# Discriminative Reranking of ASR Hypotheses with Morpholexical and N-best-list Features

Haşim Sak #1, Murat Saraçlar *2, Tunga Güngör #3

# *Department of Computer Engineering, Boğaziçi University*
*34342, Bebek, İstanbul, Turkey*
1 hasim.sak@boun.edu.tr
3 gungort@boun.edu.tr

* *Department of Electrical & Electronics Engineering, Boğaziçi University*
*34342, Bebek, İstanbul, Turkey*
2 murat.saraclar@boun.edu.tr

***Abstract*—This paper explores rich morphological and novel $n$-best-list features for reranking automatic speech recognition hypotheses. The morpholexical features are defined over the morphological features obtained by using an $n$-gram language model over lexical and grammatical morphemes in the first-pass. The $n$-best-list features for each hypothesis are defined using that hypothesis and other alternate hypotheses in an $n$-best list. Our methodology is to align each hypothesis with other hypotheses one by one using minimum edit distance alignment. This gives us a set of edit operations - substitution, addition and deletion as seen in these alignments. These edit operations constitute our $n$-best-list features as indicator features. The reranking model is trained using a word error rate sensitive averaged perceptron algorithm introduced in this paper. The proposed methods are evaluated on a Turkish broadcast news transcription task. The baseline systems are word and statistical sub-word systems which also employ morphological features for reranking. We show that morpholexical and $n$-best-list features are effective in improving the accuracy of the system (0.8%).**

## I. Introduction

Discriminative ranking and reranking approaches have been proposed as an alternative to generative history-based probabilistic models [1], [2], [3], [4]. In reranking tasks, a baseline generative model generates a set of candidates, and then these candidates are reranked by using local and global features, generally including the likelihood scores from the baseline model. For instance, in parsing, a baseline parser produces a set of candidate parses for each input sentence, with their associated probabilities which define an initial ranking over these parses. Then, a reranking model can be used to rerank these parses with the anticipation of improving the initial rankings using additional features derived from the parse tree. As an advantage over generative models, the discriminative reranking models allow the use of arbitrary features as evidence without concerning about the interaction of features or the construction of a generative model using these features.

As a discriminative reranking approach, the variants of the perceptron algorithm proved to be very successful. The perceptron is a simple artificial neural network which can be used as a binary linear classifier [5]. A variant of the perceptron algorithm, the voted perceptron has been applied to classification tasks in natural language processing [6]. Another variant of the algorithm, the averaged perceptron has been shown to outperform Maximum Entropy Models [7] in part-of-speech tagging and parsing tasks [2]. The averaged perceptron algorithm has also been used to estimate discriminatively trained $n$-gram language models for large vocabulary speech recognition [8]. The discriminative language models are trained on acoustic sequences with their transcriptions, in an attempt to directly optimize word error rate. The perceptron algorithm has also been used to build discriminative language models for Turkish using syntactic and sub-lexical features [9]. A variant of the perceptron algorithm for pairwise classification with uneven margins has been applied to the task of parse reranking and machine translation reranking [4]. The word error rate (WER) of speech recognition hypotheses have been used to choose the competitors in discriminative reranking of ASR hypotheses [10].

We previously proposed a morphology-based language model for Turkish, which is an $n$-gram language model estimated over lexical and grammatical morphemes [11]. In that study, we built a morphology-integrated search network by composing this morpholexical language model with the lexical transducer of a morphological parser for Turkish. In addition to alleviating the out-of-vocabulary word problem by using a computational lexicon and hence improving the accuracy of the speech recognition system, this approach has the advantage that we can obtain morphological units annotated with morphological features as output from the system. This paper explores using these features to further exploit morphology for improving the recognition accuracy in a reranking framework. In addition to morphological features, we propose $n$-best-list features which are extracted for each hypothesis relative to other candidate hypotheses. When the discriminative models are considered as an error corrective model, these features aim to capture morpheme confusions as discriminative features between the correct hypothesis and alternate hypotheses.

We improve speech recognition accuracy by using morpholexical and $n$-best-list features in a discriminative $n$-best hypotheses reranking framework with a variant of the perceptron algorithm. The perceptron algorithm is tailored for rerank-

978-1-4673-0367-5/11/$26.00 ©2011 IEEE

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

Fig. 1. The averaged perceptron algorithm

ing recognition hypotheses by introducing error rate dependent loss function and using one versus all hypotheses $n$-best-list features extracted by pairwise alignment of hypotheses.

## II. Discriminative Reranking with Perceptron

The introduction of arbitrary and global features into the generative models results in difficulty due to dynamic programming 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 [4]. The perceptron algorithm has been successfully applied to various NLP tasks for ranking or reranking hypotheses [6], [1], [2], [4], [8]. We also used the perceptron algorithm for morphological disambiguation of Turkish text using morpholexical features [12]. The characteristics like simplicity, fast convergence, and easy incorporation of arbitrary local and global features make the perceptron algorithm very attractive for discriminative training of linear models. Hence, we select the perceptron algorithm to train linear models with the morpholexical features to rerank the $n$-best hypotheses generated by the morpholexical language models. Since those hypotheses already have the morphological parse representation, we try to exploit the morphological information to improve the ranking accuracy. To accomplish that we experimented with various features and selected a subset of features effective in discrimination. We also introduce some variants of the perceptron algorithm and evaluate their effectiveness.

### A. The Perceptron Algorithm

The perceptron is a linear classifier [5]. The perceptron algorithm tries to learn a weight vector that minimizes the number of misclassifications. Fig. 1 shows a variant of the perceptron algorithm - the averaged perceptron [6], [2] formulated as a multiclass classifier. 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. $\mathrm{argmax}_{z \in \mathbf{GEN}(x)} \mathbf{\Phi}(x, z) \cdot \bar{\alpha}$. The given algorithm can also be used to rank the possible outputs for an input $x$ by their $\mathbf{\Phi}(x, z) \cdot \bar{\alpha}$ scores.

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 $\alpha$ 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 [2]. The averaged parameters $\gamma$ 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 the efficiency reasons and very small performance gains with the lattices.
- $y_i$ is the member of the $\mathbf{GEN}(x_i)$ with 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 $y$.
  We also define features between a given transcription and the rest of the transcriptions for an utterance. The next section describes the features in more detail.
- The expression $\mathbf{\Phi}(x, y) \cdot \alpha$ denotes the inner product $\sum_{j=1}^{d} \Phi_j(x, y) \alpha_j$, where $\alpha_j$ is the $j^{th}$ component of the parameter vector $\alpha$.
- The zeroth component $\Phi_0(x, y)$ represents the log-probability of $y$ in the lattice output from the baseline recognizer for utterance $x$. The corresponding weight $\alpha_0$ 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 $\gamma$ that can be used to choose the

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

Fig. 2. The WER-sensitive perceptron algorithm

transcription $y$ having hopefully the least number of errors for an utterance $x$ using the following function:

$$F(x) = \underset{y \in \mathbf{GEN}(x)}{\mathrm{argmax}} \; \mathbf{\Phi}(x, y) \cdot \gamma$$

### B. The WER-Sensitive Perceptron Algorithm

We improve the perceptron algorithm in two ways for reranking ASR recognition hypotheses. First, 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. The loss function of the original perceptron 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 define a better 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 $x_i$. Hence, this enables us to incorporate the word error rate metric into training of the discriminative model.

The gradient of the loss function can be found by the stochastic gradient descent as $\lambda \Delta(y_i, z_i)(\mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i))$, where $\lambda$ is the learning rate. For the experiments in this paper, we set the $\lambda$ to 1, for which we found the algorithm to converge well.

Note that a loss-sensitive perceptron algorithm has been proposed for reranking speech recognition output in [13]. 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.

The second improvement comes from the observation that the candidate hypotheses generated by the baseline model carry $n$-best-list information which can be used as discriminative features between the alternatives. For instance, these features can be effective in capturing frequently occurring errors due to acoustic similarity. They can also define $n$-best-list similarity metrics between hypotheses to improve the classification. The word error rate sensitive perceptron algorithm is shown in Fig. 2.

### C. Morpholexical Features

In this study, we used lexical stem-ending units [14] for building the first-pass language model. The speech decoder using the morpholexical language models outputs a sequence of lexical and grammatical morphemes annotated with morphological features. Most of the features that we experimented are defined as the $n$-gram counts of these lexical and grammatical morphemes. We also employed part-of-speech tag of words as a feature. The set of features that we incorporate in the model is a superset of the features used for morphological disambiguation [15]. The feature templates are given in Table I. The notation and feature representation is explained here. $w_i$ represents the current morpholexical word of the trigram - this is the concatenation of all the morphemes for a word as output from the recognizer. The morpholexical word $w_i$ is split into a lexical root $r_i$ and a lexical ending $e_i$. The lexical ending $e_i$ is the concatenation of the grammatical morphemes $m_{i,j}$ for $j = 1 \ldots n_i$, where $n_i$ is the number of grammatical morphemes in $w_i$. The morphological features not having grammatical morphemes are concatenated to the previous lexical or grammatical morpheme. Shortly, the morpholexical word $w_i$ is represented as $w_i = r_i e_i = r_i m_{i,1} m_{i,2} \ldots m_{i,n_i}$. An example for the morpholexical features of the word *sevmediği* is given below.

$w_i$ =*sev*[Verb]+*mA*[Neg]-*DHk*[Noun+PastPart]
+[A3sg]+*SH*[P3sg]+[Nom]
$r_i$ =*sev*[Verb]
$m_{i,1}$ =+*mA*[Neg]
$m_{i,2}$ =-*DHk*[Noun+PastPart]+[A3sg]
$m_{i,3}$ =+*SH*[P3sg]+[Nom]
$e_i = m_{i,1} m_{i,2} m_{i,3}$
$t_i$ =*Noun* (the POS tag of the last derived word)

### D. N-best-list Features

As a novel approach, we experimented with $n$-best-list features, which are defined between a given transcription $y$ and the other alternate transcriptions $\mathbf{GEN}(x) \setminus \{y\}$. The idea comes from the observation that sometimes the alternate hypotheses give us contrastive information for the correct hypotheses. For instance, the transformation-based learning has been used to learn a set of rules for discriminating between the correct and alternate hypotheses in a confusion set using additional knowledge sources extracted from the confusion networks [16]. The discriminative language models can also be considered as error corrective models. Therefore, it is reasonable to introduce features encoding the confusion of the words and the similarity of the hypotheses in terms of error rate. Our approach for incorporating

```
(1) uzman[Noun]+[A3sg]+[Pnon]+[Nom] kişi[Noun] +lAr[A3pl]+[Pnon]+[Nom] için[Postp]
(2) zam[Noun]+[A3sg]+[Pnon]+[Nom]   kişi[Noun] +lAr[A3pl]+[Pnon]+[Nom]
```

(1) subs: *zam[Noun]+[A3sg]+[Pnon]+[Nom]→uzman[Noun]+[A3sg]+[Pnon]+[Nom]*, add: *için[Postp]*, avg_edit_dist: *2*
(2) subs: *uzman[Noun]+[A3sg]+[Pnon]+[Nom]→zam[Noun]+[A3sg]+[Pnon]+[Nom]*, del: *için[Postp]*, avg_edit_dist: *2*

Fig. 3. An example for indicator n-best-list features extracted for each hypotheses by minimum edit distance alignment

TABLE I
FEATURE TYPES EXPERIMENTED FOR DISCRIMINATIVE RERANKING

| Gloss | Feature |
|---|---|
| morpholexical word unigram | (1) $w_i$ |
| morpholexical word bigram | (2) $w_{i-1}w_i$ |
| lexical root unigram | (3) $r_i$ |
| lexical root bigram | (4) $r_{i-1}r_i$ |
| lexical ending unigram | (5) $e_i$ |
| lexical ending bigram | (6) $e_{i-1}e_i$ |
| number of grammatical morphemes | (7) $n_i$ |
| grammatical morphemes | (8) $m_{i,j}$ $(j = 1 \dots n_i)$ |
| lexical ending with previous morpholexical | (9) $w_{i-1}e_i$ |
| lexical ending with previous lexical root | (10) $r_{i-1}e_i$ |
| POS tag unigram | (11) $t_i$ |
| POS tag bigram | (12) $t_{i-1}t_i$ |
| POS tag with previous morpholexical word | (13) $w_{i-1}t_i$ |
| POS tag with previous lexical ending | (14) $e_{i-1}t_i$ |
| $n$-best-list features | (15) *add*, *del*, *subs* |
| average edit-distance | (16) *avg_edit_distance* |

these features is to align a given hypotheses $y$ with each alternate hypothesis and set the indicator features for the seen edit operations - substitution, addition and deletion of morphemes. For instance, an indicator feature for the substitution of acoustically similar lexical endings can be *subs:+SH[P3sg]+NDA[Loc]→+SH[P3sg]+NDAn[Abl]*. These features aim to learn error corrective rules such as if a word is frequently mistaken with another word then the occurrence of the same mistake in the alternate hypotheses should signal that the word is likely to be in the utterance. We also used the average edit distance with the alternate hypotheses as a feature. Fig. 3 shows an example of $n$-best-list features for two hypotheses. The two hypotheses are aligned and edit operations are used as indicator features for both hypotheses.

## III. EXPERIMENTS

This section gives experimental results for the application of proposed discriminative reranking methods 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 10843 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 [17], [18]. 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 [15] 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 of this paper 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 [19]. 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. 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 Arısoy [18]. In the first set of experiments, we compared the discriminative reranking results of the morpholexical model with the word and morphs (statistical sub-words) model as shown in Table II. The results for word and morph models are also taken from Arısoy's work [18]. 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. 200K word, 76K morph and 200K lexical stem-ending units of vocabulary were employed while building 3-gram word, 4-gram morphs, and 4-gram lexical stem-ending models, respectively. Since the $n$-gram language models are pruned for computational reasons, the lattices generated in the first-pass at $\sim$1.5 real-time factor are rescored with unpruned language models.

The best reranking result for the word-based system is obtained by using the unigram counts of inflectional groups (IGs) and roots as features. The morphological analysis of a word is split at derivation boundaries to give the IGs. The best reranking result for the morph-based system is obtained by using the unigram counts of morphs and bigram counts of part-of-speech tags of words. POS tags are obtained by converting morphs to words and using morphological parsing and disambiguation tools. Note that best reranking results for both words and morphs are obtained by using linguistic tools to process the hypothesis to obtain the morphological information. In contrast, the morphological information is readily available for reranking in the morpholexical approach since it is carried on the morpholexical units. For the lexical stem-ending model, we experimented with the features given in Table I. The features numbered 3, 7, and 13 are selected by incrementally adding best performing features on a held-out set and these features are used on the test set.

The reranking models are trained with the perceptron algorithm of Fig. 1. 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 $\alpha_0$ used for scaling the hypothesis score from the first-pass are optimized on a held-out set. The held-out set is also used to decide which feature types will be incorporated in the model based on the reranking performance on that set. The final reranking results are given on a test set in Table II. We observe that lexical stem-ending model has the best baseline and reranking performance. However, we can see that reranking improvement of the lexical stem-ending model over the baseline is not as large as word and morph models. This can be explained by noting that the lexical stem-ending model already uses the morphological information implicitly in the morpholexical $n$-gram model of the first pass. Therefore, it can be expected that morphological information improves the word and morph model more than the lexical stem-ending model in the reranking. This is more evident when we consider that the morph model can generate invalid word forms, which may be corrected by using the morphological information.

The second set of experiments is designed to evaluate the effectiveness of WER-sensitive perceptron algorithm and $n$-best-list features proposed in section II. In this reranking experiments, we use 10-best hypotheses for two reasons. First, they show nearly identical performance with 50-best hypotheses on a held-out set with the standard features. Second, it becomes computationally prohibitive to extract $n$-best-list features with longer hypotheses lists since the run time complexity of the feature extraction algorithm is $O(n^2m^2)$, where $n$ is the number of hypotheses and $m$ is the average number of units in the hypotheses. The features are selected by incrementally adding the best performing feature on a held-out set from Table I. The feature selection is repeated for both algorithms. The reranking results are given in Table III. The WER-sensitive perceptron algorithm and $n$-best-list features show consistent improvements on the test set, however, the improvements are not very significant. Nevertheless, the reranking with the WER-sensitive perceptron and $n$-best-list features improves the WER by 0.3% relative to the original perceptron and 1-best-list features, and by 0.8% relative to the first-pass recognition result.

TABLE II
DISCRIMINATIVE RERANKING RESULTS WITH PERCEPTRON

| **Models** | **50-best oracle** | **1-best** | **Reranked** |
|---|---|---|---|
| Words | 15.0 | 23.4 | 22.5 |
| Statistical morphs | 13.9 | 22.4 | 21.4 |
| Lexical stem-ending | 14.2 | 21.7 | 21.2 |

TABLE III
EVALUATION OF THE WER-SENSITIVE PERCEPTRON AND $n$-BEST-LIST FEATURES FOR RERANKING 10-BEST HYPOTHESES

| **Algorithm-Features** | **Test** |
|---|---|
| 10-best oracle | 16.5 |
| 1-best | 21.7 |
| Perceptron | |
| + morpholexical features (3,7,13) | 21.2 |
| + $n$-best-list features | 21.0 |
| WER-sensitive Perceptron | |
| + morpholexical features (3,7,13) | 21.1 |
| + $n$-best-list features | 20.9 |

## IV. CONCLUSION

In this paper, we experimented with a proposed word error rate sensitive perceptron algorithm using morpholexical and novel $n$-best-list features to exploit morphological information in a discriminative reranking framework. We tailored the perceptron algorithm to better suit for reranking recognition hypotheses by introducing error rate dependent loss function and using one versus all hypotheses $n$-best-list features extracted by pairwise minimum edit distance alignment of hypotheses. We applied the proposed methods for reranking $n$-best list hypotheses from a first-pass recognition using a morphology-based language model for Turkish. We compared our results with a baseline word and statistical sub-word model in the same reranking framework. The improvements of the first-pass in word error rate are mostly preserved in the reranking as 1.6% absolute over word models and 0.5% absolute over statistical sub-word models.

## ACKNOWLEDGMENT

This work was supported by the Boğaziçi University Research Fund under the grant numbers 06A102 and 08M103, the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the grant numbers 107E261, 105E102 and 109E142. Haşim Sak is supported by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610 and TÜBİTAK BİDEB 2211. Murat Saraçlar is supported by the TUBA-GEBIP award.

# REFERENCES

[1] 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.

[2] M. Collins, “Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms,” in *EMNLP*, 2002.

[3] M. Collins and T. Koo, “Discriminative Reranking for Natural Language Parsing,” *Computational Linguistics*, vol. 31, no. 1, pp. 25–70, Mar. 2005.

[4] L. Shen and A. K. Joshi, “Ranking and reranking with perceptron,” *Mach. Learn.*, vol. 60, pp. 73–96, September 2005.

[5] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[6] Freund and Schapire, “Large margin classification using the perceptron algorithm,” *Machine Learning*, vol. 37, no. 3, pp. 277–296, 1999.

[7] R. Rosenfeld, “Adaptive statistical language modeling: A maximum entropy approach,” Ph.D. dissertation, Carnegie Mellon University, 1994.

[8] B. Roark, M. Saraçlar, and M. Collins, “Discriminative n-gram language modeling,” *Computer Speech and Language*, vol. 21, no. 2, pp. 373–392, April 2007.

[9] E. Arısoy, M. Saraçlar, B. Roark, and I. Shafran, “Syntactic and sub-lexical features for Turkish discriminative language models,” in *ICASSP*, 2010, pp. 5538–5541.

[10] T. Oba, T. Hori, and A. Nakamura, “An approach to efficient generation of high-accuracy and compact error-corrective models for speech recognition,” in *INTERSPEECH*, 2007, pp. 1753–1756.

[11] H. Sak, M. Saraçlar, and T. Güngör, “Integrating morphology into automatic speech recognition,” in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 354–358.

[12] H. Sak, T. Güngör, and M. Saraçlar, “Morphological disambiguation of Turkish text with perceptron algorithm,” in *CICLing 2007*, vol. LNCS 4394, 2007, pp. 107–118.

[13] N. Singh-Miller and M. Collins, “Trigger-Based Language Modeling using a Loss-Sensitive Perceptron Algorithm,” in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, 2007, pp. IV–25–IV–28.

[14] E. Arısoy, H. Sak, and M. Saraçlar, “Language modeling for automatic Turkish broadcast news transcription,” in *INTERSPEECH*, 2007, pp. 2381–2384.

[15] H. Sak, T. Güngör, and M. Saraçlar, “Resources for turkish morphological processing,” *Language Resources and Evaluation*, vol. 45, no. 2, pp. 249–261, 2011.

[16] L. Mangu and M. Padmanabhan, “Error corrective mechanisms for speech recognition,” *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, pp. 29–32, 2001.

[17] E. Arısoy, D. Can, S. Parlak, H. Sak, and M. Saraçlar, “Turkish broadcast news transcription and retrieval,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 17, no. 5, pp. 874–883, 2009.

[18] E. Arısoy, “Statistical and discriminative language modeling for turkish large vocabulary continuous speech recognition,” Ph.D. dissertation, Boğaziçi University, 2009.

[19] A. Stolcke, “SRILM – an extensible language modeling toolkit,” in *Proceedings of ICSLP*, vol. 2, 2002, pp. 901–904, http://www.speech.sri.com/projects/srilm/.