, and b_0 is a global bias term shared by all words.

The three models differ in the representation of context (word history). Specifically, using $\mathbf{q}_i$ to represent the history $\mathbf{h}_i$ at time i

$$\mathbf{q}_i^{\text{ME}} = \mathbf{I}(\mathbf{h}_i) \tag{2}$$

$$\mathbf{q}_i^{\text{LBL}} = \sum_{j=1}^{n-1} \mathbf{C}_j \mathbf{s}_{w_{i-j}} \tag{3}$$

$$\mathbf{q}_i^{\text{RNN}} = f(\mathbf{s}_{w_{i-1}} + \mathbf{U}\mathbf{q}_{w_1^{i-2}}) \tag{4}$$

where $\mathbf{I}(\cdot)$ is a concatenation of binary indicator vectors (e.g., one-hot vectors for n-grams of different length), $\mathbf{s}_{w_{i-j}}$ is a D-dimensional continuous-space representation of the j-th word in the history, $\mathbf{C}_j$ and $\mathbf{U}$ are automatically-learned matrix transformations, and $f(\cdot)$ is a nonlinearity. For the ME model, the context representation is very high dimensional and very sparse. The LBL and RNN models use a relatively low-dimensional continuous-space context representation (also referred to as a distributed representation or embedding), which incorporates a matrix transformation from a one-hot indicator vector representation of words v in the vocabulary $\mathbf{s}_v = \mathbf{S}I(v)$. Both the ME and LBL models use a fixed-length context window $h_i = w_{i-n+1}, \ldots, w_{i-1}$, while the RNN represents the full sentence history $h_i = w_1, \ldots, w_{i-1}$, by using a recursive update of the context state. The two fixed-length context models differ in that the ME representation uses indicators for specific n-grams, whereas the LBL represents n-grams by concatenating context-dependent unigram indicators. A difference between the two continuous-space models explored here is the use of multiple position-dependent transformations $\{\mathbf{C}_j \in \mathbb{R}^{D\times D}; j = 1, \ldots, n-1\}$ for the $n-1$ different words in the context history for LBL vs. a single transformation $\mathbf{U} \in \mathbb{R}^{D\times D}$ of the previous context history in the RNN. (C_j can be thought of as the transformation of the embedding of the j-th word in the history that makes it more useful for word prediction.) Since we tune the dimension D of the continuous space representation for each model based on a development set, this is not so much a difference in the number of parameters as it is in the explicit representation of relative word position in the LBL case. In addition, the RNN uses a nonlinear transformation on each element of the context vector (unlike the LBL), specifically the Sigmoid function:

$$\text{Sigmoid}(x) = \frac{1}{1+\exp(-x)}. \tag{5}$$

The three different models are illustrated with a neural network style representation in Figs. 1–3 to contrast the structures. In all cases, the initial representation of word context takes a binary form, and the normalization in equation (1) corresponds to a softmax output layer:

$$\text{Softmax}(x) = \frac{\exp(x)}{\sum_{x'\in\mathcal{X}} \exp(x')} \tag{6}$$

where the denominator sums over all neurons of the output layer and ensures the activations of the output layer form a probability distribution.

In the experiments here, we use two different implementations of the ME LM: the SRILM [38] and the hash-based implementation of ME LM as described in [39]. We implemented the hash-based ME LM, the LBL LM, and the RNN LM for our experiments following the same design on most implementation details but incorporating noise contrastive estimation (NCE) [37] in training and modifications described in subsequent sections.

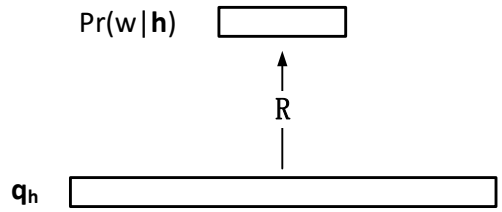


Fig. 1: Structure of the ME LM: The binary input layer is directly connected to the softmax output layer using the weight matrix $\mathbf{R}$.

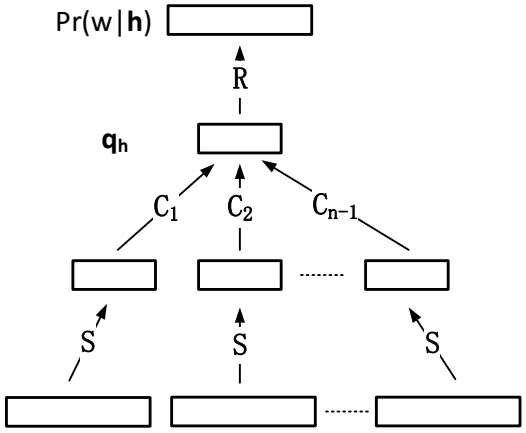


Fig. 2: Structure of the LBL LM: The input layers use a one-hot encoding for each context word, each of which is connected to the projection layers using the same projection matrix $\mathbf{S}$, and the projection layers are connected to the hidden layer using position-dependent weight matrices $\mathbf{C}_j$. Finally, the hidden layer is connected to the softmax output layer using the weight matrix $\mathbf{R}$. The projection layers and the hidden layer usually have the same size, i.e., $\mathbf{C}_j$'s are square matrices [37].

B. Model Structures With Morphological Features

For each word, we can obtain a morph decomposition via unsupervised learning. For example, the morph decomposition learned for the Turkish word "dizgelerimde" (meaning "on my lists") is "diz ge ler im de", i.e., "arrange - Der.Noun - Plural - 1SG.Possessive - Locative". Details about the morphology learning are provided in Subsection IV-B.

Two methods are used to incorporate morphological features into the three different LMs. The first set of LMs (ME+c, LBL+c, and RNN+c) exploit morphological decomposition for the context words only;

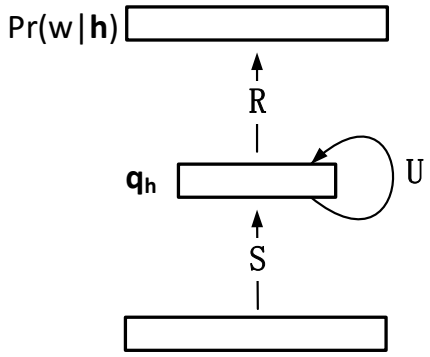


Fig. 3: Structure of the RNN LM: The input layer uses a one-hot encoding for the closest context word, which is connected to the hidden layer using the weight matrix $\mathbf{S}$, and the hidden layer is recurrently connected to the last hidden layer using the weight $\mathbf{U}$. A Sigmoid transformation is applied to the hidden layer, which is then connected to the softmax output layer using the weight matrix $\mathbf{R}$.

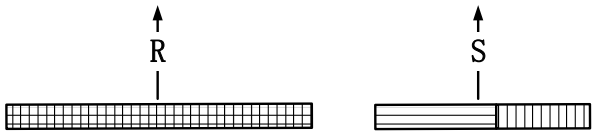


Fig. 4: Left: input layer for ME+c LM, a binary feature vector representing all possible combinations of word and/or individual morphs to characterize the word history. Right: input layer for LBL+c and RNN+c LMs, a binary vector of length $V_1 + V_2$ representing a context word (dimension V_1) and its morphs (dimension V_2). Horizontal lines and vertical lines represent words and morphs, respectively. Mixed horizontal and vertical lines represent mixed-word-morph features.

the word being predicted is kept in the word form. As shown in Fig. 4, they augment the input word index by encoding the morphs for the context words at the input layer(s). The models in the second set of LMs (ME++, LBL++, RNN++) additionally utilize the morph decomposition for the word being predicted. Fig. 5 illustrates this method by adding an itermediate factor hidden layer between the layer representing $\mathbf{q}_h$ and the output layer. The weight matrix $\mathbf{G}$ between $\mathbf{q}_h$ and the factor hidden layer needs to be learned, whereas the dictionary-specific weight matrix $\mathbf{F}$ between the factor hidden layer and the output layer is determined by the morph decomposition of the word.

ME+c LM: With the morph decomposition available, there are multiple ways to factor the word-only

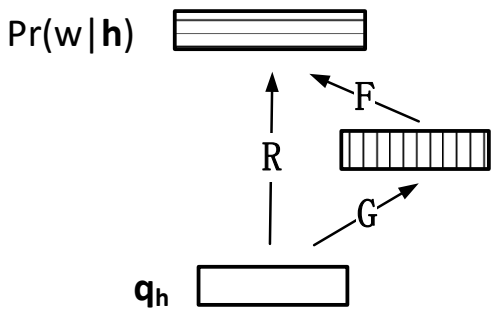


Fig. 5: Additional factor hidden layer between the layer representing $\mathbf{q}_h$ and the output layer. The factor hidden layer is a linear layer with no transformation. It is connected to the softmax output layer with a dictionary-specified weight matrix $\mathbf{F}$ representing the mapping from the word to its morphs.

n-gram into mixed-word-morph n-gram: the j-th unit of the n-gram can be associated with any one of the morphological decomposed units of the j-th word or the word itself. For example, if there are 3 morphs for each of the two context words in a trigram, we can have 16 (4^2) ways to factor the 4 features (the 3 morphs and the word) of each of these two words. The resulting binary vector q_h has 16 non-zero entries for the trigram, as well as additional (analogous) non-zero entries for the bigram and unigram contexts. Therefore, many more features are obtained. This way of constructing ME features is inspired by the factored LM [20], [21], where for each word n-gram, it considers multiple backoff paths according to different ways to factor the context words.

LBL+c and RNN+c LMs: Instead of using binary vectors of length V_1 for context words, the input layers are now vectors of length $V_1 + V_2$ where V_1 and V_2 are the vocabulary (inventory) sizes of words and morphs, respectively. For each word w, the entry representing the word is set to 1, and the entries representing the morphs are set to the their occurrence in the morph decomposition of the word. In this way, the continuous representation of a context word becomes the sum of its word-form representation and its morph-form representations, which is called an additive word representation in [18]. For the LBL case, the model structure is identical to the LBL+c LM described in [18].

ME++ LM: The ME+c LM only decomposes the context words. The word being predicted can be decomposed into morphs as well. The ME++ LM utilizes this additional information by adding ME features that represent the resulting mixed-word-morph n-grams. Considering the trigram example, this time with a word to be predicted with 2 morphs, there are 3 ways to factor the predicted word, combined with the 16 ways to factor the context words, giving 48 (16×3) mixed-word-morph features for this

word triple. This is equivalent to inserting an additional factor hidden layer of size V_2 between the input layer and the softmax output layer. Each entry of the factor hidden layer represents a morph in the morph vocabulary. No transformation is done on the factor hidden layer. The weight matrix $\mathbf{F}$ between the factor hidden layer and the softmax output layer is not learned, but rather is defined by the mapping from words to their morphs.

LBL++ and RNN++ LMs: As discussed in [18], the additive word representation can be used on the word being predicted as well as the context words. This leads to the LBL++ LM described in [18]. Compared with the LBL+c LM, there is an additional factor hidden layer of size V_2 between the hidden layer and the softmax output layer. Here, we apply the same idea to the RNN to obtain the RNN++ model. As for the ME++ LM, each entry of the factor hidden layer represents a morph in the morph vocabulary, and no transformation is done on the factor hidden layer. The weight matrix $\mathbf{F}$ between the factor hidden layer and the softmax output layer is simply defined by the mapping from words to their morphs.

C. Multi-task Learning Objective

The LBL++ LM and the RNN++ LM utilize the morph decomposition for the word being predicted by the additional factor hidden layer. In this subsection, we study another way to utilize such information via multi-task learning. To train the LBL+c LM and the RNN+c LM, the objective function can be modified to the log-likelihood of the word sequences plus the weighted log-likelihood of the morph sequences, i.e.,

$$\begin{aligned}&\sum_{t=1}^{T}\left[\log\Pr(w_t|w_1^{t-1})+\mu\log\Pr(m_{K^{(t)}}^{(t)},\ldots,m_1^{(t)}|w_1^{t-1})\right]\\ =&\sum_{t=1}^{T}\left[\log\Pr(w_t|w_1^{t-1})+\mu\sum_{k=1}^{K^{(t)}}\log\Pr(m_k^{(t)}|w_1^{t-1})\right]\end{aligned}\tag{7}$$

where w_t is the word at time t, $m_k^{(t)}$ is the k-th morph in the morph sequence of w_t, $K^{(t)}$ is the length of the morph sequence of w_t, and μ is the weight on the log-likelihood of morph sequences in the training objective function. As reflected in the multi-task learning objective (7), the morphs of the predicted word are assumed to be independent with each other conditioned on context words. Note if two words share the same morph units but have different ordering of the morph sequences, the training objective will be different due to the term $\Pr(w_t|w_t^{t-1})$.

To train the LBL+c LM and the RNN+c LM with the multi-task learning objective, the models need to have an additional softmax output layer for morphs of the predicted word, as shown in Fig. 6. In this way,

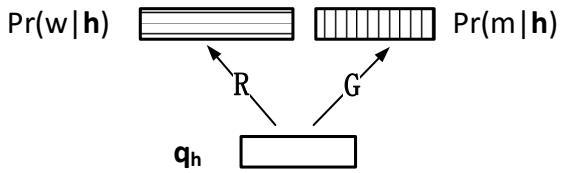


Fig. 6: Additional softmax output layer for factors when using multi-task learning. The hidden layer is connected to the softmax output layers for words and morphs via the weight matrices $\mathbf{R}$ and $\mathbf{G}$, respectively.

additional error signals are obtained from the softmax output layer for morphs and are back-propogated to the hidden layer and other layers. Note that it does not make sense to train the ME+c LM with multi-task learning since the input layer is directly connected to the softmax output layer.

IV. Experiments

We conduct multiple experiments in order to assess the impact of different model structures and the multi-task learning mechanism on PPL. In this section, we first describe the corpora used in the experiments, the morphology learning approach, and the training protocol. Then we present a series of baseline LM experiments on four different languages to compare the word-based exponential models to the standard n-gram and factored n-gram models. Finally, we present a series of LM experiments on four different languages to compare the X LMs (X is ME, LBL, or RNN), the X+c LMs, the X++ LMs and the X+c LMs with multi-task learning.

A. Corpora

Our experiments use the transcriptions of the IARPA Babel conversational speech data. Specifically, we investigate four languages: Turkish (IARPA-babel105b-v0.4), Benagli (IARPA-babel103b-v0.4b), Tamil (IARPA-babel204b-v1.1b), and Zulu (IARPA-babel206b-v0.1e) [40]. The transcribed data are post-processed by using a common special token to replace all tokens of speech events and non-speech events such as mispronounced words, background noise, etc., except <hes> and <laugh> which represent verbal hesitations (filled pauses) and laughter, respectively. For each language, the LMs are trained on two versions of training set, i.e., the 10-hour limited language pack (LLP) and the 80-hour full language pack (FLP). For each language, another 10-hour of speech data is split into a development (dev) set for

	Statistics	Turkish	Bengali	Tamil	Zulu
LLP	vocabulary size	10k	8k	14k	14k
	# tokens	71k	76k	74k	61k
	% OOV tokens (dev)	16.2	11.7	19.3	23.6
	% OOV tokens (test)	15.4	11.7	19.8	24.0
FLP	vocabulary size	38k	24k	52k	54k
	# tokens	554k	464k	438k	362k
	% OOV tokens (dev)	8.9	6.7	12.2	14.3
	% OOV tokens (test)	8.1	6.7	12.7	15.2
	# dev tokens	23k	25k	25k	19k
	# test tokens	48k	49k	49k	41k

TABLE I: Statistics of Babel data with a word-based vocabulary.

	Statistics	Turkish	Bengali	Tamil	Zulu
LLP	vocabulary size	3k	1k	2k	4k
	# tokens	122k	164k	176k	119k
	% OOV tokens (dev)	2.5	1.4	1.2	1.7
	% OOV tokens (test)	2.2	1.4	1.5	1.8
	# dev tokens	41k	54k	59k	39k
	# test tokens	85k	108k	117k	85k
FLP	vocabulary size	9k	2k	4k	10k
	# tokens	879k	871k	842k	631k
	% OOV tokens (dev)	2.4	1.6	1.5	1.5
	% OOV tokens (test)	2.1	1.6	1.8	1.5
	# dev tokens	37k	47k	48k	33k
	# test tokens	76k	93k	94k	73k

TABLE II: Statistics of Babel data with a morph-based vocabulary.

parameter tuning during the training, and a test set used for PPL evaluation.[1] The statistics of the data before morph decomposition are summarized in Table I. All four languages in the LLP case have OOV token rates higher than 10%, as well as Tamil and Zulu in the FLP case (see rows "% OOV tokens" in Table I). In addition, a large fraction of the word types in the LLP training data occur only once or twice – roughly 70-80% for these languages, as explored further in the next section. This severe data sparsity

[1]The test set is an internal test set, not the official BABEL eval set.

	Turkish	**Bengali**	**Tamil**	**Zulu**
% w tail types in training	72.6	67.1	78.7	80.9
% w+m tail types in training	3.5	0.8	0.9	4.5
% w tail tokens in training	12.7	8.7	17.9	21.5
% w+m tail tokens in training	0.7	0.1	0.2	1.3
% w tail types in dev	19.8	17.6	16.9	16.6
% w+m tail types in dev	0.9	0.2	0.1	0.9
% w tail tokens in dev	6.3	4.8	6.5	7.9
% w+m tail tokens in dev	0.3	0.1	0.0	0.5
% w tail types in test	19.3	19.3	17.0	16.0
% w+m tail types in test	0.9	0.2	0.2	0.9
% w tail tokens in test	6.2	5.0	6.8	7.5
% w+m tail tokens in test	0.4	0.1	0.1	0.5

TABLE III: Statistics of Babel LLP data tail word coverage, defined with and without consideration of morph coverage. w: word; w+m: word+morph.

problem makes it difficult to train a robust LM for those words.

B. Morphology Learning

The morph features were learned in a fully unsupervised manner using the state-of-the-art morphological segmentation toolkit *Morfessor Categories-MAP* [41], which is a standard used in many other studies. To obtain word-internal segmentations, Morfessor recursively splits words into morphs, and tries to find a lexicon of morphs that is both accurate and minimal given the corpus by maximizing the posteriori probability

$$\begin{aligned} &\mathrm{argmax}_{\mathrm{lexicon}} \Pr(\mathrm{lexicon}|\mathrm{corpus}) \\ &= \mathrm{argmax}_{\mathrm{lexicon}} \Pr(\mathrm{corpus}|\mathrm{lexicon}) \Pr(\mathrm{lexicon}), \end{aligned} \tag{8}$$

which is a variant of the minimum-description-length principle. The learned morphs are labeled as prefixes (PRE), stems (STM) or suffixes (SUF) using a hidden Markov model and are restricted to represent words of the form $(\mathrm{PRE}^* \ \mathrm{STM} \ \mathrm{SUF}^*)^+$, where $*$ and $+$ represent the Kleene star and Kleene plus, respectively.

The degree of segmentation can be manipulated via Morfessor's PPL threshold parameter τ, which controls the likelihood of a given morph being a prefix or suffix in the context of a word. The optimal value of this parameter is usually determined using a labeled development set [41], as its effect strongly

depends on the morphological structure of the language and the size of the training set. Since we do not have any labeled data, we used an extensive search over τ to find a value that minimizes the percentage of rare morphs in the lexicon while still providing high coverage of the corpus.

For all four languages under the LLP and FLP conditions, we trained Morfessor on the transcriptions of the corresponding training data, i.e. the words in their standard orthography or grapheme representation were used as training data. The learned models were then used to obtain a morphological segmentation of the entire development set.

The statistics of the data after morph decomposition based on the learned segmentation are shown in Table II. Note the number of dev and test tokens are less under the FLP condition than that under the LLP condition, since the morphs learned under the FLP condition are longer in average. Compared with Table I, it can be observed that the morph-only vocabulary is much smaller than the word-only vocabulary, and the morph OOV rate is lower – all of which we would expect from a good morph decomposition. This motivates the use of morphs for alleviating the data sparsity problem.

For a word-based LM, we are interested in how well the observed words are covered in terms of their component morphs. Table III shows how morph features of a word can improve coverage in the LLP case. We define word "tail" types as words that occurred only once or twice in the training text,[2] and word+morph tail types as words for which all morphs in the word occur once or twice in the morph-expanded training data. The table shows that the use of morphs introduces features that allow generalization of a large fraction of these tail word type: 73-81% of the tail words in the different languages have more frequent representation of their component morphs. The precentage of tokens represented by these tail tokens is sizable in the training data (9–21%). In the dev and test sets, these tokens are less frequent (5–8%) but still non-trivial. The tail rate is reduced to less than 1% when morphs are used as features, so improved estimation of these rare words can have an impact.

C. LM Training and Tuning

When training the neural network LMs, computing the denominator in (1) is very expensive. In our experiments, noise-contrastive estimation (NCE) is used to avoid normalization during training [37]. For each word/morph, 50 noise samples are used for all experiments. In other words, for the multi-task

[2]Another reasonable definition of tail words can include OOV words. However, since the vocabulary includes all the words in the training data, there are no OOVs in the training data; hence, the LMs cannot learn anything for OOV words. Thus, we do not include OOV words as tail words in the statistics here.

learning, if there are K morphs for the word being predicted, then $50K$ noise samples for the morphs are used in addition to the 50 noise samples for the word.

All weight matrices except the weight matrices in the ME structures are initialized randomly according to a sharp, zero-mean Gaussian distribution. Bias terms and the weight matrices for the ME LM, the ME+c LM and the ME++ LM are initialized as zeros.

The standard back-propagation algorithm with stochastic gradient descent is used. For RNN strucutres, the truncated back-propagation through time (BPTT) algorithm is used [17]. The BPTT unfold level is set to 1 in our experiments; we found that a larger BPTT unfold level does not improve the LMs using the studied corpora and it has higher computation cost. No regularization is applied. The training data are randomly shuffled at the sentence level, i.e. changing the order that sentences are presented during each epoch of training. By doing so, the PPL can be reduced by around 3-4% compared to using non-shuffled training data [39]. The learning rate is initialized as some value α_0. Once the PPL of the dev set increases, it restores the model parameters in the last epoch and the learning rate is halved at each new epoch. The training is terminated when the PPL of the dev set increases for the second time.

The initial learning rate α_0, the weight μ in the multi-task learning objective (7), and the dimension D of the feature vector for LBL and RNN structures are separately tuned for each model based on the PPL on the dev set. For ME and LBL structures, the order of the n-gram is 3 throughout the experiments in the paper. No benefit was found from higher-order n-grams in early experiments due to limited amount of training data. To train the ME LMs, we use a hash array of length 10^8 to store the feature weights for word n-grams; to train the ME+c LMs and ME++ LMs, an additional hash array of length 10^9 is used to store the feature weights for the mixed-word-morph n-grams and morph n-grams.

D. Baselines

Several baseline LMs are implemented, including:

- mKN smoothed tri-gram LM. This model is trained using SRILM [42] and does not use morphological features.
- Non-hash-based ME tri-gram LM. This is non-hash-based ME LM and is trained using SRILM [42]. No morphological features are used. No ℓ_2 regularization is used.
- Non-hash-based ME tri-gram LM + ℓ_2. This ME LM is trained with ℓ_2 regularization. It is effective in reducing the PPL of the ME LMs. The ℓ_2 regularization parameter is tuned according to the PPL on the validation data.

Language Model	Turkish	Bengali	Tamil	Zulu
mKN backoff LM	242.8	255.1	395.6	275.7
Non-hash-based ME LM	256.6	272.5	425.5	287.0
Non-hash-based ME LM + ℓ_2	**234.0**	**244.4**	**386.1**	**262.7**
Hash-based ME LM	239.5	252.5	389.1	265.2
LBL LM	277.2	267.0	438.0	309.1
RNN LM	243.5	248.0	407.3	278.4
Factored LM	254.3	259.1	443.8	277.4

TABLE IV: PPL of different baseline trigram LMs on the test sets under the LLP condition. Bold numbers indicate the lowest PPL of all baseline LMs, whereas numbers with underline indicate lower PPL than the mKN backoff LM.

Language Model	Turkish	Bengali	Tamil	Zulu
mKN backoff LM	293.8	275.1	501.5	410.5
Non-hash-based ME LM	309.4	293.7	533.2	427.1
Non-hash-based ME LM + ℓ_2	282.0	263.7	483.5	391.8
Hash-based ME LM	296.7	279.4	498.0	396.4
LBL LM	303.1	276.4	522.5	442.2
RNN LM	**277.9**	**253.4**	**455.7**	**394.2**
Factored LM	307.4	288.9	547.1	418.1

TABLE V: PPL of different baseline trigram LMs on the test sets under the FLP condition. Bold numbers indicate the lowest PPL of all baseline LMs, whereas numbers with underline indicate lower PPL than the mKN backoff LM.

- Factored tri-gram LM. This model uses the morphological features. It is trained using SRILM [42] and is tuned based on the PPL on the validation set using a genetic algorithm to learn the back-off strategy [43] with the publicly-available tool at http://ssli.ee.washington.edu/people/duh/research/gaflm.html.

Tables IV (LLP) and V (FLP) summarize the PPL of the different baseline trigram LMs for the four languages explored here, with the word-based results for three exponential models (hash-based ME, LBL and RNN) for comparison. The PPL under the FLP condition is higher than that under the LLP condition as the vocabulary covers more words. The only method that consistently beats the baseline mKN LMs for both LLP and FLP is the non-hash-based ME LM with regularization. This gives the best result

for the LLP condition, but the RNN has better results for the FLP condition. Besides the mechanism for storing the feature weights, the training of the hash-based ME LMs differs from the training of the non-hash-based ME LMs, and some preliminary experiments show no improvement on the hash-based LM can be gained via ℓ_2 regularization. Also, when training the hash-based ME LMs, a validation set is used for early stopping, which should avoid overffiting. This explains why the hash-based ME LMs are better than the non-hash based ME LMs without ℓ_2 regularization.

The LBL and RNN word-based models have very different performance, with the RNN model coming close to or beating the mKN baseline, but the LBL is usually among the worst performing models. This may be due to overtraining for the LBL given the large number of free parameters associated with the position-dependent weight matrices, although performance is poor for both LLP and FLP conditions and the results in [37] with diagonal matrices do not beat the mKN baselines until using a longer context window.

Although the factored LM utilizes the morphological features, no gain can be obtained over the mKN baseline for any of the four languages for both LLP and FLP cases. It is not simply a matter of overtraining the genetic algorithm, however, since performance degrades on both the tuning and evaluation sets. In the remaining experiments, comparisons will only be with the mKN and best case ME baselines.

E. Main Results

Tables VI–IX provide the complete set of results for the different exponential LMs for the four languages. The X+c and X++ LMs use morphological features (X is ME, LBL, or RNN), whereas all other LMs do not use morphological features. In Figs. 7 and 8, we show under the LLP condition, the relative reduction in PPL associated with the different uses of morph features for the LBL and RNN models repectively, comparing to the model-specific baseline. (The ME models degrade with the use of morphology, so that figure is omitted.) In the discussion to follow, we highlight comparisons related to different modeling assumptions. While the discussion emphasizes the LLP results, the main findings hold for the FLP condition and the relative improvements for the best case models are larger.

Morph Features in ME vs. Continuous-Space Models

For the LMs with ME structures implemented using the one-dimensional hash array, using morph features of context words (ME+c) often degrades performance and only benefits performance in the Zulu FLP condition. For the ME model, using morph features for both context and prediction words (ME++) degrades performance relative to the word-only model for all conditions. For the LBL and RNN models,

Language Model	LLP	FLP
mKN tri-gram LM	242.8	293.8
Non-hash-based ME tri-gram LM + ℓ_2	**234.0**	**282.0**
ME LM	239.5	296.7
ME+c LM	242.7	296.3
ME++ LM	256.3	308.3
LBL LM	277.2	303.1
LBL+c LM	257.7	289.1
LBL++ LM	257.4	291.9
LBL+c LM (multi-task)	250.4	279.6
RNN LM	243.5	277.9
RNN+c LM	232.1	262.3
RNN++ LM	237.1	262.9
RNN+c LM (multi-task)	**224.6**	**250.8**

TABLE VI: PPL of different LMs on test set for Turkish.

Language Model	LLP	FLP
mKN tri-gram LM	255.1	275.1
Non-hash-based ME tri-gram LM + ℓ_2	**244.4**	**263.7**
ME LM	252.5	279.4
ME+c LM	256.9	276.4
ME++ LM	269.6	290.3
LBL LM	267.0	276.4
LBL+c LM	249.2	261.0
LBL++ LM	249.5	258.6
LBL+c LM (multi-task)	241.0	256.6
RNN LM	248.0	253.4
RNN+c LM	234.5	243.6
RNN++ LM	234.7	241.2
RNN+c LM (multi-task)	**227.1**	**237.3**

TABLE VII: PPL of different LMs on test set for Bengali.

on the other hand, using morphs with context words (LBL+c, RNN+c) always leads to improvement over the baseline for that model. We hypothesize that the ME LMs with morphological features overfit to the training data very quickly since there are too many features due to the joint n-gram indicators. Thus,

Language Model	LLP	FLP
mKN tri-gram LM	395.6	501.5
Non-hash-based ME tri-gram LM + ℓ_2	**386.1**	**483.5**
ME LM	389.1	498.0
ME+c LM	403.9	498.6
ME++ LM	431.1	535.5
LBL LM	438.0	522.5
LBL+c LM	413.0	477.4
LBL++ LM	404.5	472.2
LBL+c LM (multi-task)	396.2	462.4
RNN LM	407.3	455.7
RNN+c LM	375.8	428.9
RNN++ LM	372.8	434.4
RNN+c LM (multi-task)	**355.0**	**412.7**

TABLE VIII: PPL of different LMs on test set for Tamil.

Language Model	LLP	FLP
mKN tri-gram LM	275.7	410.5
Non-hash-based ME tri-gram LM + ℓ_2	**262.7**	**391.8**
ME LM	263.1	396.4
ME+c LM	265.2	386.0
ME++ LM	279.3	401.5
LBL LM	309.1	442.2
LBL+c LM	288.4	398.3
LBL++ LM	284.5	388.2
LBL+c LM (multi-task)	280.8	384.6
RNN LM	278.4	394.2
RNN+c LM	261.2	354.8
RNN++ LM	259.7	354.0
RNN+c LM (multi-task)	**250.5**	**339.5**

TABLE IX: PPL of different LMs on test set for Zulu.

early stopping of the stochastic gradient descent is not very effective to avoid overfitting. The initial position-independent dimension reduction transformation $\mathbf{S}$ used in the continuous-space models helps to prevent this problem. For the LBL LMs, we find that using the additive continuous representation for

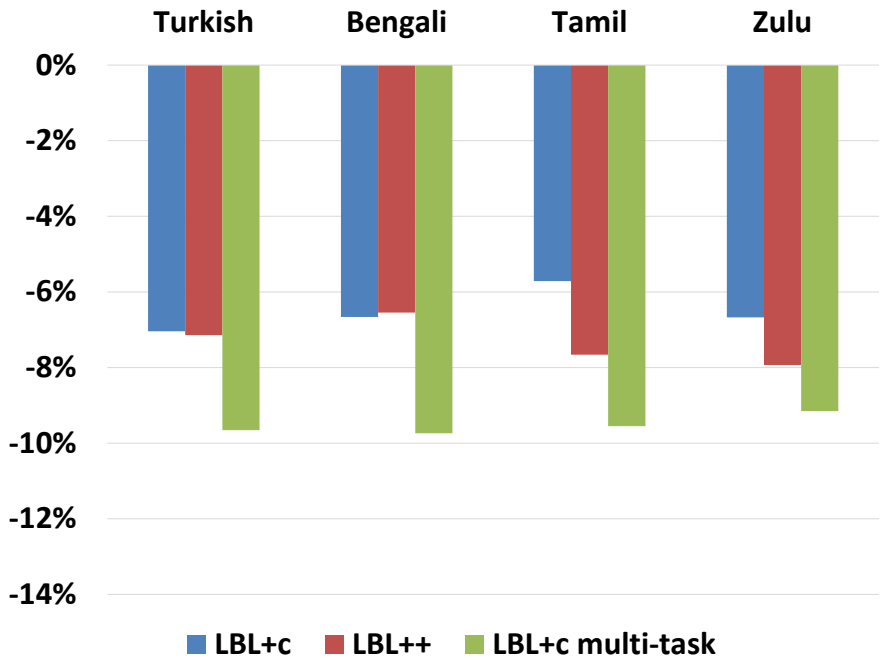


Fig. 7: Relative PPL (on test) reduction of LBL+c LMs and LBL++ LMs over LBL LMs for the LLP condition.

history words helps to reduce the PPL of the LMs by around 6%[3]. For the RNN LMs, the impact varies from 4% to 8%.

Morph Features in the Prediction Space vs. Multi-task Learning

For LBL and RNN LMs, using morphs in the prediction space (X++) does not consistently help under either the LLP condition or the FLP condition, but multi-task training always helps. As shown in Figs. 7–8, both the LBL+c and RNN+c models with multi-task learning outperform their counterparts with prediction word factorization, reducing the PPL by around 7–11% compared to the respective baseline for the LLP condition and 12–18% for the FLP condition.

Best Case Exponential Models

In the figures above, we compared the impact of various uses of morph features to a word-only baseline for the same model structure in order understand what strategies for using morph features are useful in

[3]In [18] the reduction ranges from -2% to 6%. The difference can be caused by the language differences and the low-resource condition.

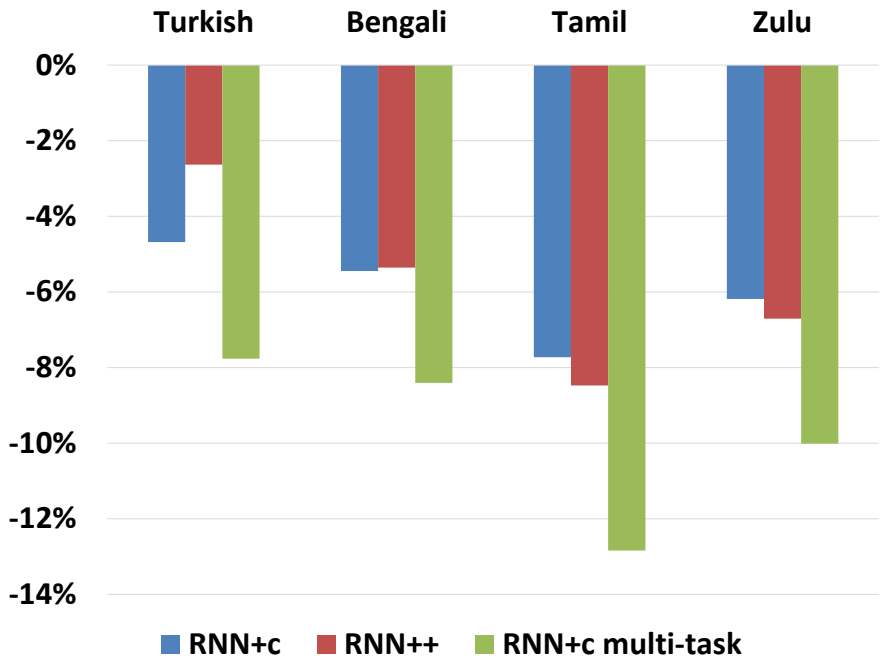


Fig. 8: Relative PPL (on test) reduction of RNN+c LMs and RNN++ LMs over RNN LMs for the LLP condition.

general. However, as presented in Section IV-D, the word-based LBL and RNN models mostly do not beat the mKN baseline. In order to determine which model structures are most useful for the low resource languages explored here, we compare the best configuration for each of the three exponential model forms to the regularized non-hash ME LM baseline. Figs. 9 and 10 show the relative PPL reduction (or increase) of the different models over the mKN trigram LMs for the LLP and FLP conditions, respectively. Most of the time, the ME LMs are slightly better than the mKN trigram LMs, but not as good as their unregularized counterparts. (Recall that for the ME LMs, it is best to *not* use morph features.) The RNN+c LMs with multi-task learning outperform all other models across all four studied languages in terms of PPL in both the LLP and FLP conditions. Although multi-task training gives a substantial boost to the LBL+c configuration, the resulting model only improves over mKN for one language (Bengali) under the LLP condition. As noted earlier, the LBL starts with a baseline PPL that is substantially higher than the mKN baseline, and the improvements due to the use of morph features and multi-task training cannot overcome this disadvantage. Under the FLP condition, the LBL+c with multi-task training improves over mKN baselinse for all languages, though not as significant as RNN+c with multi-task training.

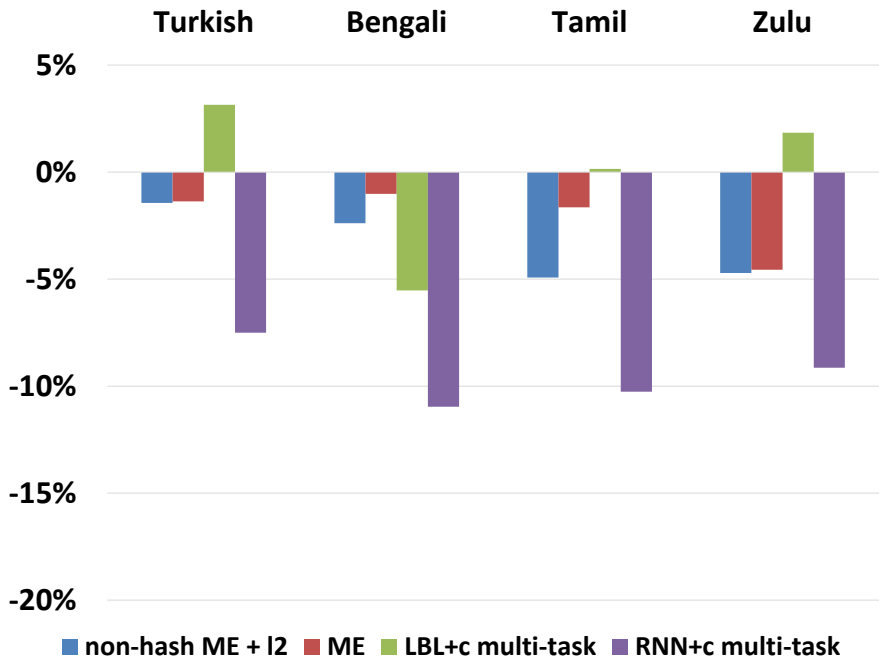


Fig. 9: Relative PPL (on test) reduction of different LMs over mKN tri-gram LMs for the LLP condition.

There are several differences between the LBL and RNN models, including a fixed-length context history for the LBL vs. the recurrent RNN word history, the sigmoid non-linearity in the RNN, and the position-dependent weight matrices (larger number of free parameters for a fixed embedding dimension) in the LBL. The extra parameters associated with the position-dependent matrices are likely driving the optimization of the LBL model to a lower dimension, as discussed further below. The consistently better performance in the FLP condition suggests that the LBL model is better suited to scenarios with large amounts of training data, as suggested by the good results from leveraging morphology in [18]. The use of diagonal transformation matrices or tied parameters may be worth exploring in the LBL for low-resource scenarios.

Impact of Morph Configurations on the Embedding Dimension

The size of the continuous space representation (embedding dimension) is tuned separately for each model and each language; it ranges from 10-90 for the LBL models and 20-100 for the RNN models. The learned dimension is typically larger for the FLP model than the corresponding LLP model, as would be expected with the larger amount of training data. Similarly, it tends to be larger for the models that

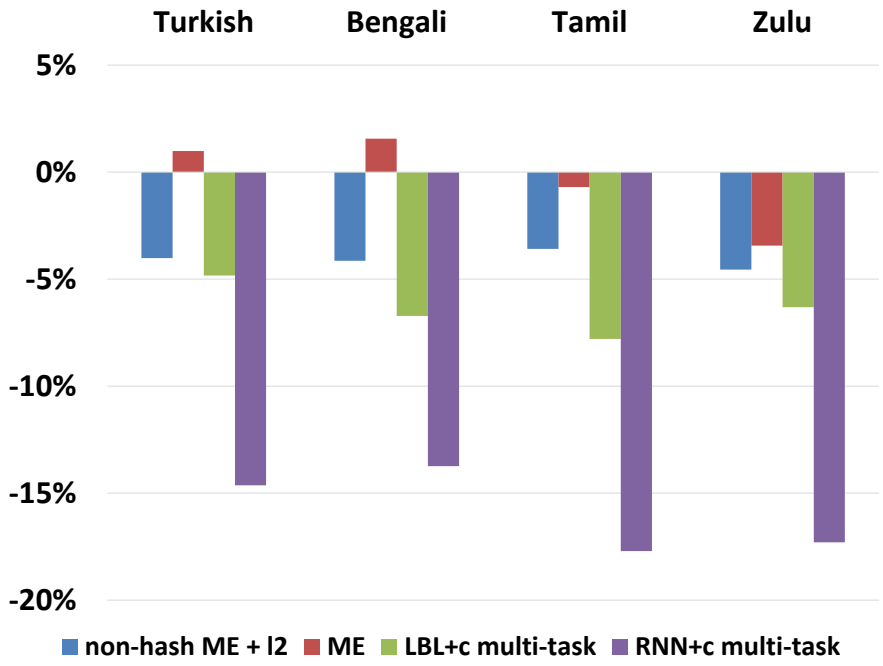
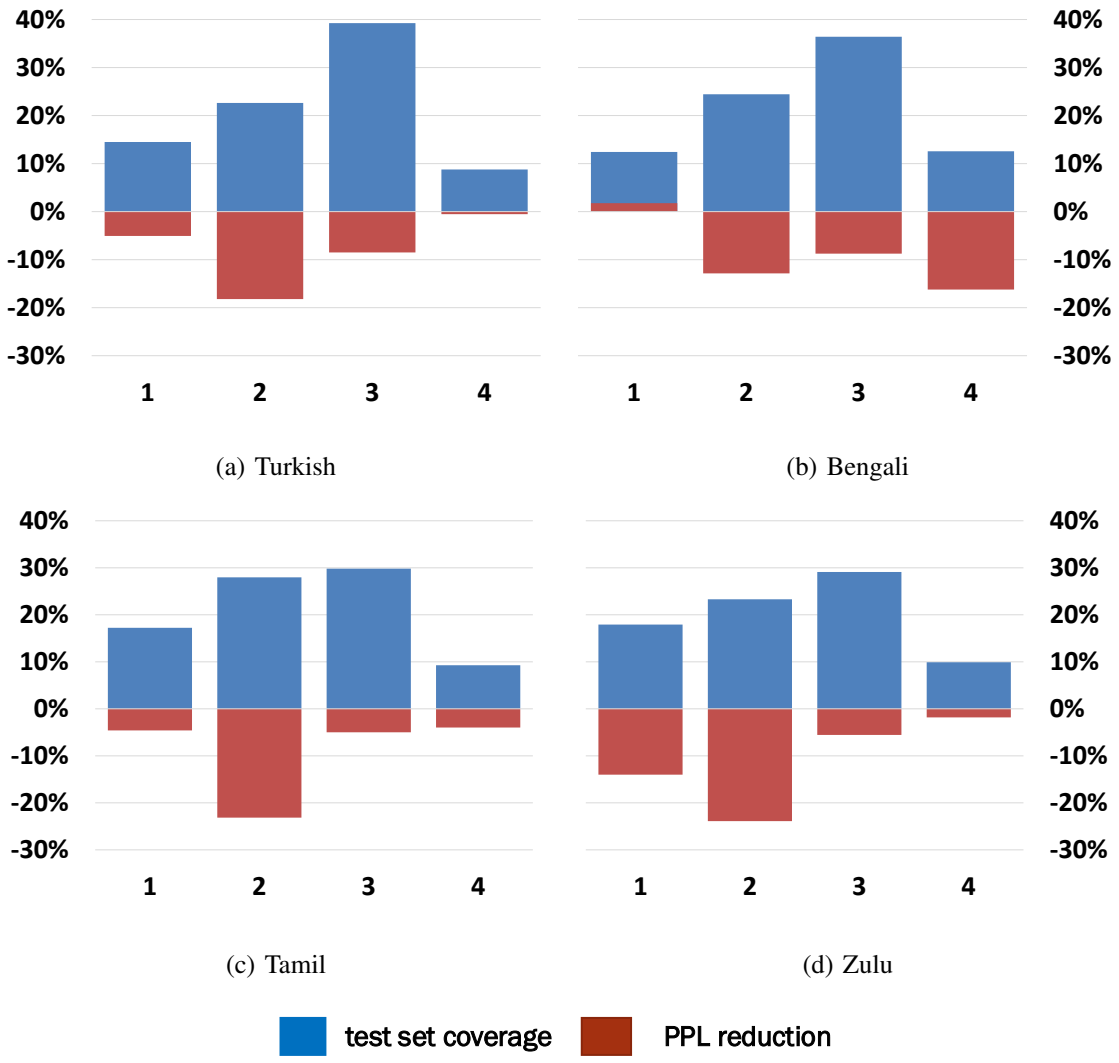


Fig. 10: Relative PPL (on test) reduction of different LMs over mKN tri-gram LMs for the FLP condition.

use morph features and larger for the RNNs than the corresponding LBL model. While these model/data changes are also associated with trends of improvements to PPL, the exceptions to the trends in embedding size do not correspond to exceptions in PPL improvements. In fact, for Turkish, adding morph features in the LLP case is associated with a reduction in the embedding dimension for both the LBL and RNN models, but an improved PPL in both cases. The morph features lead to a more efficient representation. The dimensions tend to be more stable across languages for the models that use morph features: 30-60 for most of the LBL models, and 70-100 for all but one of the RNN models. Neither multi-task training nor adding morph features to the output lead to a substantial or consistent difference in the embedding dimensionality.

PPL Reductions By Token Frequency

Following the PPL reduction analysis done in [18], we partition the tokens in test data into bins according to the token frequency in the training set under the LLP condition. Specially for the best case system, we study the PPL reduction from the RNN LMs to the RNN+c LMs using multi-task training, as shown in Figure 11. PPL reduction is most significant for words that occur less than 100 times but more



(a) Turkish (b) Bengali

(c) Tamil (d) Zulu

Fig. 11: Relative PPL (on test) reduction of RNN+c LMs using multi-task training over RNN LMs. The X-axis is labelled with the number b, which indicatest the number of tokens in test set that occur $c \in [10^{b-1}, 10^b)$ times in the LLP training data.

than 10 times in the LLP training data. An exception is for Bengali, where a large benefit is observed for words that occur more than 1000 times. Since there are only 4–5 words in this category for each language, and frequent words are more likely to be irregular, we are reluctant to attach significance to this exception.

V. CONCLUSION

In summary, this paper has explored strategies for leveraging unsupervised morphological features in exponential language models for limited resource scenarios, including ME, LBL, and RNN models, in comparison to standard discrete (non-parametric) baselines. An important innovation is the use of multi-task training to improve performance in the continuous-space approaches. As observed from our experiments with conversational speech on four limited-resource languages, 7-11% relative reduction in PPL compared to an mKN baseline can be gained from using RNN with morphological features and multi-task learning, with larger gains (12-18%) obtained when training data is increased. Also we observe that, for all languages and both training set sizes, morphology is useful in continuous-space models compared to their word-only baselines, and multi-task learning improves all continuous-space models.

By contrasting these different forms of exponential language models, and including comparisons to a discrete factored LM, we can examine the usefulness of morphs for discrete vs. continuous-space representations of context and in different continuous-space modeling frameworks. In experiments with four high-vocabulary-growth languages, we find that the morph features are much more useful in continuous-space than discrete representations, and that they lead to more stable embedding dimensions across languages. Finally, we find that the RNN model leads to the best performance in this low-resource scenario. With more data, the findings still hold and the resulting benefits can be even greater.

Experiments with the ME model show the importance of regularization for this scenario, and multi-task learning can be seen as a form of regularization. This raises the question of whether additional benefit can be obtained by combining ℓ_2 regularization with morphological features in the ME model or with multi-task learning with the continuous-space models. In exploratory experiments, we found no gain from ℓ_2 regularization in either context, and the training cost is much higher because of the need to run full experiments in tuning the ℓ_2 hyperparameter.

In this work, we have only considered word-based LMs, using unsupervised morphological analysis to provide features that improve learning in limited resource scenarios. In speech recognition under such scenarios, it is useful to have mixed word and subword vocabularies, where subwords may be morphological units, syllables (as in [19]), or other data-driven sublexical units. Any of these subword representations could be incorporated in the framework described here, and the use of multi-task training may be beneficial in learning language models for mixed word-subword systems. Finally, data sparsity can be a problem even with much more training data than was available for our task, either because the language has a high vocabulary growth rate or because higher level features (e.g. supertags) are used to

provide context information. It is possible that gains may be had from multi-task training of word and morph predictors even with much more training data, when these features are augmented by higher level features.

While this paper focused on language modeling, it is worth noting that there is a growing body of work using continuous-space models in a variety of language processing tasks, particularly for deriving semantic representations of words. Of particular relevance here is the method described in [18]. Another related method is described in [44] for word similarity tasks, which uses morphological features as input to a recursive neural network,[4] which provides a continuous-space word representation that can be used as the input to a feedforward neural network. The continuous-space representation developed here with multi-task learning can similarly be used as a semantic representation in other applications.

ACKNOWLEDGEMENTS

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

REFERENCES

[1] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2005, pp. 726–729.

[2] H. Sak, M. Saraclar, and T. Güngör, "Morpholexical and discriminative language models for Turkish automatic speech recognition," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 20, no. 8, pp. 2341–2351, 2012.

[3] T. Hirsimäki, J. Pylkkonen, and M. Kurimo, "Importance of high-order n-gram models in morph-based speech recognition," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 17, no. 4, pp. 724–732, 2009.

[4] E. Arisoy, D. Can, S. Parlak, H. Sak, and M. Saraclar, "Turkish broadcast news transcription and retrieval," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 17, no. 5, pp. 874–883, 2009.

[5] E. Arisoy, M. Saraclar, B. Roark, and I. Shafran, "Syntactic and sub-lexical features for Turkish discriminative language models," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2010, pp. 5538–5541.

[6] A. El-Desoky, C. Gollan, D. Rybach, R. Schluter, and H. Ney, "Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2009, pp. 2679–2682.

[4]Note that a recursive neural network characterizes tree structures, and thus differs from a recurrent neural network. Here we use RNN only for the recurrent case.

[7] A.-D. Mousa, M. Shaik, R. Schluter, and H. Ney, "Sub-lexical language models for German LVCSR," in *Proc. IEEE Spoken Language Technology Workshop*, 2010, pp. 171–176.

[8] M. Shaik, A.-D. Mousa, R. Schluter, and H. Ney, "Using morpheme and syllable based sub-words for Polish LVCSR," in *Proc. Int. Conf. Acoustic, Speech, and Signal Process. (ICASSP)*, 2011, pp. 4680–4683.

[9] M. Shaik, D. Rybach, S. Hahn, R. Schluter, and H. Ney, "Hierarchical hybrid language models for open vocabulary continuous speech recognition using WFST," in *Proc. Workshop on Statistical and Perceptual Audition*, 2012, pp. 46–51.

[10] Y. He, B. Hutchinson, P. Baumann, M. Ostendorf, E. Fosler-Lussier, and J. Pierrehumbert, "Subword-based modeling for handling OOV words in keyword spotting," in *Proc. Int. Conf. Acoustics, Speech and Signal Process.*, May 2014, pp. 7914–7918.

[11] P. F. Brown, V. J. Della Pietra, P. V. de Souza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, pp. 467–479, 1992.

[12] J. Goodman, "A bit of progress in language modeling," *Computer Speech and Language*, vol. 15, no. 4, pp. 403–434, 2001.

[13] E. Whittaker and P. Woodland, "Language modelling for Russian and English using words and classes," *Computer Speech and Language*, vol. 17, no. 1, pp. 87 – 104, 2003.

[14] A.-D. Mousa, R. Schluter, and H. Ney, "Investigations on the use of morpheme level features in language models for Arabic LVCSR," in *Proc. Int. Conf. Acoustics, Speech and Signal Process.*, 2012, pp. 5021–5024.

[15] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[16] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proc. Int. Conf. Machine Learning (ICML)*, 2007, pp. 641–648.

[17] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2010, pp. 1045–1048.

[18] J. A. Botha and P. Blunsom, "Compositional Morphology for Word Representations and Language Modelling," in *Proc. Int. Conf. Machine Learning (ICML)*, 2014.

[19] T. He, X. Xiang, Y. Qian, and K. Yu, "Recurrent neural network language model with structured word embeddings for speech recognition," in *Proc. Int. Conf. Acoustic, Speech, and Signal Process. (ICASSP)*, 2015, pp. 5396–5400.

[20] 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, 2006.

[21] J. A. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Proc. Conf. North American Chapter Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, vol. 2, 2003, pp. 4–6.

[22] B. Roark, M. Saraclar, and M. Collins, "Discriminative n-gram language modeling," *Computer Speech and Language*, vol. 21, no. 2, pp. 373–392, 2007.

[23] S. F. Chen, "Shrinking exponential language models," in *Proc. Conf. North American Chapter Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2009, pp. 468–476.

[24] A. Enami and S. F. Chen, "Multi-class Model M," in *Proc. Int. Conf. Acoustics, Speech and Signal Process.*, 2011, pp. 5516–5519.

[25] B. Hutchinson, M. Ostendorf, and M. Fazel, "A sparse plus low rank maximum entropy language models," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2012.

[26] ——, "A sparse plus low rank maximum entropy language model for limited resource scenarios," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 494–504, 2015.

[27] B. Hutchinson, "Rank and sparsity in language processing," Ph.D. dissertation, Aug 2013.

[28] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[29] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, no. 3, pp. 492–518, 2007.

[30] A. Alexandrescu and K. Kirchhoff, "Factored neural language models," in *Proc. Conf. North American Chapter Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2006.

[31] M. Kang, T. Ng, and L. Nguyen, "Mandarin word-characater hybrid-input neural network language model," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2011.

[32] A.-D. Mousa, H.-K. J. Kuo, L. Mangu, and H. Soltau, "Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic," in *Proc. Int. Conf. Acoustic, Speech, and Signal Process. (ICASSP)*, 2013, pp. 8435–8439.

[33] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Černocký, "Empirical evaluation and combination of advanced language modeling techniques," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2011.

[34] T. Mikolov, I. Sutskever, A. Deoras, H.-S. Le, S. Kombrink, and Černocký Jan, "Subword language modeling with neural networks," unpublished, available at http://www.fit.vutbr.cz/~imikolov/rnnlm/char.pdf, 2012.

[35] Y. Shi, P. Wiggers, and C. M. Jonker, "Towards recurrent neural networks language models with linguistic and contextual features," in *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2012, pp. 1664–1667.

[36] Y. Wu, X. Lu, H. Yamamoto, S. Matsuda, C. Hori, and H. Kashioka, "Factored language model based on recurrent neural network," in *Proc. Int. Conf. Computational Linguistics (COLING)*, 2012.

[37] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in *Proc. Int. Conf. Machine Learning (ICML)*, 2012.

[38] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: Update and outlook," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, 2011.

[39] T. Mikolov, A. Deoras, D. Povey, L.Burget, and J. Cernockyr, "Strategies for training large scale neural network language model," in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011, pp. 196–201.

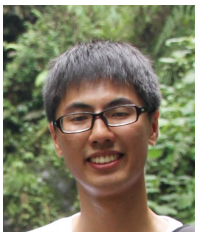
[40] "IARPA babel program," http://www.iarpa.gov/index.php/research-programs/babel, 2011.

[41] M. Creutz and K. Lagus, "Inducing the morphological lexicon of a natural language from unannotated text," in *Proc. Int. and Interdisciplinary Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, June 2005.

[42] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. ICSLP*, 2002, pp. 901–904.

[43] K. Duh and K. Kirchhoff, "Automatic learning of language model structure," in *Proc. Int. Conf. Computational Linguistics (COLING)*, 2004.

[44] M.-T. Luong, R. Socher, and C. Manning, "Better word representations with recursive neural networks for morphology," in *Proc. Conf. Computational Natural Language Learning (CoNLL)*, 2013.



Hao Fang received the B.Eng. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2011 and the M.Sc. degree in electrical and computer engineering from the University of Alberta, Edmonton, Alberta, Canada, in 2013. He is currently working towards his Ph.D. degree at the University of Washington, Seattle, USA. His research interests include natural language processing, deep learning, and signal processing.

Mari Ostendorf (M'85-SM'97-F'05) received a Ph.D. in electrical engineering from Stanford University in 1985. She has worked at BBN Laboratories (1985–1986) and Boston University (1987–1999), and is currently an Endowed Professor of System Design Methodologies in Electrical Engineering at the University of Washington. Her research interests are in dynamic and linguistically-motivated statistical models for speech and language processing. She is a Fellow of IEEE and ISCA, and winner of the 2010 IEEE HP/Rigas Award.

Peter Baumann is currently a Ph.D. student in Linguistics at Northwestern University. He holds degrees in Physics, Linguistics, and Cognitive Science from Northwestern University and University of Freiburg. His research interests include computational linguistics and language processing, in particular computational morphology for language and speech processing.

Janet Pierrehumbert received a Ph.D in Linguistics from MIT in 1980. She was with AT&T Bell Laboratories (1982 – 1989) and Northwestern University (1989-2015), and is current the Professor of Language Modelling in the Oxford e-Research Centre, University of Oxford. Her research uses experimental and computational methods to explore the structure and dynamics of the lexicon in human languages. She is a fellow of the Linguistic Society of America, the Cognitive Science Society, and the American Academy of Arts and Sciences.