# Feature Combination Approaches for Discriminative Language Models

*Ebru Arısoy, Bhuvana Ramabhadran, Hong-Kwang Jeff Kuo*

IBM T.J. Watson Research Center
Yorktown Heights, NY, 10598
{earisoy, bhuvana, hkuo}@us.ibm.com

10.21437/Interspeech.2011-245

## Abstract

This paper focuses on feature combination approaches for discriminative language models (DLMs). DLM is a feature-based log-linear language modeling approach where the feature parameters are estimated discriminatively. DLM allows for easy integration of various knowledge sources into language modeling. Choosing the proper strategy when combining features coming from different information sources is important. We investigated three approaches for combining lexical, word class, and acoustic features in DLMs. The three approaches are joint parameter estimation, cascade training, and model score combination. The cascade approach is an interesting approach that finally gave the best test set performance, improving the word error rate by 0.49% absolute (3% relative) on transcription of English Broadcast News. The word class features and state duration features were found to be very complementary, and their combination provided most of the improvement.

**Index Terms**: feature combination, discriminative language models

## 1. Introduction

In state-of-the-art ASR systems, $n$-gram language model probabilities are calculated with Maximum Likelihood Estimation. The discriminative language model (DLM) is a log-linear model complementary to the baseline $n$-gram language model. In contrast to the generative language model, it is trained on utterances with their transcripts, containing negative (recognition errors) and positive examples, to optimize the word error rate. The DLM utilizes lattices or $N$-best lists as the training data. When $N$-best lists are used, each hypothesis in the training data is represented as a feature vector. The feature parameters can be learned using different algorithms [1, 2, 3].

DLM features are extracted from the training sentence hypotheses. A basic feature set utilized in DLM training is the word $n$-grams. The number of occurrences of each $n$-gram is used as feature values to represent a sentence hypothesis. In [2], a DLM utilizing word $n$-grams as features outperformed the traditional generative $n$-gram approach. As a feature-based language modeling approach, the DLM can incorporate any kind of information easily into the language model. Linguistic information is incorporated into DLMs as part-of-speech (PoS) tag features [4, 5], parse tree features [4, 5], morphological features [6, 7], and discourse level trigger features [8]. Automatically induced sub-word classes are used as DLM features in sub-word-based ASR systems [5]. In addition, even acoustic information such as state transitions and state durations have been encoded as DLM features [9]. The common approach in all these previous studies is to concatenate all the features coming from different information sources in the hypothesis feature vector and to estimate the parameters of these features jointly.

In this paper we first investigate the traditional joint estimation approach when combining feature sets from different information sources. Then we propose a cascade parameter estimation approach which aims to reveal complementary information in different feature sets by sequential training of the feature parameters. Finally we try hypothesis-level score combination for DLMs after training separate models for each feature set. We use the perceptron algorithm [1, 2] for DLM training. Our experiments show that cascade parameter training may be an interesting approach that sometimes outperforms joint training.

This paper is organized as follows. Section 2 briefly explains DLMs and the feature sets used in this paper. Section 3 describes the feature combination approaches considered in this paper. Experiments and results are presented in Section 4. Discussion and conclusion are given in Section 5.

## 2. Discriminative Language Models

In this paper, we used $N$-best hypotheses obtained by the baseline ASR system as DLM training data. Since DLM is a feature-based language modeling approach, each candidate hypothesis in training data is mapped to a sparse $d$-dimensional feature vector, $\Phi(x, y)$, which is a function of the acoustic input, $x$, and the candidate hypothesis, $y$. We use $n$-grams as DLM features throughout the paper. Each element of the feature vector, $\Phi_1(x, y), \Phi_2(x, y), \cdots, \Phi_d(x, y)$, represents a particular $n$-gram feature. The number of unique $n$-grams in the DLM training data determines the size of the feature vector, $d$. $n$-gram features are defined as the number of times an $n$-gram is seen in the candidate hypothesis. An example word bigram feature is as follows:

$\Phi_i(x, y)$ = number of times "`the paper`" is seen in candidate hypothesis $y$ for acoustic input $x$

Any kind of information extracted from the candidate hypotheses can be easily encoded as DLM features. In this paper we used three different kinds of feature sets. The first one is word $n$-gram features. The second one is word class $n$-gram features. In this feature set the vocabulary words are clustered using linguistic or data-driven approaches and the $n$-gram features are defined in terms of the classes of consecutive words in the hypothesis. Here the main idea is to capture generalization of the data using a coarser feature set than word $n$-grams. Linguistic clustering uses the part-of-speech (PoS) tags of words. In addition to PoS tags, we also used a data-driven clustering algorithm [10] based on maximum mutual information to find word classes automatically. The third feature set is the acoustic features. This feature set is proposed in [9] to extend language modeling to incorporate acoustic state transitions and state durations. Here state and duration $n$-gram features are extracted from the clustered allophone state sequences of the acoustic input $x$ corresponding to the hypothesis $y$. Consider the below sequence for an $(x, y)$ pair where the IDs correspond to clustered allophone states [1]:

$$\cdots 1000, 1000, 4546, 4789, 1000, 1000, 4546 \cdots$$

An example 4-gram duration feature extracted from this state sequence is as follows:

[1] This example is taken from [9].

Copyright © 2011 ISCA

**Inputs:** Training examples $(x_i, r_i)$ for $i = 1 \ldots N$
**Initilization:** $\bar{\alpha}_0^N = (\alpha_0, 0, \ldots, 0)$
**Algorithm:**
For $t = 1 \ldots T$
$\bar{\alpha}_t^0 = \bar{\alpha}_{t-1}^N$
For $i = 1 \ldots N$
$y_i = \underset{y \in GEN(x_i)}{\text{argmax}} \langle \bar{\Phi}(x_i, y),\ \bar{\alpha}_t^{i-1} \rangle$
$\bar{\alpha}_t^i = \bar{\alpha}_t^{i-1} + \Phi(x_i, r_i) - \Phi(x_i, y_i)$
**Output:** Averaged parameters $\bar{\alpha} = \sum_{i,t} \bar{\alpha}_t^i / NT$

Figure 1: $\bar{\alpha}_t^i$ represents the feature parameters after the $t$'th pass on the $i$'th example. $r_i$ is the gold-standard hypothesis.

$\Phi_j(x, y)$ = the number of times "`4546 4789 1000`$_2$ `4546`" is seen in state sequence of $(x, y)$

Duration features incorporate consecutive occurrences of the same state in the feature definition, e.g., `1000`$_2$. The state features ignore the consecutive occurrences of the same state, i.e., `1000`$_2$ is replaced with `1000` in the feature example. See [9] for more details on acoustic features in DLMs.

In addition to $n$-grams, one important feature is the contribution of the baseline ASR system to DLM. This feature is defined as the "log-probability of $y$ in the lattice produced by the baseline recognizer for the acoustic input $x$" and is denoted by $\Phi_0(x, y)$. This feature is added to all the feature sets as the zeroth feature, i.e., $\bar{\Phi}(x, y) = [\Phi_0(x, y)\Phi(x, y)]$. $\bar{\Phi}(x, y)$ represents the full feature vector, including $\Phi_0(x, y)$.

After representing the training data in terms of features, the parameters associated with the features, $\bar{\alpha}$, are learned discriminatively. The best hypothesis under the $\bar{\alpha}$ model, $y^*$, maximizes the inner product of the feature and the parameter vectors, as given in Eq. 1. $GEN(x)$ is a function that enumerates a finite set of candidates for the input $x$. In our case it represents all the hypotheses in the $N$-best list for the acoustic input $x$. The values of $\bar{\alpha}$ are learned in training and the best hypothesis under this model is searched for in decoding.

$$y^* = \underset{y \in GEN(x)}{\text{argmax}} \langle \bar{\Phi}(x, y),\ \bar{\alpha} \rangle \quad (1)$$

We used a variant of the perceptron algorithm [2] (see Fig. 1) to train the feature parameters. The main idea of this perceptron algorithm is to penalize features associated with the current 1-best hypothesis, and to reward features associated with the gold-standard hypothesis (reference or lowest-WER hypothesis). Following the findings of [2], we use the lowest-WER hypothesis (oracle) as the gold-standard hypothesis, and $\alpha_0$ is not updated during perceptron training. Multiple models are trained with different $\alpha_0$, and a development set is used to select the best $\alpha_0$ and epoch number. Averaged perceptron parameters, $\bar{\alpha}$, are utilized in decoding development and test sets [1].

# 3. Feature Combination Approaches

This section describes three different feature combination approaches for DLMs.

## 3.1. Joint Parameter Estimation

The traditional approach when combining different kinds of features in DLMs is to concatenate the feature vectors associated with the same hypothesis and train the weights of these features jointly with discriminative training. Let $\Phi^{fs_1}(x, y)$ and $\Phi^{fs_2}(x, y)$ be the feature vectors belonging to the same hypothesis, $y$, for two different feature sets. Then the concatenated feature vector is as follows.

$$\bar{\Phi}^{fs_1+fs_2}(x, y) = [\Phi_0(x, y)\ \Phi^{fs_1}(x, y)\ \Phi^{fs_2}(x, y)]$$

The advantage of this approach is that it requires only a single DLM training phase to combine different feature sets. However, this approach can drastically increase the number of feature parameters that need to be estimated at the same time if each of the concatenated vectors introduces many features.

## 3.2. Cascade Parameter Estimation

The main idea in this approach is to combine different features by cascading several DLMs, each of which uses a single feature set. In this paper, the features are considered in a coarse-to-fine strategy, and feature parameters associated with each feature set are trained sequentially. According to the coarse-to-fine strategy, the initial model contains the coarsest and the final model contains the finest feature set. After training the initial DLM with the coarsest feature set, the score assigned to each hypothesis, $\langle \bar{\Phi}(x, y),\ \bar{\alpha} \rangle$, after modification, is included as one feature within the next model containing a finer feature set. This procedure continues until the finest feature set.

We convert the dot product scores from the previous model into posterior probabilities to use as features in the next model. Consider a coarse feature set $fs_1$. Let the $\bar{\alpha}^{fs_1}$ be the output of the perceptron training. We normalize each parameter by dividing by $\alpha_0^{fs_1}$, i.e. $\alpha_i^{fs_1}/\alpha_0^{fs_1}$. Note that this does not affect the decoding decision in Eq. 1. The posterior probabilities for the sentence hypotheses are calculated using softmax:

$$p_{\bar{\alpha}^{fs_1}}(y|x) = \frac{\exp(\langle \bar{\Phi}^{fs_1}(x, y), \bar{\alpha}^{fs_1} \rangle)}{\sum_{y' \in GEN(x)} \exp(\langle \bar{\Phi}^{fs_1}(x, y'), \bar{\alpha}^{fs_1} \rangle)} \quad (2)$$

Then the log-probability is used as the zeroth feature, $\Phi_0^{fs_1 \to fs_2}(x, y) = \log p_{\bar{\alpha}^{fs_1}}(y|x)$, in training the next DLM that contains a finer feature set $fs_2$. So the feature vectors of a hypothesis are represented as follows according to the cascade scheme $fs_1 \to fs_2 \to fs_3$:

$$\begin{aligned} \bar{\Phi}^{fs_1}(x, y) &= [\Phi_0(x, y)\ \Phi^{fs_1}(x, y)] \\ \bar{\Phi}^{fs_1 \to fs_2}(x, y) &= [\Phi_0^{fs_1 \to fs_2}(x, y)\ \Phi^{fs_2}(x, y)] \\ \bar{\Phi}^{fs_1 \to fs_2 \to fs_3}(x, y) &= [\Phi_0^{fs_1 \to fs_2 \to fs_3}(x, y)\ \Phi^{fs_3}(x, y)] \end{aligned}$$

Note that $\log p_{\bar{\alpha}^{fs_1}}(y|x)$ is $\langle \bar{\Phi}^{fs_1}(x, y),\ \bar{\alpha}^{fs_1} \rangle - \log Z$, where $Z$ is the denominator term in Eq. 2. As a result, cascade training corresponds to keeping the parameters of the previous model fixed and learning just a scaling factor ($\alpha_0^{fs_1 \to fs_2}$) together with the parameters of the new feature set. The advantage of this approach is that even if the total number of features increases dramatically due to a large number of feature sets, while training each model in the cascade, the number of features and parameters is limited to those of that single feature set. A possible disadvantage is that the cascaded models have to be trained sequentially and there is no clear optimal order for training. Separate DLMs with each feature set need to be trained sequentially. In reranking the development and test data, the same cascade scheme is applied.

## 3.3. Score Combination at Hypotheses Level

In the score combination approach, different DLMs are trained for each feature set and then the scores, $\langle \bar{\Phi}(x, y),\ \bar{\alpha} \rangle$, are combined at the hypotheses level. The best hypothesis under different feature sets is chosen as follows:

$$y^* = \underset{y \in GEN(x)}{\text{argmax}} \sum_i \lambda_{fs_i} \times \langle \bar{\Phi}^{fs_i}(x, y),\ \bar{\alpha}^{fs_i} \rangle \quad (3)$$

This approach requires several separately trained DLM models, which can be done in parallel. The weights, $\lambda_{fs_i}$, are optimized using the simplex algorithm to minimize the word error rate on the development set. We use the simplex algorithm implementation in the SRILM toolkit [11].

# 4. Results

## 4.1. ASR System and DLM Data

The DLM experiments are performed on an English Broadcast News task. The baseline system is based on the 2007 IBM GALE speech transcription system [12]. The speaker independent acoustic model was trained on 430h of Broadcast News audio. The generative language model training text consists of a total of 400M words from six different sources. The vocabulary is 80k words and the baseline language model is a linear interpolation of word 4-gram models, one for each corpus. Dev04 (2.1h) and the test set rt04 (4h) are decoded with the baseline language model.

As mentioned in Section 2, DLM requires $N$-best hypotheses as the training data. We decode the acoustic model training data (430h) to obtain the candidate outputs for the $GEN(x)$ function. Since the reference transcriptions of the acoustic model training data is used in generative language model training, we applied 24-fold cross validation while building the language models for DLM and decoding the acoustic data. The acoustic training data are broken into 24 folds, and the utterances in each fold are decoded using the baseline acoustic model (trained on all of the utterances) and a 4-gram language model trained on the other 23 folds and the other text resources to alleviate over-training of the language models. Acoustic model training is not typically controlled in the same manner.

## 4.2. Feature Combination Results

We first performed the DLM experiments using only single feature sets. We experimented with unigram and bigram features for word, word class and acoustic features. The perceptron algorithm was used to train multiple models with different $\alpha_0$, and the development set dev04 was used to select the best $\alpha_0$ and epoch number. The test set rt04 was a truly blind set and the results were not obtained until after all the design decisions had been made and all the models had been trained.

The results with the best models on dev04 are reported in Table 1. Word features are denoted by W(1,2), automatically induced word class features by C(1,2), PoS features by P(1,2), state features by S(1,2) and duration features by D(1,2) . #feats and #actfeats show the number of features extracted from the training data and the number of non-zero features after parameter estimation, respectively.

As a standard recipe for all the feature sets, we initially trained 30 different models with $\alpha_0$ values from 0 to 29 and picked the best model based on the dev set. This recipe works for word and word class features, but for state and duration features, we obtained non-optimal, mis-trained models Sm(1,2) and Dm(1,2). For example, for Dm(1,2), even though there was an improvement on dev04 (0.33%), there was hardly any improvement on the test set rt04 (0.05%). In the perceptron algorithm for DLM (see Fig. 1), there is no explicit step size, but $1/\alpha_0$ works like a step size in the update equations. Since state or duration features are extracted at the frame level, each hypothesis is represented by many more features than word features. It makes sense to use a smaller step size or larger $\alpha_0$. We therefore trained 200 models with $\alpha_0$ values from 0 to 199. The best models did not change for the word and word class feature sets, but the gains obtained with acoustic features were increased. For example, D(1,2) ($\alpha_0 = 99$) is better than baseline by 0.56% on dev04, compared with 0.33% for Dm(1,2). On the test set, D(1,2) is now better than baseline by 0.21%.

There are two alternatives for word class features, C(1,2) and P(1,2) and two alternatives for acoustic-based features, S(1,2) and D(1,2). One design decision was to select one alternative in each type for further experiments combining different types of features. Based on the dev set results, automatically induced class features C(1,2) yield slightly better gains compared to PoS features P(1,2). Therefore C(1,2) was chosen for further feature combination experiments. For the acoustic feature sets, duration features D(1,2) outperform state features S(1,2) on the dev set. We found out later that on the test set, S(1,2) was actually better than D(1,2). However, since our design decisions were all based on the dev set, we chose D(1,2) to represent the acoustic feature set for feature combination experiments.

Table 1: DLM results with lexical, word class and acoustic features on development (dev04) and test (rt04) sets.

| | #feats ($\times 10^3$) | #actfeats ($\times 10^3$) | WER (dev04) | ΔWER | WER (rt04) | ΔWER |
|---|---|---|---|---|---|---|
| Baseline ASR | - | - | 20.60 | - | 17.70 | - |
| DLM | | | | | | |
| W(1,2) | 2198.3 | 334.7 | 20.28 | 0.32 | 17.50 | 0.20 |
| C(1,2) | 2.7 | 2.7 | 20.30 | 0.30 | 17.48 | 0.22 |
| P(1,2) | 1.6 | 1.4 | 20.36 | 0.24 | 17.60 | 0.10 |
| Sm(1,2) $\alpha_0 < 30$ | 39.8 | 37.1 | 20.36 | 0.24 | 17.52 | 0.18 |
| Dm(1,2) $\alpha_0 < 30$ | 1791.6 | 712.2 | 20.27 | 0.33 | 17.65 | 0.05 |
| S(1,2) | 39.8 | 36.1 | 20.20 | 0.40 | 17.39 | 0.31 |
| D(1,2) | 1791.6 | 667.4 | 20.04 | 0.56 | 17.49 | 0.21 |

We started feature combination experiments using dual feature sets. We combined class features with word features, class features with duration features and duration features with word features using the feature combination approaches explained in Section 3. The ordering of the combined features is only important for the cascade approach. The feature ordering is selected as C(1,2) $\rightarrow$ D(1,2) $\rightarrow$ W(1,2) by looking at the percentage of the active features seen (hit rate) in the dev set. Class features have 100%, duration features have 97% and word features have 81% hit rates. The results with dual feature combination experiments on rt04 are given in Table 2. The second column shows the feature combination approach. The cascade order is shown with a right arrow in the first column.

When combining class and word feature sets, joint and cascade approaches yield almost the same result. The result with the cascade approach is significantly [2] ( $p < 0.001$) better than the baseline WER. No additive gain is obtained with class and word feature sets in score combination, i.e. the combined model did not outperform the individual models.

When combining class and duration features, the cascade approach, C(1,2) $\rightarrow$ D(1,2), yields the best result by reducing the WER from 17.70% to 17.25% (0.45% improvement, significant at $p < 0.001$). The second best performance is obtained with joint training (significant at $p < 0.001$). Class and duration are completely different information sources and the gains obtained by combining these two features in joint and cascade approaches are totally additive. However, the score combination approach does not give any improvement on top of separately trained models. The interpolation weights in score combination is optimized to reduce the WER in the dev set, not the training set. Since most of the errors on dev set are fixed by D(1,2) alone, D(1,2) gets a much higher interpolation weight than C(1,2) ($\lambda_c$=0.06 and $\lambda_d$=0.24) [3].

When combining duration and word features, joint parameter estimation yields the best result, 0.35% (significant at $p < 0.001$) gain over the baseline WER. Cascade and score combination approaches do not give any additive improvements on top of duration features.

The results of the triple feature set combination experiments on rt04 are given in Table 3. The cascade approach yields 0.08% higher gain (not statistically significant) compared to joint training, and it gives our best result on rt04 by reducing the WER from 17.70% to 17.21% (significant at $p < 0.001$). Joint train-

[2] Statistical significance is measured by the NIST MAPSSWE test.

[3] This is log-linear combination and $\lambda_{fs_i}$'s do not sum to 1.

Table 2: DLM results after dual feature combination on rt04.

| | Feature Combination | WER (rt04) | ΔWER |
|---|---|---|---|
| Baseline ASR | - | 17.70 | - |
| DLM (two feature sets) | | | |
| C(1,2) + W(1,2) | joint | 17.43 | 0.27 |
| C(1,2)→W(1,2) | cascade | 17.44 | 0.26 |
| $\lambda_c$C(1,2)+$\lambda_w$W(1,2) | score | 17.49 | 0.21 |
| C(1,2) + D(1,2) | joint | 17.28 | 0.42 |
| C(1,2)→D(1,2) | cascade | 17.25 | 0.45 |
| $\lambda_c$C(1,2)+$\lambda_d$D(1,2) | score | 17.48 | 0.22 |
| D(1,2) + W(1,2) | joint | 17.35 | 0.35 |
| D(1,2)→W(1,2) | cascade | 17.47 | 0.23 |
| $\lambda_d$D(1,2)+$\lambda_w$W(1,2) | score | 17.48 | 0.22 |

ing with three feature sets did not improve over using two feature sets. Similar to the results obtained on dual feature combination experiments, score combination does not give any gains on top of duration features.

## 5. Conclusion and Discussion

In this paper we investigated three different feature combination approaches for DLMs using features such as words, word classes, and state durations. We first tried joint parameter estimation which is the traditional approach in DLMs. Then we proposed the cascade training and model score combination.

One important lesson we learned about using the perceptron algorithm for training DLMs is the importance of considering a larger range of $\alpha_0$ values, particularly with dense feature sets such as state and duration features where a smaller step size for updates is better.

In the traditional approach for feature combination, the parameters for all features are jointly trained without regard to the nature of the type of features, such as the density of the features (e.g. there are many more state and duration features than word features in an utterance) or how likely the features in the training data will re-appear in the test data (e.g. word-class features are more likely than duration features). Nevertheless, in our experiments, joint training consistently gave competitive results. Joint training allows the most flexibility for model parameters to be adjusted, and theoretically should give the best performance on the training data. However, the model may not generalize to give the best performance on the test data. In addition, there are scalability issues to be considered, e.g. when there are tens or hundreds of feature sets to be combined.

The proposed cascade approach trains on each feature set in sequence, while freezing the previously trained parameters of prior feature sets. Thus compared with joint training, the parameters do not have as much freedom to be adjusted, and the performance will be suboptimal on the training data. However, empirically it gave the overall best test results for dual combination C(1,2)→D(1,2) and triple combination C(1,2)→D(1,2)→W(1,2), but was sometimes worse than joint training, e.g. D(1,2)→W(1,2). Compared to a single large model in joint training, the cascade approach can be thought of as breaking up the model into multiple smaller models, and can thus be more scalable with the number of feature sets. However it requires $k$ different sequential DLM model training phases, where $k$ is the number of feature sets to be combined. It is also not clear how to determine the optimal order of the cascade, and there are $k!$ possible orders. In this paper we tried a coarse-to-fine strategy, but different strategies are also possible. One possible reason that the order C(1,2)→D(1,2) works so well could be that the first model C(1,2) fixes errors using features likely to re-appear in the test data (since C(1,2) has 100% hit rate), and lets D(1,2) try to fix the remaining errors. We also tried D(1,2)→C(1,2) which does not work as well (0.34% improvement vs. 0.45%). Perhaps D(1,2), with its large number of features, is able fix many more errors in the training data and leave fewer errors for C(1,2) to learn from. On the test data, the D(1,2) features may not generalize well since it has a lower hit rate. Overall the cascade approach seems interesting and warrants further study.

Table 3: DLM results after triple feature combination on rt04.

| | Feature Combination | WER (rt04) | ΔWER |
|---|---|---|---|
| Baseline ASR | - | 17.70 | - |
| DLM (three feature sets) | | | |
| C(1,2)+D(1,2)+W(1,2) | joint | 17.29 | 0.41 |
| C(1,2)→D(1,2)→W(1,2) | cascade | 17.21 | 0.49 |
| $\lambda_c$C(1,2)+$\lambda_w$ W(1,2)+$\lambda_d$ D(1,2) | score | 17.48 | 0.22 |

Finally, with model score combination, the parameters have even less freedom to be adjusted since only two weights are to be determined after the individual models have been trained. This model is extremely scalable because models with different feature sets can be trained separately in parallel. In our experiments, score combination of the three models does not do any better than using duration features alone. Because the duration features are fixing most of the errors on the dev set, the interpolation weights of other feature sets are very small compared to that of the duration features. This result could be an artifact of our dev set and experimental setup. Using a different dev set not used to tune the individual models can be helpful.

Most of the improvements we obtained came from a combination of word class and duration features because these are very different feature types. A further area of research would be to find more disparate features to be combined in DLMs.

## 6. References


[1] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Proceedings of EMNLP*, Philadelphia, PA, USA, 2002, pp. 1 – 8.

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

[3] T. Oba, T. Hori, and A. Nakamura, "Round-robin discrimination model for reranking ASR hypotheses," in *Proceedings of Interspeech*, Makuhari, Japan, 2010, pp. 2446 – 2449.

[4] M. Collins, M. Saraçlar, and B. Roark, "Discriminative syntactic language modeling for speech recognition," in *Proceedings of ACL*, Ann Arbor, MI, USA, 2005, pp. 507 – 514.

[5] E. Arısoy, M. Saraçlar, B. Roark, and I. Shafran, "Syntactic and sub-lexical features for Turkish discriminative language models," in *Proceedings of ICASSP*, Dallas, Texas, USA, 2010, pp. 5538 – 5541.

[6] I. Shafran and K. Hall, "Corrective models for speech recognition of inflected languages," in *Proceedings of EMNLP*, Sydney, Australia, 2006, pp. 390 – 398.

[7] E. Arısoy, B. Roark, I. Shafran, and M. Saraçlar, "Discriminative n-gram language modeling for Turkish," in *Proceedings of Interspeech*, Brisbane, Australia, 2008, pp. 825 – 828.

[8] N. Singh-Miller and M. Collins, "Trigger-based language modeling using a loss-sensitive perceptron algorithm," in *Proceedings of ICASSP*, Honolulu, Hawaii, USA, 2007, pp. 25 – 28.

[9] M. Lehr and I. Shafran, "Discriminatively estimated joint acoustic, duration and language model for speech recognition," in *Proceedings of ICASSP*, Dallas, Texas, USA, 2010, pp. 5542 – 5545.

[10] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467 – 479, December 1992.

[11] A. Stolcke, "SRILM – An extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 2, Denver, 2002, pp. 901 – 904.

[12] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1596 – 1608, 2006.