Morpholexical and Discriminative Language Models for Turkish Automatic Speech Recognition

Haşim Sak, *Member, IEEE*, Murat Saraçlar, *Member, IEEE*, and Tunga Güngör

***Abstract*—This paper introduces two complementary language modeling approaches for morphologically rich languages aiming to alleviate out-of-vocabulary (OOV) word problem and to exploit morphology as a knowledge source. The first model, *morpholexical language model*, is a generative n-gram model, where modeling units are lexical-grammatical morphemes instead of commonly used words or statistical sub-words. This paper also proposes a novel approach for integrating the morphology into an automatic speech recognition (ASR) system in the finite-state transducer framework as a knowledge source. We accomplish that by building a morpholexical search network obtained by the composition of lexical transducer of a computational lexicon with a morpholexical language model. The second model is a linear reranking model trained discriminatively with a variant of the perceptron algorithm using morpholexical features. This variant of the perceptron algorithm, WER-sensitive perceptron, is shown to perform better for reranking n-best candidates obtained with the generative model. We apply the proposed models in Turkish broadcast news transcription task and give experimental results. The morpholexical model leads to an elegant morphology-integrated search network with unlimited vocabulary. Thus, it is highly effective in alleviating OOV problem and improves the word error rate (WER) over word and statistical sub-word models by 1.8% and 0.4% absolute, respectively. The discriminatively trained morpholexical model further improves the WER of the system by 0.8% absolute.**

***Index Terms*—Automatic speech recognition (ASR), disambiguation, discriminative model, morpholexical language model, morphology, reranking.**

I. Introduction

LANGUAGE modeling for morphologically rich languages such as Arabic, Czech, Finnish, and Turkish has proven to be challenging. The out-of-vocabulary (OOV) rate for a fixed vocabulary size is significantly higher in these languages due to large number of words in language vocabulary. Having a large number of words contributes also to high perplexity numbers for standard n-gram language models due to data sparseness. Language modeling for Turkish as an agglutinative language with a highly productive inflectional and derivational morphology suffers from these problems. We can reduce the OOV rate by increasing the vocabulary size if it is not limited by the size of the text corpus available for ASR systems. However, this also increases the computational and memory requirements of the system. Besides, it may not lead to significant performance improvement due to data sparseness problem of insufficient data for robust estimation of language model parameters.

To overcome the high growth rate of vocabulary and the OOV problem in morphologically rich languages, using grammatical or statistical sub-lexical units for language modeling has been a common approach. The grammatical sub-lexical units can be morphological units such as morphemes or some grouping of them such as stems and endings (grouping of suffixes). The statistical sub-lexical units can be obtained by splitting words using statistical methods. The morphological information is also useful for improving language modeling.

This paper presents a morphology oriented linguistic approach for language modeling in morphologically rich languages as an alternative to word and sub-word based models. This is motivated by the fact that in such languages, grammatical features and functions associated with the syntactic structure of a sentence in morphologically poor languages are often represented in the morphological structure of a word in addition to the syntactic structure. Therefore, morphological parsing of a word may reveal valuable information in its constituent morphemes annotated with morphosyntactic and morphosemantic features to exploit for language modeling.

Standard n-gram language models are difficult to beat if there is enough data. They also lead to efficient dynamic programming algorithms for decoding due to local statistics, and they can be efficiently represented as deterministic weighted finite-state automata [6]. First, this paper proposes a novel approach for language modeling of morphologically rich languages. The proposed model, called the *morpholexical language model*, can be considered as a linguistic sub-lexical n-gram model in contrast to statistical sub-word models.

Second, this paper proposes a novel approach to build a *morphology-integrated search network* for ASR with unlimited vocabulary in the weighted finite-state transducer framework (WFST). The proposed *morpholexical search network* is obtained by the composition of the lexical transducer of the morphological parser and the transducer of morpholexical language model. This model has the advantage of having a

Manuscript received October 11, 2011; revised March 16, 2012; accepted May 07, 2012. Date of publication May 25, 2012; date of current version August 13, 2012. This work was supported in part by the Boğaziçi University Research Fund under Grants 06A102 and 08M103 and in part by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grants 105E102, 107E261, and 109E142. The work of H. Sak was supported by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610 and TÜBİTAK BİDEB 2211. The work of M. Saraçlar was supported by the TUBA-GEBIP award. Parts of this study have been presented in conferences [1], [2], [3], [4], [5].The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Steve Renals.

H. Sak was with the Computer Engineering Department, Boğaziçi University, 34342 İstanbul, Turkey. He is now with Google, Inc., Google Inc., New York, NY 10011 USA (e-mail: hasim@google.com).

M. Saraçlar is with the Electrical and Electronics Engineering Department, Boğaziçi University, 34342 İstanbul, Turkey (e-mail:murat.saraclar@boun.edu.tr).

T. Güngör is with the Computer Engineering Department, Boğaziçi University, 34342 İstanbul, Turkey (e-mail: gungort@boun.edu.tr).

Digital Object Identifier 10.1109/TASL.2012.2201477

dynamic vocabulary in contrast to word models and it only generates valid word forms in contrast to sub-word models.

And finally, we further improve ASR performance by using unigram morpholexical features in a discriminative n-best hypotheses reranking framework with a variant of the perceptron algorithm. The discriminative model is complementary to the generative model and uses the features from the generative model. The perceptron algorithm is tailored for reranking recognition hypotheses by introducing error rate dependent loss function.

The next section gives a summary of previous work and the rest of the paper is organized as follows: In Section III, we describe the language resources that we built for morphological language modeling of Turkish. In Section IV, we present the generative language models that we experimented with. In Section V, we describe the methodology to integrate the morphology into the search network of ASR system. In Section VI, we introduce the discriminative reranking method. In Section VII, we give experimental results and conclude with Section VIII.

II. Related Work

The previous studies on language modeling for morphologically rich languages follow two orthogonal approaches. The first approach uses decomposition of words into sub-lexical units to alleviate the OOV problem and increase the robustness of the language model. The second approach makes use of improved modeling to incorporate other information sources into language modeling.

The decomposition approach can be divided into two classes. The first class of studies uses a linguistically motivated approach, where words are decomposed morphologically into linguistic units. Morpheme-based language models have been proposed for German [7], Czech [8], and Korean [9]. A statistical language model based on morphological decomposition of words into roots and inflectional groups which contain the inflectional features for each derived form has been proposed for morphological disambiguation of Turkish text [10]. Stems and endings have been used for language modeling for Turkish [11]–[13] and Slovenian [14]. Using linguistic information has the advantage that speech recognition output can be processed to filter invalid sequences of morphological units.

The second class of studies on agglutinative languages uses a purely statistical approach to decompose words into sub-word units. Statistical sub-word units so-called morphs have been used for language modeling of Finnish [15], Hungarian [16], and Turkish [13]. Sub-word language models are effective in alleviating the OOV problem, and they have the advantage that they do not require any language specific linguistic processing, which can be costly to build for all languages. However, the speech decoder can generate ungrammatical sub-word sequences and postprocessing of the sub-word lattices may be required to correct the errors and increase the accuracy using linguistic information [11], [17].

In addition to the decomposition approach, the language models can be extended to use morphological information. For instance, a morphology-based language modeling approach, *Factored Language Models (FLMs)* have been shown to reduce language model perplexity and lead to WER reductions in Arabic speech recognition systems [18]. FLMs decompose words into a set of features (or factors) and estimate a language model over these factors, smoothed with *generalized parallel backoff* mechanism which improves the robustness of probability estimates for rarely observed n-grams. We previously experimented with FLMs for Turkish [4] and observed that FLMs are effective in reducing perplexity of language models but only when the training data is limited. The computational cost and the inability to be represented efficiently and compactly as finite-state models also limit their usefulness. Morphological information has also been employed later in the system as in [19], where a maximum entropy model has been trained with morphological and lexical features to rescore n-best hypotheses for Arabic speech recognition and machine translation.

TABLE I
Statistics for the NewsCor Corpus

# of word tokens	182,622,247
OOV rate of the parser (word token)	1.3
# of word types	1,819,157
OOV rate of the parser (word type)	38.8
average # of parses per word type	2.4
average # of morphemes per word type	3.7
root with max # of parses (3545)	çık[Verb]
word with max # of morphemes (9)	ruhsatlandırılamamasındaki

III. Language Resources

We have built and compiled some language resources and tools for morphological processing of Turkish. These resources were presented in a previous work [20], and we describe them here briefly for completeness and introducing the methods for extraction and representation of morphological information. The resources and tools[1] are composed of a morphological parser, a morphological disambiguator, and a text corpus.

A. Text Corpus

Statistical language models require large text corpora to train accurate models. Productive morphology, morphological parsing ambiguity, and free word order characteristics of a language all make this requirement more pronounced. Due to the lack of such a large text corpus for Turkish, we compiled a text corpus by crawling and sampling from Turkish web pages [20]. For this research, we use the *NewsCor* corpus which contains news articles from three major news portals, since we do the experiments on a broadcast news transcription task. The statistics for the number of tokens (words and lexical units such as punctuation marks), types (distinct tokens), and morphological parsing are shown in Table I.

B. Finite-State Morphological Parser

The extraction of morphological information hidden in the structure of words calls for morphological parsing, which is the

[1]Available at http://busim.ee.boun.edu.tr/~speech/langres.html.

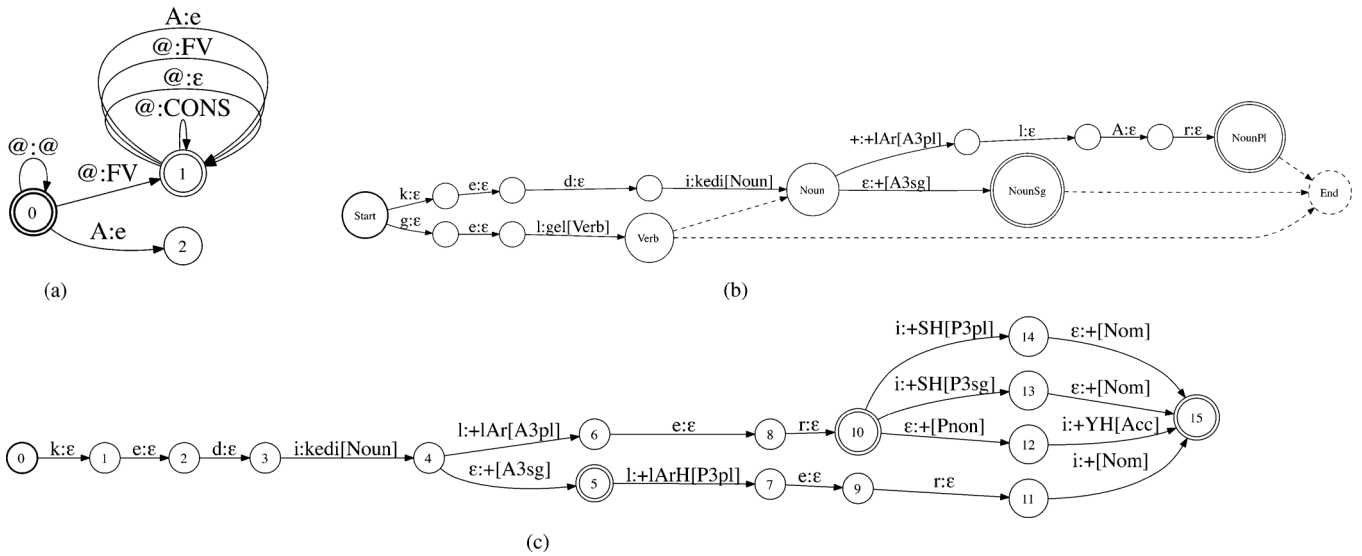

Fig. 1. (a) Example transducer encoding Turkish vowel harmony rule: "@" symbol represents any feasible lexical or surface symbol absent in the . (b) Example transducer for Turkish nominal inflection. (c) Part of the lexical transducer of the morphological parser showing ambiguous parses for the word *kedileri*. (a) Morphographemics, (b) Lexicon and Morphotactics, (c) Lexical transducer.

decomposition of words into constituent morphemes and associated morphosyntactic and morphosemantic features. Finite-state machines offer an elegant and unified framework for modeling and computation in language and speech processing [21]. They have also enough power to model morphological phenomena in most languages including Turkish. This is especially important for enabling seamless integration of morphology as a knowledge source with other finite-state models by finite-state operations like composition.

The two-level morphology formalism of Koskenniemi [22] provides the mechanism to describe the phonological alternations in a two-level rule specification. These two-level rule definitions can be compiled into finite-state transducers. The following rule is an example for vowel harmony phenomena in Turkish which forces change of vowels in surface form of suffixes to agree in backness with the preceding vowel:

$$\texttt{A:e} \Rightarrow \texttt{@:FV}\,[\,\texttt{:CONS} \mid \texttt{:0}\,]* \_.$$

This rule states that symbol "A" in lexical level may be converted to /e/ vowel only if it is preceded with a surface front vowel followed possibly by a number of symbols having consonant or epsilon realizations in the surface level. Finite-state transducer implementation of this rule in a compact form can be seen in Fig. 1(a). The compilation and intersection of all the rule transducers as finite-state automata is a morphographemics transducer.

The morphotactics which encodes the morphosyntax—the ordering of morphemes—can also be represented as a finite-state machine. Fig. 1(b) shows a small part of the lexicon and morphotactics for Turkish represented as a finite-state transducer. We refer to this transducer as the morphotactics transducer. The finite-state transducer of the morphological parser is obtained as the composition of the morphographemics transducer and the morphotactics transducer. Fig. 1(c) shows the part of this lexical transducer corresponding to all the parses of the ambiguous word *kedileri*. The two-level phonological rules and the morphotactics were adapted from the PC-KIMMO implementation of Oflazer [23]. The rules were compiled using the *twolc* rule compiler [24]. A new root lexicon of 89 484 words based on the Turkish Language Institution dictionary[2] and the analysis of NewsCor corpus was compiled. For finite-state operations, we use the AT&T FSM tools [21] and the OpenFST weighted finite-state transducer library [25].

The morphological feature representation is similar to the one used in [26]. Each output of the morphotactics begins with the root word and its part-of-speech tag in brackets. These are followed by a set of lexical morphemes associated with morphological features (nominal features such as case, person, and number agreement; verbal features such as tense, aspect, modality, and voice information). The inflectional morphemes start with a + sign. The derivational morphemes start with a − sign and the first feature of a derivational morpheme is the part-of-speech of the derived word form. An example morphological analysis for the word *ölümsüzleştirilebileceğini* is shown below:

ölüm[Noun]+[A3sg]+[Pnon]+[Nom]−sHz[Adj+Without]
−lAş[Verb+Become]−DHr[Verb+Caus]−Hl[Verb+Pass]
−$YAbil$[Verb+Able]+[Pos]−$YAcAk$[Noun+FutPart]+[A3sg]
+SH[P3sg]+NH[Acc]

This word can be translated as "... *that s/he can be immortalized"*.

The (inverse of the) morphological parser can generate infinite number of words, due to iteration of some suffixes such as causation and noun-verb-noun cycles in morphotactics of the language. However, the words having more than six morphemes are rarely used in practice with a frequency of about 0.7%. The statistics for the morphological analysis of the NewsCor corpus is given in Table I. The parser is highly efficient and can analyze about 8700 words per second on a 2.33-GHz Intel Xeon processor.

[2]http://www.tdk.gov.tr

TABLE II
STATISTICAL AND GRAMMATICAL WORD SPLITTING APPROACHES ON AN EXAMPLE SENTENCE WITH THE GLOSS "*HELLO YOU ARE GETTING THE NEWS FROM THE AGENCY*"

Model	Decomposition	Level	Example
word	grammatical	surface	merhaba haberleri ajanstan alıyorsunuz
morpheme (lexical)	grammatical	lexical	merhaba[Noun]+[A3sg]+[Pnon]+[Nom] haber[Noun] +lAr[A3pl] +SH[P3sg]+[Nom] ajans[Noun]+[A3sg]+[Pnon] +DAn[Abl] al[Verb]+[Pos] +Hyor[Prog1] +sHnHz[A2pl]
stem+ending (lexical)	grammatical	lexical	merhaba[Noun]+[A3sg]+[Pnon]+[Nom] haber[Noun] +lAr[A3pl]+SH[P3sg]+[Nom] ajans[Noun]+[A3sg]+[Pnon] +DAn[Abl] al[Verb]+[Pos] +Hyor[Prog1]+sHnHz[A2pl]
stem+ending (surface)	grammatical	surface	merhaba haber +leri ajans +tan al +ıyorsunuz
morph	statistical	surface	merhaba haber +ler +i ajans +tan al +ıyor +sun +uz

TABLE III
STATISTICS FOR LANGUAGE MODELING UNITS OVER A BROADCAST NEWS CORPUS OF 1.3 MILLION WORDS

Model	units/word	tokens	types
word	1.00	1,342,597	106,789
morpheme (lexical)	1.90	2,555,427	42,057
stem+ending (surface)	1.46	1,954,665	40,182
morph	1.41	1,890,774	28,139
stem+ending (lexical)	1.52	2,046,437	46,118

C. Morphological Disambiguator

The morphological parser may return more than one possible analysis for a word due to ambiguity. For example, the parser outputs four different analyses for the word *kedileri* as shown below. The English glosses are given in parentheses.

kedi[Noun]+lAr[A3pl]+SH[P3sg]+[Nom] (his/her cats)
kedi[Noun]+lAr[A3pl]+[Pnon]+YH[Acc] (the cats)
kedi[Noun]+lAr[A3pl]+SH[P3pl]+[Nom] (their cats)
kedi[Noun]+[A3sg]+$lArH$[P3pl]+[Nom] (their cat)

This parsing ambiguity needs to be resolved for further language processing such as for language modeling using a morphological disambiguator (morphosyntactic tagger). The averaged perceptron algorithm previously applied to classification problems [27] has also been adapted very successfully to natural language processing (NLP) tasks such as syntactic parsing of English text [28] and part-of-speech tagging and noun phrase chunking [29]. This methodology also proved to be quite successful for morphological disambiguation of Turkish text [1], [2]. The disambiguation system achieves about 97.05% disambiguation accuracy on the test set.

IV. GENERATIVE LANGUAGE MODELS

In the following sections, we describe the word, sub-word and morpholexical language models. The corresponding statistical and grammatical splitting approaches are shown for an example sentence in Table II and the unit statistics of average number of units per word, number of unit tokens and types are given in Table III for all the models over the broadcast news corpus of 1.3 million words, which is used as in-domain data in the experiments.

A. Word and Statistical Sub-Word Language Models

The conventional approach for language modeling is estimating a statistical n-gram language model over a fixed vocabulary of words. As a baseline word language model, we built 200 K vocabulary *3*-gram language model which is also used as a baseline in [13].

For unlimited vocabulary speech recognition, splitting words into morphs (morpheme-like sub-words) using an unsupervised algorithm based on minimum description length principle has been very effective by alleviating OOV problem and reducing language model perplexity [15]. The baseline Morfessor algorithm introduced in [30] is used to segment word types in the text corpus. We used the best performing segmentations of the study in [13]. Statistical morphs have the advantage that no linguistic knowledge is required about the language. On the other hand, since morphs do not generally correspond to grammatical morphemes, we cannot easily employ linguistic information in later stages of processing such as rescoring sub-word lattices. Moreover, speech decoder can generate ungrammatical sub-word sequences and postprocessing of the sub-word lattices are required to correct the errors and increase the accuracy [11], [17].

B. Morpholexical Language Models

In this section, we introduce a linguistic approach to exploit morphology and alleviate OOV problem in language modeling. This can be considered as a grammatical sub-lexical language modeling approach. The modeling units are lexical and grammatical morphemes annotated with morphosyntactic and morphosemantic features. This is motivated by the fact that lexical and grammatical morphemes (*morpholexical units*) constitute natural sub-lexical units of a morphologically complex language. For instance, the constituent morphemes are generally the output symbols of a morphological parser when represented as finite-state models.

The morpholexical language modeling can be considered as replacing a static lexicon of words or sub-words with a dynamic computational lexicon. The dynamic lexicon over grammatical and lexical morphemes greatly solves the OOV problem by providing a root lexicon with a good coverage and makes it unnecessary to list all word forms that can be generated from a root word, which may not be even possible for languages like Turkish. For instance, the OOV rate of the morphological parser is 1.3% on the test set. This model also provides better probability estimates for rarely seen or unseen word n-grams by morphological decomposition of words.

We can train the morpholexical language models as standard n-gram language models over morpholexical units. For this, we need to parse a text corpus to get the morpholexical units using a morphological parser. Since the morphological parser can give multiple analyses due to morphological ambiguity, we need to

use a morphological disambiguator to choose the correct parse of the words using the contextual information. We then split the morphological analyses of words at morpheme boundaries and use standard n-gram estimation methods to train a language model over morpholexical units.

We also experimented with combining grammatical morphemes to build a lexical stem+ending model to alleviate the problem of large number of morphemes preventing n-grams to have a proper coverage of context. Using lexical units rather than surface forms as in statistical morphs is also beneficial in terms of decreasing data sparsity since a lexical morpheme may be realized in multiple surface forms due to phonological alternations [12]. Such an example for Turkish is the lexical plural morpheme $+lAr$ which can have the surface form of *ler* or *lar* depending on the previous vowel this morpheme is suffixed. In this study, we use the lexical stem+ending decompositions to obtain the surface form stem+ending decompositions of words which can be considered as grammatical sub-words in contrast to statistical sub-words. The different modeling units for morpholexical language models can be seen in Table II.

The morpholexical language models have the advantage that when combined with the lexical transducer of the morphological parser, they give probability estimates for only valid word sequences. This is not possible with statistical sub-word model or surface form stem+ending model, but this is possible with morpholexical language models since the morphotactics effectively constrains the language model over valid morpheme sequences. In this paper, we show the effect of morphotactics in language modeling by giving experimental results where we relax the morphotactics to allow any morpheme sequences in the lexical transducer. We also study the effect of morphological disambiguation in language modeling by comparing the proper morphological disambiguation of training corpus and choosing the morphological parse with the least number of morphemes.

V. Morpholexical Search Network for ASR

In this section, we explain how a morpholexical language model can be integrated into speech recognition in the finite-state transducer framework.

The weighted finite-state transducers (WFSTs) provide a unified framework for representing different knowledge sources in ASR systems [31]. In this framework, the speech recognition problem is treated as a transduction from input speech signal to a word sequence. A typical set of knowledge sources consists of a hidden Markov model H mapping HMM state ID sequences to context-dependent phones, a context-dependency network C transducing context-dependent phones to context-independent phones, a lexicon L mapping context-independent phone sequences to words, and a language model G assigning probabilities to word sequences. The composition of these models $H \circ C \circ L \circ G$ results in an all-in-one search network that directly maps HMM state ID sequences to weighted word sequences.

The morphology as another knowledge source can be represented as a WFST and can be integrated into the WFST framework of an ASR system. The lexical transducer of the morphological parser maps the letter sequences to lexical and grammatical morphemes annotated with morphological features. The lexical transducer can be considered as a computational dynamic lexicon in ASR in contrast to a static lexicon. The computational lexicon has some advantages over a fixed-size word lexicon. It can generate many more words using a relatively smaller number of root words in its lexicon. So it achieves lower OOV rates. Different than the static lexicon, even if we have never seen a specific word in the training corpus, the speech decoder has the chance to recognize that word. Another benefit of the computational lexicon is that it outputs the morphological analysis of the word generated. We can exploit this morphological information in a language model.

Since most of the words in Turkish have almost one-to-one mapping between graphemics and pronunciation, we use the Turkish letters as our phone set in Turkish ASR.[3] In the WFST framework, the lexical transducer of the morphological parser can be considered as a computational lexicon M replacing the static lexicon L. The transducer M outputs some symbols representing morphological features not corresponding to any lexical form in addition to lexical and grammatical morphemes. The morpholexical language model is estimated over some combination of these features and morphemes. Therefore, we need an intermediate transducer T to do the symbol mapping between these models. Then the search network with the morpholexical language G_{mlex} model can be built as $H \circ C \circ M \circ T \circ G_{\mathrm{mlex}}$.

The WFST offers finite-state operations such as *composition*, *determinization* and *minimization* to combine all the knowledge sources used in speech recognition and optimize into a single compact search network [32]. This approach works well for certain types of transducers, but presents some problems related to the applicability of *determinization* and *weight-pushing* with more general transducers [33]. In this respect, Turkish morphology presents a problem, since the number of ambiguities is infinite and the cycle-ambiguous finite-state transducer of the morphological parser is not determinizable. Still, we can apply the local determinization algorithm for locally optimizing the search network using the *grmlocaldeterminize* utility from *AT&T Grammar Library* [34]. The experimental results show that this approach works well.

VI. Discriminative Reranking With Perceptron

The introduction of arbitrary and global features into the generative models results in difficulty due to the finite-state nature of these models. Therefore, the common approach in NLP research has been to use a baseline generative model to generate ranked n-best candidates, which are then reranked by a rich set of local and global features [35], [36].

The perceptron algorithm has been successfully applied to various NLP tasks for ranking or reranking hypotheses [27]–[29], [35], [36]. The perceptron has shown significant improvements for discriminative language modeling for Turkish using linguistic and statistically derived features [37]. It also gives the best performance for morphological disambiguation of Turkish text using morpholexical features [1]. The characteristics like simplicity, fast convergence, and easy

[3]We built a finite-state transducer based pronunciation lexicon similar to [26] and extended the phone set, however it did not lead to performance improvement possibly due to a small number of Turkish words with exceptional pronunciation.

input set of training examples $\{(x_i, y_i) : 1 \leq i \leq n\}$
input number of iterations T
$\bar{\alpha} = 0$, $\bar{\alpha}_{sum} = 0$
for $t = 1 \ldots T$, $i = 1 \ldots n$ **do**
  $z_i = \text{argmax}_{z \in \mathbf{GEN}(x_i)} \mathbf{\Phi}(x_i, z) \cdot \bar{\alpha}$
  $\bar{\alpha} = \bar{\alpha} + \Delta\,(y_i, z_i)(\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i))$
  $\bar{\alpha}_{sum} = \bar{\alpha}_{sum} + \bar{\alpha}$
end for
return $\bar{\alpha}_{avg} = \bar{\alpha}_{sum}/(nT)$

Fig. 2. WER-sensitive perceptron algorithm.

incorporation of arbitrary local and global features make the perceptron algorithm very attractive for discriminative training of linear models. In this section, we introduce a variant of the perceptron, WER-sensitive perceptron, which is better suited to rerank n-best speech recognition hypotheses.

A. Perceptron Algorithm

The perceptron is a linear classifier [38]. The perceptron algorithm tries to learn a weight vector that minimizes the number of misclassifications. Fig. 2 shows a variant of the perceptron algorithm, *WER-sensitive perceptron*, formulated as a multiclass classifier which is very similar to the the averaged perceptron [27], [29]. The algorithm estimates a parameter vector $\bar{\alpha} \in \Re^d$ using a set of training examples $(x_i, y_i) : 1 \leq i \leq n$. The function $\mathbf{GEN}$ enumerates a finite set of candidates $\mathbf{GEN}(x) \subseteq Y$ for each possible input x. The representation $\mathbf{\Phi}$ maps each $(x, y) \in X \times Y$ to a feature vector $\mathbf{\Phi}(x, y) \in \Re^d$. The learned parameter vector $\bar{\alpha}$ can be used for mapping unseen inputs $x \in X$ to outputs $y \in Y$ by searching for the best scoring output, i.e., $\arg\max_{z \in \mathbf{GEN}(x)} \mathbf{\Phi}(x, z) \cdot \bar{\alpha}$. The given algorithm can also be used to rank the possible outputs for an input x by their scores, $\mathbf{\Phi}(x, z) \cdot \bar{\alpha}$.

The algorithm makes multiple passes (denoted by T) over the training examples. For each example, it finds the highest scoring candidate among all candidates using the current parameter values. If the highest scoring candidate is not the correct one, it updates the parameter vector $\bar{\alpha}$ by the difference of the feature vector representation of the correct candidate and the highest scoring candidate. This way of parameter update increases the parameter values for features in the correct candidate and downweights the parameter values for features in the competitor. For the application of the model to the test examples, the algorithm calculates the "averaged parameters" since they are more robust to noisy or inseparable data [29]. The averaged parameters $\bar{\alpha}_{\text{avg}}$ are calculated by summing the parameter values for each feature after each training example and dividing this sum by the total number of updates. We define X, Y, x_i, y_i, $\mathbf{GEN}$, and $\mathbf{\Phi}$ of the perceptron algorithm in a reranking setting of ASR hypotheses as follows:

- X is the set of all possible acoustic inputs.
- Y is the set of all possible strings, $\sum^*$, for a vocabulary $\sum$ which can be a set of words, sub-words, or morpholexical units of the generative language model.
- Each x_i is an utterance—a sequence of acoustic feature vectors. The training set contains n such utterances.
- $\mathbf{GEN}(x_i)$ is the set of alternate transcriptions of x_i as output from the speech decoder. Although the speech decoders can generate lattices which encode alternate recognition results compactly, we prefer to work on n-best lists for efficiency reasons and very small performance gains with the lattices.
- y_i is the member of the $\mathbf{GEN}(x_i)$ with the lowest word error rate with respect to the reference transcription of x_i. Since there can be multiple transcriptions with the lowest error rate, we take y_i to be the one with the best score among them.
- Each component $\Phi_j(x, y)$ of the feature vector representation $\mathbf{\Phi}(x, y) \in \Re^d$ holds the number of occurrences of a feature or indicates the existence of a feature. For instance one of the features can be defined on part of speech tags of the words as follows:
  $\Phi_1(x, y) =$ number of times an *adjective* is followed by a *noun* in y.
- The expression $\mathbf{\Phi}(x, y) \cdot \bar{\alpha}$ denotes the inner product $\sum_{j=1}^{d} \Phi_j(x, y)\alpha_j$, where α_j is the jth component of the parameter vector $\bar{\alpha}$.
- The zeroth component $\Phi_0(x, y)$ represents the log-probability of y (weighted sum of the baseline language and acoustic model scores) in the lattice output from the baseline recognizer for utterance x. We experimented with the perceptron algorithm where this baseline score can be included or omitted in training. During testing, the baseline score as the zeroth feature is always included. The corresponding weight α_0 for $\Phi_0(x, y)$ is fixed and optimized on a held-out set.

With this setting, the perceptron algorithm learns an averaged parameter vector $\bar{\alpha}_{\text{avg}}$ that can be used to choose the transcription y having hopefully the least number of errors for an utterance x using the following function:

$$F(x) = \underset{y \in \mathbf{GEN}(x)}{\arg\max}\ \mathbf{\Phi}(x, y) \cdot \bar{\alpha}_{\text{avg}}.$$

The WER-sensitive perceptron algorithm is obtained by defining a better loss function tailored for reranking ASR hypotheses. The loss function of the averaged perceptron [29] algorithm can be written as follows:

$$L(\bar{\alpha}) = \sum_{i=1}^{n} [\bar{\alpha} \cdot \mathbf{\Phi}(x_i, z_i) - \bar{\alpha} \cdot \mathbf{\Phi}(x_i, y_i)]$$

where $[x] = 0$ if $x < 0$ and 1 otherwise. We can define a better loss function which is based on the total number of extra errors we do by selecting the candidates with higher WER rather than the best candidates. Then minimizing the loss function corresponds to minimizing the WER of the reranker. We define the word error rate sensitive loss function as follows:

$$L(\bar{\alpha}) = \sum_{i=1}^{n} \Delta(y_i, z_i)[\bar{\alpha} \cdot \mathbf{\Phi}(x_i, z_i) - \bar{\alpha} \cdot \mathbf{\Phi}(x_i, y_i)]$$

where the loss function $\Delta(y_i, z_i)$ for each example x_i is defined as the difference of edit distances of z_i and y_i with the reference transcription of x_i.

The gradient of the loss function is $\Delta(y_i, z_i)(\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i))$ which yields a simple modification to the perceptron update rule. We provide a proof of convergence for the WER-

sensitive perceptron algorithm for linearly separable training sequences in the Appendix.

Note that a loss-sensitive perceptron algorithm has been proposed for reranking speech recognition output in [39]. Although this work is similar in using edit distance as a loss function, they use it for scaling the margin to ensure that hypotheses with a large number of errors are more strongly separated from the members of the set of lowest error (optimal) hypotheses. They also update the weight vector using features from optimal and non-optimal set of hypotheses that violate the scaled margin.

VII. Experiments

This section gives experimental results for the application of proposed generative and discriminative language models to a Turkish broadcast news transcription task.

A. Broadcast News Transcription System

The automatic transcription system uses hidden Markov models (HMMs) for acoustic modeling and WFSTs for model representation and decoding. The HMMs are decision-tree state clustered cross-word triphone models with 10 843 HMM states and each state is a Gaussian mixture model (GMM) having 11 mixture Gaussian densities with the exception of silence model having 23 mixtures. The model has been trained on 188 hours of acoustic data from the Boğaziçi broadcast news (BN) database [13], [40]. Separate from the training data, disjoint held-out (3.1 hours) and test (3.3 hours) data sets are used for parameter optimization and final performance evaluation, respectively.

The language models are trained using two text corpora. The larger corpus is the NewsCor corpus (184 million words) described in Section III-A and acts as a generic corpus collected from news portals. The other one is the BN corpus (1.3 million words) and it contains the reference transcriptions of BN database and acts as in-domain data. The generative language models are built by linearly interpolating the language models trained on these corpora. The interpolation constant is chosen to optimize the perplexity of held-out transcriptions. The baseline n-gram language models are estimated with interpolated Kneser–Ney smoothing and entropy-based pruning using the SRILM toolkit [41]. It was observed that aggressive entropy pruning of Kneser–Ney models leads to severe degradation in modeling accuracy [42]. However, our model sizes are relatively small and we employ mild pruning. The discriminative models are trained using only the BN corpus. The speech recognition experiments are performed by using the AT&T DCD library. This library is also used for the composition and optimization of the finite-state models to build the search network for decoding.

B. Generative Language Models

We evaluate the performance of the proposed morpholexical language model against the word and morph models on the broadcast news transcription task. We experiment with two different morpholexical language models with modeling units of lexical-grammatical morpheme and lexical stem+ending.

TABLE IV
Size of Finite-State Models

Model	states	arcs
word	8,717,593	16,386,105
stem+ending (surface)	8,555,347	17,459,319
morpheme (lexical)	10,768,226	24,266,655
morph	8,862,922	22,390,876
stem+ending (lexical)	7,115,253	19,783,696

All model parameter settings including the vocabulary size and n-gram order are optimized for each model individually to get the best recognition accuracy given the memory limit of 64 GB during the construction of the static decoder network. For the experimental setup of the word and morph based models, we use the same settings as in the previous studies of [13], [40]. For the word based model, the vocabulary size of 200 K and n-gram order of 3 are chosen. The OOV rate of the test set with the 200 K word vocabulary is 2%. For the morph based model, we employ the best performing method of marking non-initial morphs with "$-$", which is used to locate the word boundaries for the purpose of conversion from morph sequences of recognition results to word sequences. This method increases the vocabulary size from 50 K to 76 K. The OOV rate of the test set with the 76 K morph vocabulary is 0%, since the letters are also included in the morph lexicon. The morph based experiments are conducted with 4-gram language models. To build the morpholexical language models, the text corpora are morphologically parsed and disambiguated to get the lexical-grammatical morpheme and lexical stem+ending representations of corpora. The lexicon of the morphological parser contains about 88 K symbols. The OOV rate of the morphological parser on the test set is about 1.3%. The lexical-grammatical morpheme representation results in about 175 K symbols. The lexical stem+ending representation yields about 200 K symbols. For both morpholexical units, the n-gram order of 4 is chosen. The morpholexical search networks are built using the lexical transducer of the morphological parser and the weighted finite-state automata representation of the morpholexical language models as explained in Section V. The search network is optimized using local determinization from GRM library [34]. The size of search network is given for all the models in Table IV.

Fig. 3 shows the word error rate versus real-time factor for the word, morph, morpheme, lexical stem+ending, and surface form stem+ending models for the first-pass. Note that since the morpholexical models output recognition results in morphological representation, we use the inverse of lexical transducer of morphological parser as a word generator to convert them to words to calculate the WERs. Since the language models are pruned for computational reasons to about one tenth of their size when building the optimized search networks of first-pass recognition, we rescore the output lattices with unpruned language models as a second-pass. Table V shows the second-pass and the 1000-best oracle WERs for all the models with the largest beam width in Fig. 3. As can be seen from the second-pass results, the sub-word and sub-lexical models perform significantly better than the word-based model. This is mostly due to the reduction in the OOV rate. As a

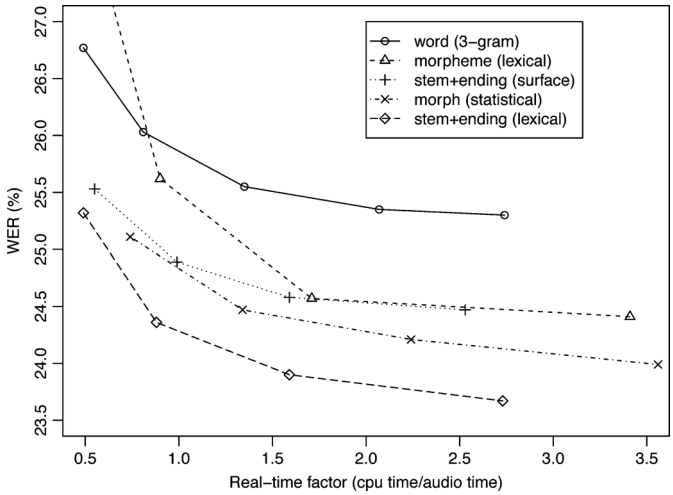

Fig. 3. Word error rate for the first-pass versus real-time factor obtained by changing the pruning beam width.

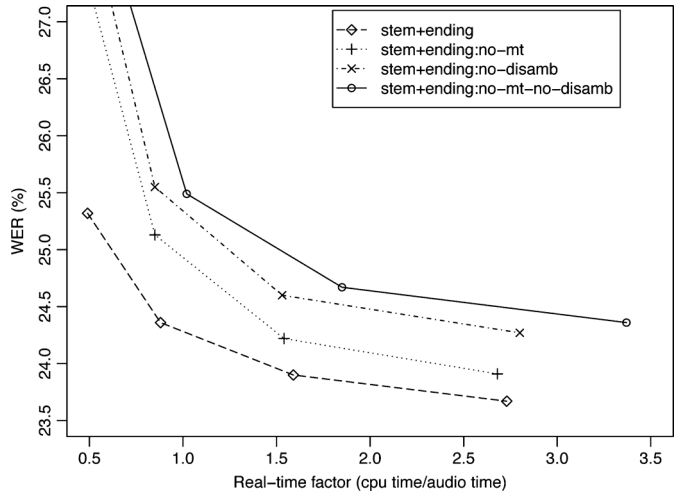

Fig. 4. Effects of morphotactics and morphological disambiguation for the lexical stem+ending model.

TABLE V
RESULTS FOR TWO-PASS RECOGNITION SETUP

Model	Oracle WER	Second-pass WER
word	10.2	23.1
stem+ending (surface)	9.6	21.9
morpheme (lexical)	12.1	21.8
morph	9.7	21.7
stem+ending (lexical)	10.5	21.3

morpholexical language model, the lexical stem+ending model has the best performance. This is partly due to the correction of invalid words. An analysis shows that out of 23 303 words in the test set, the surface stem+ending model yields 90 invalid words (0.4%) and the morph model yields 86 invalid words. We also did WER analysis on the stems by parsing the recognition results and removing the suffixes leaving only the stems. The stem error rates for word, surface stem+ending, morpheme, morph and lexical stem+ending models are 20.7, 20.1, 19.9, 19.8, and 19.5, respectively. These results are consistent with the word error rates. This analysis confirms that the lexical stem+ending model also improves the recognition of stems.

C. Effectiveness of Morphotactics and Morphological Disambiguation

In this section, we give experimental results showing the effect of morphotactics and morphological disambiguation on speech recognition performance using the lexical stem+ending model. Fig. 4 shows the word error rate of the first-pass speech recognition at various real-time factors using four different language models. The baseline model is the lexical stem+ending model with the correct morphotactics and morphological disambiguation (97.05% disambiguation accuracy on the disambiguation test set). First, we experimented with the morphotactics. The *stem+ending:no-mt* model represents the experiment where the morphotactics component of the lexical transducer allows any ordering of the morphemes. Second, we tested the effectiveness of doing morphological disambiguation on the language model text corpus. The *stem+ending:no-disamb* model represents the case where the morphological disambiguation is replaced with choosing the morphological parse with the least number of morphemes (89.93% disambiguation accuracy on the disambiguation test set). The final model *stem+ending:no-mt-no-disamb* shows the cumulative effect for the absence of morphotactics and morphological disambiguation. It is clear that morphotactics is effective in reducing the error rate. This result shows that morphotactics is successful in constraining the search space to valid morpheme sequences. Besides, this figure shows that morphological disambiguation also improves speech recognition performance. We can conclude that morphological disambiguation improves the prediction power of morpholexical language model. The absence of morphotactics and disambiguation together has a larger impact on recognition performance.

D. Discriminative Reranking of ASR Hypotheses

The speech decoder generates word, sub-word or morpheme lattices depending on the units of the language model used in the first pass. Then, we extract an n-best list of hypotheses from these lattices which are ranked by the combined score obtained from the language and acoustic model. The resulting n-best hypotheses are reranked with a discriminative linear model trained with the perceptron algorithm using the features extracted from the hypotheses.

In the reranking experiments, we used the experimental setup of [40]. The n-best hypotheses for all systems are generated by decoding the acoustic training data with the corresponding generative model. The acoustic model trained on all the utterances in the training data is used to decode all the utterances. However, in language modeling, 12-fold cross validation is employed to prevent over-training of the discriminative model. This is done by decoding utterances in each fold with a fold-specific language model which is built by interpolating the generic language model trained on NewsCor corpus with the in-domain language model trained with the reference transcriptions of the utterances in the other 11 folds. The same interpolation constant -0.5 is used for building fold-specific language models of all systems. 200 K word, 76 K morph and 200 K lexical stem+ending units of vocabulary were employed while building 3-gram word, 4-gram morph, and 4-gram lexical

TABLE VI
DISCRIMINATIVE RERANKING RESULTS WITH THE PERCEPTRON USING UNIGRAM FEATURES

WER sensitive			–	–	✓	✓
Baseline score in training			–	✓	✓	–
Model	oracle	1-best	reranking			
word	15.0	23.4	23.2	23.0	22.9	23.0
morph	13.9	22.4	21.9	21.8	21.7	21.5
stem+ending (lexical)	13.7	21.6	21.1	20.9	20.9	20.8

stem+ending models, respectively. Since the n-gram language models are pruned for computational reasons, the lattices generated in the first-pass at ~ 1.5 real-time factor are rescored with unpruned language models.

We used the unigram counts of corresponding generative modeling units as features in the discriminative reranking experiments as these features give the most significant improvements in [40]. The word, morph, and lexical stem and ending unigram counts are our features respectively for word, morph and lexical stem+ending models. This results in 156 081 features for word, 46 251 features for morph and 63 887 features for lexical stem+ending model. The discriminative reranking using morphological features presents a complementary approach to the generative morpholexical language model in the sense that the recognition output from the generative model explicitly contains the reranking features. This prevents the necessity for any complicated feature extraction process after the first pass and it enables possibly to rerank the hypotheses on-the-fly during decoding alleviating rescoring latency using an algorithm similar to [43]. For word and statistically derived sub-word models, it was shown that using richer linguistic and statistically derived features further improves the reranking performance [37]. Note that this required to use linguistic processing steps like morphological parsing to extract the linguistic features. We have also recently experimented with more complex morpholexical and n-best list features [5].

The reranking models are trained both with the WER-sensitive perceptron algorithm of Fig. 2 and the original averaged perceptron algorithm. We also carried out experiments to see the effect of using baseline score in discriminative training. The 50-best hypotheses extracted for each utterance from the rescored lattices are used for the training and reranking. The number of iterations of the algorithm and the weight α_0 used for scaling the hypothesis score from the first-pass are optimized on a held-out set.

The final reranking results are given on the test set in Table VI. The WER-sensitive perceptron algorithm shows consistent improvements for all the models on the test set. Dismissing the first-pass score in the training of the standard perceptron degrades reranking performance on the test set. In contrast, it seems generally better to dismiss the first-pass score for the WER-sensitive perceptron. This is important since we might not have these scores if we want to train the discriminative model in an unsupervised manner where we don't have the transcribed acoustic data.

To evaluate the effectiveness of the WER-sensitive perceptron algorithm, we carried out significance tests using the NIST MAPSSWE test for the morph model where the algorithm seems to make significant difference. The WER-sensitive perceptron without the baseline score in training gives the best word error rate of 21.5% for the morph model as can be seen in Table VI. The performance improvement of this model over three other configurations is significant at the levels of $p = 0.048$, $p = 0.004$, and $p < 0.001$ with respect to the increasing word error rates of the configurations. The stem+ending model performs significantly better than the word and morph model for all the configurations $(p < 0.001)$.

VIII. DISCUSSION AND CONCLUSION

In this paper, we first introduced the morpholexical language model which is a morphological sub-lexical n-gram language model. Second, we showed that we can build a morphology-integrated search network for ASR using a morpholexical language model and the lexical transducer of the morphological parser in the finite-state transducer framework. This proposed approach is superior to word n-gram models in the following aspects:

- The vocabulary is unlimited since the modeling units are sub-lexical units.
- It alleviates the OOV and vocabulary growth problem. The OOV rate is effectively reduced to about 1.3% on the test set. For comparison, the 200 K word model has about 2% OOV rate.
- Using lexical units alleviates data sparsity problem.
- Lexical stem+ending model gives the best results, and it improves the WER over word model by 1.8% absolute.

Besides, it is superior to statistical sub-word (morph) models in some other aspects:

- The modeling units as being lexical and grammatical morphemes provide a linguistic approach.
- The linguistic approach enables integration with other finite-state models like pronunciation lexicon.
- The morphology-integrated search network only allows valid word sequences thanks to the morphotactics.
- The morphological features can be further exploited in a rescoring or reranking model.
- Lexical stem+ending model improves the WER over morph model by 0.4% absolute.

The experimental results show that lexical stem+ending model as a morpholexical language model outperforms all the other models significantly. We also show that morphotactics and morphological disambiguation are effective for better language modeling.

Third, we presented a variant of the perceptron algorithm, WER-sensitive perceptron, for discriminative training of reranking models. The experimental results show that this algorithm is better for reranking speech recognition hypotheses. The reranking WER for the lexical stem+ending model is lower by 2.2% and 0.7% absolute than word and morph models, respectively.

The proposed methods present an elegant approach for language modeling for morphologically complex languages. However, it should be noted that these approaches increases the language modeling complexity and requires to have a finite-state morphological parser for a language to apply these methods.

Although, the language models and techniques in this work have been developed and applied for Turkish speech recognition, they can be applied for other morphologically rich languages such as Arabic, Finnish, and Czech and in other language processing applications. We believe that using grammatical sublexical units in language modeling can be even more beneficial for other applications especially for machine translation.

APPENDIX

We give a proof of the convergence of the WER-sensitive perceptron algorithm for a separable training sequence following a similar proof given for the averaged perceptron in [29].

Definition 1: Let the set of incorrect candidates for an example x_i be $\overline{\mathbf{GEN}}(x_i) = \mathbf{GEN}(x_i) - \mathcal{Y}_i$ where $\mathcal{Y}_i$ is the set of all candidates with the lowest error rate. The training sequence $(x_i, y_i \in \mathcal{Y}_i)$ for $i = 1 \ldots n$ is separable with margin $\delta > 0$ if there exists some vector $\mathbf{U}$ with $\|\mathbf{U}\| = 1$ such that

$$\forall i, \forall z_i \in \overline{\mathbf{GEN}}(x_i), \mathbf{U} \cdot \mathbf{\Phi}(x_i, y_i) - \mathbf{U} \cdot \mathbf{\Phi}(x_i, z_i) \geq \delta \quad (1)$$

Theorem 1: For any training sequence (x_i, y_i) which is separable with margin δ, for the perceptron algorithm in Fig. 2:

$$\text{Number of mistakes} \leq \frac{R^2 r^2}{\delta^2} \quad (2)$$

where R is a constant such that $\forall i, \forall z_i \in \overline{\mathbf{GEN}}(x_i)$, $\|\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)\| \leq R$ and r is an upper bound on loss for any candidate, that is $\forall i, \forall z_i \in \overline{\mathbf{GEN}}(x_i)$, $\Delta_i(z_i, y_i) \leq r$ where $\Delta_i(z_i, y_i)$ is the difference in the edit distances of z_i and y_i with the reference transcription of x_i, and $y_i \in \mathcal{Y}_i$ is a candidate with the lowest error rate.

Proof: Suppose that $k'th$ mistake is made at the $i'th$ example and let $\bar{\alpha}^k$ be the weights before that mistake is made and hence $\bar{\alpha}^1 = 0$. Take z_i as the output proposed at this example, $z_i = \arg\max_{z \in \mathbf{GEN}(x_i)} \mathbf{\Phi}(x_i, z) \cdot \bar{\alpha}^k$. It follows from the algorithm updates that $\bar{\alpha}^{k+1} = \bar{\alpha}^k + (\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i))\Delta_i(z_i, y_i)$. First we derive a lower bound for $\|\bar{\alpha}^{k+1}\|$ as follows:

$$\begin{aligned}\mathbf{U} \cdot \bar{\alpha}^{k+1} &= \mathbf{U} \cdot \bar{\alpha}^k + \mathbf{U} \cdot (\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i))\Delta_i(z_i, y_i) \\ &\geq \mathbf{U} \cdot \bar{\alpha}^k + \delta \Delta_i(z_i, y_i)\end{aligned}$$

where the inequality follows from the definition of $\mathbf{U}$. Since $\mathbf{U} \cdot \bar{\alpha}^1 = 0$, it follows by induction on k that for all k, $\mathbf{U} \cdot \bar{\alpha}^{k+1} \geq \delta \sum_k \Delta_i(z_i, y_i)$ where $\sum_k \Delta_i(z_i, y_i)$ is the sum of losses made at each mistake up to $k'th$ mistake. Because $\mathbf{U} \cdot \bar{\alpha}^{k+1} \leq \|\mathbf{U}\| \|\bar{\alpha}^{k+1}\|$ and $\|\mathbf{U}\| = 1$, it follows that $\|\bar{\alpha}^{k+1}\| \geq \delta \sum_k \Delta_i(z_i, y_i)$. Because the loss is at least one for each mistake by definition, it follows that $\|\bar{\alpha}^{k+1}\| \geq \delta k$

Now we derive an upper bound for $\|\bar{\alpha}^{k+1}\|^2$ as follows:

$$\begin{aligned}\|\bar{\alpha}^{k+1}\|^2 &= \|\bar{\alpha}^k\|^2 + \|\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)\|^2 \Delta_i(z_i, y_i)^2 \\ &\quad + 2\bar{\alpha}^k \cdot (\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i))\Delta_i(z_i, y_i) \\ &\leq \|\bar{\alpha}^k\|^2 + R^2 \Delta_i(z_i, y_i)^2\end{aligned}$$

where the inequality follows because $\|\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)\|^2 \leq R^2$ by assumption, and $\bar{\alpha}^k \cdot (\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)) \leq 0$ because z_i is the highest scoring candidate for x_i under the parameters $\bar{\alpha}^k$. It follows by induction that $\|\bar{\alpha}^{k+1}\|^2 \leq R^2 \sum_k \Delta_i(z_i, y_i)^2$. Because the loss for any candidate has an upper bound r by assumption, it follows that $\|\bar{\alpha}^{k+1}\|^2 \leq R^2 k r^2$.

Combining the bounds $\|\bar{\alpha}^{k+1}\| \geq \delta k$ and $\|\bar{\alpha}^{k+1}\|^2 \leq R^2 k r^2$ gives the result for all k that

$$\delta^2 k^2 \leq \|\bar{\alpha}^{k+1}\|^2 \leq R^2 k r^2 \Rightarrow k \leq \frac{R^2 r^2}{\delta^2}.$$

Because the upper bound on the number of mistakes the algorithm makes is constant, the algorithm must converge within a finite number of iterations. ■

REFERENCES

[1] H. Sak, T. Güngör, and M. Saraçlar, "Morphological disambiguation of Turkish text with perceptron algorithm," in *Proc. CICLing '07*, 2007, vol. LNCS 4394, pp. 107–118.

[2] H. Sak, T. Güngör, and M. Saraçlar, "Turkish language resources: Morphological parser, morphological disambiguator and web corpus," in *GoTAL 2008*. New York: Springer, 2008, vol. 5221, pp. 417–427, ser. LNCS.

[3] H. Sak, M. Saraçlar, and T. Güngör, "Integrating morphology into automatic speech recognition," in *Proc. ASRU*, 2009, pp. 354–358.

[4] H. Sak, M. Saraçlar, and T. Güngör, "Morphology-based and sub-word language modeling for Turkish speech recognition," in *Proc. ICASSP*, 2010, pp. 5402–5405.

[5] H. Sak, M. Saraçlar, and T. Güngör, "Discriminative reranking of ASR hypotheses with morpholexical and n-best-list features," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 202–207.

[6] C. Allauzen, M. Mohri, and B. Roark, "Generalized algorithms for constructing statistical language models," in *Proc. ACL*, 2003, pp. 40–47.

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

[8] P. Ircing, P. Krbec, J. Hajic, J. Psutka, S. Khudanpur, F. Jelinek, and W. Byrne, "On large vocabulary continuous speech recognition of highly inflectional language-Czech," in *Proc. Eurospeech*, 2001, pp. 487–490.

[9] O.-W. Kwon and J. Park, "Korean large vocabulary continuous speech recognition with morpheme-based recognition units," *Speech Commun.*, vol. 39, no. 3–4, pp. 287–300, 2003.

[10] D. Z. Hakkani-Tür, "Statistical language modeling for agglutinative languages," Ph.D. dissertation, Bilkent Univ., Ankara, Turkey, 2000.

[11] H. Erdogan, O. Buyuk, and K. Oflazer, "Incorporating language constraints in sub-word based speech recognition," in *Proc. ASRU*, 2005, pp. 98–103.

[12] E. Arisoy, H. Sak, and M. Saraçlar, "Language modeling for automatic Turkish broadcast news transcription," in *Proc. Interspeech*, 2007, pp. 2381–2384.

[13] E. Arısoy, D. Can, S. Parlak, H. Sak, and M. Saraçlar, "Turkish broadcast news transcription and retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 5, pp. 874–883, Jul. 2009.

[14] T. Rotovnik, M. S. Maučec, and Z. Kačič, "Large vocabulary continuous speech recognition of an inflected language using stems and endings," *Speech Commun.*, vol. 49, pp. 437–452, Jun. 2007.

[15] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to Finnish," *Comput. Speech Lang.*, vol. 20, no. 4, pp. 515–541, 2006.

[16] P. Mihajlik, T. Fegyó, Z. Tüske, and P. Ircing, "A morpho-graphemic approach for the recognition of spontaneous speech in agglutinative languages-like Hungarian," in *Proc. Interspeech*, 2007, pp. 1497–1500.

[17] E. Arısoy and M. Saraçlar, "Lattice extension and vocabulary adaptation for Turkish LVCSR," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 1, pp. 163–173, Jan. 2009.

[18] K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Comput. Speech Lang.*, vol. 20, no. 4, pp. 589–608, 2006.

[19] R. Sarikaya, M. Afify, Y. Deng, H. Erdogan, and Y. Gao, "Joint morphological-lexical language modeling for processing morphologically rich languages with application to dialectal Arabic," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 7, pp. 1330–1339, Sep. 2008.

[20] H. Sak, T. Güngör, and M. Saraçlar, "Resources for Turkish morphological processing," *Lang. Resources Eval.*, vol. 45, no. 2, pp. 249–261, 2011.
[21] M. Mohri, "Finite-state transducers in language and speech processing," *Comput. Linguist.*, vol. 23, no. 2, pp. 269–311, 1997.
[22] K. Koskenniemi, "A general computational model for word-form recognition and production," in *Proc. ACL*, 1984, pp. 178–181.
[23] K. Oflazer, "Two-level description of Turkish morphology," *Literary Linguist. Comput.*, vol. 9, no. 2, pp. 137–148, 1994.
[24] L. Karttunen and K. R. Beesley, "Two-level rule compiler," Xerox Palo Alto Research Center, Palo Alto, CA, Tech. Rep., 1992.
[25] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Proc.. CIAA '07*. New York: Springer, 2007, vol. 4783, pp. 11–23, ser. LNCS.
[26] K. Oflazer and S. Inkelas, "The architecture and the implementation of a finite state pronunciation lexicon for Turkish," *Comput. Speech Lang.*, vol. 20, pp. 80–106, Jan. 2006.
[27] Freund and Schapire, "Large margin classification using the perceptron algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, 1999.
[28] M. Collins and N. Duffy, "New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron," in *ACL*, 2002, pp. 263–270.
[29] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proc. EMNLP*, 2002, pp. 1–8.
[30] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *ACL*, Stroudsburg, PA, 2002, pp. 21–30, ser. MPL'02.
[31] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 69–88, 2002.
[32] M. Mohri and M. Riley, "Integrated context-dependent networks in very large vocabulary speech recognition," in *Proc. Eurospeech*, 1999, pp. 811–814.
[33] C. Allauzen, M. Mohri, M. Riley, and B. Roark, "A generalized construction of integrated speech recognition transducers," in *Proc. ICASSP*, 2004, pp. 761–764.
[34] C. Allauzen, M. Mohri, and B. Roark, "The design principles and algorithms of a weighted grammar library," *Int. J. Foundations Comput. Sci.*, vol. 16, no. 3, pp. 403–421, 2005.
[35] L. Shen and A. K. Joshi, "Ranking and reranking with perceptron," *Mach. Learn.*, vol. 60, pp. 73–96, Sep. 2005.
[36] B. Roark, M. Saraçlar, and M. Collins, "Discriminative n-gram language modeling," *Comput. Speech Lang.*, vol. 21, no. 2, pp. 373–392, Apr. 2007.
[37] E. Arısoy, M. Saraçlar, B. Roark, and I. Shafran, "Discriminative language modeling with linguistic and statistically derived features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 540–550, Feb. 2012.
[38] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.
[39] N. Singh-Miller and M. Collins, "Trigger-based language modeling using a loss-sensitive perceptron algorithm," in *Proc. ICASSP*, 2007, vol. 4, pp. IV-25–IV-28.
[40] E. Arısoy, "Statistical and discriminative language modeling for Turkish large vocabulary continuous speech recognition," Ph.D. dissertation, Boğaziçi Univ., Ankara, Turkey, 2009.
[41] A. Stolcke, "SRILM—An extensible language modeling toolkit," in *Proc. ICSLP*, 2002, vol. 2, pp. 901–904.
[42] C. Chelba, T. Brants, W. Neveitt, and P. Xu, "Study on interaction between entropy pruning and Kneser-Ney smoothing," in *Proc. Interspeech*, 2010, pp. 2242–2245.
[43] H. Sak, M. Saraçlar, and T. Güngör, "On-the-fly lattice rescoring for real-time automatic speech recognition," in *Proc. Interspeech*, 2010, pp. 2450–2453.

Haşim Sak (M'10) received the B.S. degree from the Computer Engineering Department, Bilkent University, Ankara, Turkey, in 2000 and the M.S. and Ph.D. degrees from the Computer Engineering department, Boğaziçi University, Istanbul, Turkey, in 2004 and 2011, respectively. His thesis study focused on the language modeling and speech decoding challenges associated with agglutinative languages and rich morphology.

His main research interests include speech decoding, statistical language modeling, morphological parsing, spelling correction, and morphological disambiguation. He is currently with Google, Inc., New York.

Murat Saraçlar (M'00) received the B.S. degree from the Electrical and Electronics Engineering Department, Bilkent University, Ankara, Turkey, in 1994 and the M.S.E. and Ph.D. degrees from the Electrical and Computer Engineering Department, Johns Hopkins University, Baltimore, MD, in 1997 and 2011, respectively.

He is currently an Associate Professor in the Electrical and Electronic Engineering Department, Boğaziçi University, Istanbul, Turkey. From 2000 to 2005, he was with the Multimedia Services Department at AT&T Labs Research. His main research interests include all aspects of speech recognition, its applications, speech and language processing, and machine learning.

Dr. Saraçlar was a member of the IEEE Signal Processing Society Speech and Language Technical Committee (2007–2009). He is currently serving as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS and the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. He is on the editorial boards of *Computer Speech and Language*, and *Language Resources and Evaluation*.

Tunga Güngör received the Ph.D. degree from Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey, in 1995.

He is an Associate Professor in the Department of Computer Engineering, Boğaziçi University. His research interests include natural language processing, machine translation, machine learning, pattern recognition, and automated theorem proving. He published about 60 scientific articles, and participated in several research projects and conference organizations. He is currently a Visiting Professor at the Center for Language and Speech Technologies and Applications at Universitat Politecnica de Catalunya, Barcelona, Spain.