

A Decade of Discriminative Language Modeling for Automatic Speech Recognition

Murat Saraclar¹(✉), Erinc Dikici¹, and Ebru Arisoy²

¹ Electrical and Electronics Engineering Department, Bogazici University,
34342 Bebek, Istanbul, Turkey

{murat.saraclar,erinc.dikici}@boun.edu.tr

<http://busim.ee.boun.edu.tr/~speech>

² Electrical and Electronics Engineering Department, MEF University,
34396 Sariyer, Istanbul, Turkey

ebruarisoy.saraclar@mef.edu.tr

Abstract. This paper summarizes the research on discriminative language modeling focusing on its application to automatic speech recognition (ASR). A discriminative language model (DLM) is typically a linear or log-linear model consisting of a weight vector associated with a feature vector representation of a sentence. This flexible representation can include linguistically and statistically motivated features that incorporate morphological and syntactic information. At test time, DLMs are used to rerank the output of an ASR system, represented as an N-best list or lattice. During training, both negative and positive examples are used with the aim of directly optimizing the error rate. Various machine learning methods, including the structured perceptron, large margin methods and maximum regularized conditional log-likelihood, have been used for estimating the parameters of DLMs. Typically positive examples for DLM training come from the manual transcriptions of acoustic data while the negative examples are obtained by processing the same acoustic data with an ASR system. Recent research generalizes DLM training by either using automatic transcriptions for the positive examples or simulating the negative examples.

Keywords: Automatic speech recognition · Discriminative training · Language modeling

1 Introduction

Discriminative language models (DLMs) have been part of ASR systems for over a decade and have been applied to tasks such as LVCSR [29], utterance and call classification [33], automatic transcription and retrieval of broadcast news [2], along with their use in parsing [37] and machine translation [24].

DLMs are typically formulated as a linear [8], log-linear [29] or an exponential model [41]. In this paper, we follow the linear model framework that reranks alternative hypotheses produced by a function $\mathbf{GEN}(\cdot)$ using the inner product of a feature vector Φ and a weight vector \mathbf{w} as a score.

The use of a feature representation allows DLMs to integrate many different sources (n -gram, morphological, syntactic, and semantic information) into a single mathematical structure. The details of various feature sets used in DLMs are given in Sect. 2.

The objective of DLM training is to estimate the model vector \mathbf{w} by optimizing an objective function directly related to the word error rate (WER). DLM training algorithms can be categorized as classification and reranking approaches. While the classification approaches aim to improve the score of the least errorful hypothesis, the reranking approaches aim to adjust the scores so as to match the ordering of the hypotheses with respect to error rate. The perceptron algorithm is a popular method which formulates discriminative modeling as a structured prediction task [29]. It has also been adapted to reranking [37], which gives higher accuracies at the cost of a longer training duration [13]. Modifications of the perceptron optimization criterion have been proposed to make it more directly related to the WER [32, 38]. Large margin methods such as the support vector machines (SVM) and the margin-infused relaxed algorithm (MIRA) [10] are among the other methods that are used to train a DLM. Classification and reranking variants of the SVM [16] have been used for tasks such as ASR [42], lexical disambiguation [4], parsing and machine translation [7]. The MIRA has also been applied to statistical machine translation [40] and parsing [27]. The details of DLM training algorithms are given in Sect. 3.

Traditionally, DLMs are trained in a supervised setting where the training data is composed of pairs of spoken utterances and corresponding manual transcriptions. DLM training requires a set of alternative hypotheses produced by a function $\mathbf{GEN}(\cdot)$. In the supervised setting this represents the ASR system itself and produces lattices or n -best lists corresponding to the spoken utterances.

In cases where the data is not sufficient to train a DLM in a supervised manner, semi-supervised training techniques are applied. Here, the function $\mathbf{GEN}(\cdot)$ generates alternative hypotheses from the transcriptions using a confusion model (CM). In [17] and [22], the CM is constructed using phonetic similarities estimated from an acoustic model. In [5] and [12], CMs based on words and different sub-word units are compared. Other approaches [25, 39] make use of a machine translation system to learn phrase similarities as if the ASR outputs were translations of a source reference transcription. On the other hand, [41] finds competing words (cohorts) directly from the ASR outputs of untranscribed speech to form the CM. A comparison of these three main approaches is given in [31].

Unsupervised training of discriminative language models is a more recent area of interest. In this approach, DLMs are trained using examples without any reference. Like the supervised setting, the $\mathbf{GEN}(\cdot)$ function is an ASR system producing the alternative hypotheses, however the reference is replaced by an automatically derived hypothesis. In [21] the reference is chosen according to the Minimum Bayes Risk criterion. Alternatively in [18], a weak acoustic model is used for generating alternative hypotheses and a stronger acoustic model is used to automatically derive the reference. Further details are given in Sect. 4.

2 Features

In DLMs, $\Phi_d(x, y)$ represents a particular sentence-level feature in the feature vector $\Phi(x, y)$ extracted from the candidate hypothesis y for the utterance x . The most common feature type used in DLMs is the n -gram features [29] defined as the number of times an n -gram is seen in the candidate hypothesis y . In this case, the features do not explicitly depend on x , so the notation can be simplified to $\Phi(y)$. For instance if the candidate hypothesis is “*in the paper*”, the feature corresponding to the bigram “*the paper*” is defined as follows:

$$\Phi_d(y) = \text{number of times “the paper” is seen in candidate hypothesis } y = 1$$

Using word n -grams as features in discriminative language modeling outperformed the traditional generative n -gram language modeling approach for English [29] as well as Turkish [2] ASR systems.

This section reviews various feature sets used in DLMs. The features discussed in this section are linguistic, statistically derived, and acoustic features.

2.1 Linguistic Features

Linguistic features used in DLMs involve morphological and syntactic features as well as features extracted from the topic or the conversation context.

Morphology is an important information source for feature-based language modeling, especially for agglutinative or highly inflectional languages both in generative [20, 35] and discriminative language modeling [3, 32, 36] frameworks. In the DLM framework, the words in the candidate hypotheses are parsed with a morphological parser and the features are extracted from these parses. Possible choices of DLM features are n -grams extracted from stem, ending and morphological tag sequences. In stem and ending features, the stem is extracted from the morphological decomposition and the remaining part of the word is taken as the ending. After converting all the hypotheses into stem and ending sequences, either in surface [3] or lexical form [32], the n -gram features are extracted in the same way as words, as if the stem and endings were words. The same procedure can also be applied to morphological tag sequences. Morphological n -gram features in DLMs have been shown to yield significant improvements on top of generative n -gram language models both for Czech [36], a highly inflectional language, and Turkish [3, 32], an agglutinative language, ASR systems.

Syntax, the rules of sentence formation, is also an important information source for language modeling, especially for capturing long distance dependencies in addition to the previous $n-1$ words. Therefore, syntactic information has been incorporated into conventional generative language models [6, 28] and feature-based conditional exponential language models [19, 30]. In the DLM framework, the candidate hypotheses are parsed using a syntactic parser and the syntactic features are extracted from these syntactic analyses. For instance in [9], each candidate hypothesis received a Part-of-Speech (PoS) tag annotation and a parse tree annotation from a syntactic parser and the syntactic features were extracted from PoS tag and shallow parse tag sequences, again using the n -gram feature

approach. Also [9] proposed syntactic features that make use of the full parse tree, such as context-free rule features and head-to-head dependency features. An example context-free rule feature is given as follows:

$\Phi_d(y)$ = number of times the context-free rule $S \rightarrow NP VP$ is seen in the parse tree of the candidate hypothesis y

Head-to-head features were also defined in the same way by using the lexical heads with their constituents and head-to-head dependencies. Among these proposed features, PoS tag n -grams yielded the most of the gain on top of the word n -gram features on an English ASR system [9].

In [3], these syntactic features were adopted for Turkish using the dependency parser output for the candidate hypotheses. Moreover, PoS speech tag n -gram features yielded the most of the gain, consistent with the findings of [9], on a Turkish Broadcast News (BN) transcription system.

Wider context in the form of trigger-based features that identify the re-occurrence of words within a conversation were utilized in the DLM framework [38]. Additionally, semantic context was incorporated as an additional information source using features based on automatically annotated topics for each utterance [3].

2.2 Statistically Derived Features

In addition to linguistic tools, useful information for DLMs can be derived statistically. For example, instead of using PoS tags for words, automatically derived word classes are used as DLM features in [3] resulting in similar gains.

Similar to the grammatically-driven sub-word units, such as morphemes or morpheme groupings (stems and endings), statistical sub-lexical units called morphs [11] can be obtained using statistical approaches. The n -gram DLM features can be directly extracted from the statistical morph sequences. Since, statistical morphs do not convey explicit linguistic information, features similar to PoS tags and trigger dependencies are obtained from morphs with statistical approaches [3]. For instance, automatically clustering morphs into syntactic categories and considering the cluster associated with a particular morph as the tag of that morph provide morpho-syntactic features that resemble the PoS tag features for words. It was shown that morph unigram features yield significant gains on top of the generative n -gram language model and automatic morph clusters give a significant additive gain on top of morph unigram features on a Turkish BN transcription system [3].

2.3 Acoustic Features

Acoustic features in DLMs incorporate acoustic state transitions and state durations in language modeling [23]. State and duration n -gram features are extracted from the clustered allophone state sequence obtained by the alignment of the hypothesis y to the acoustic input x , in contrast to extracting features directly

from the hypothesis y . Considering the following clustered allophone state IDs “... 1000, 1000, 4546, 4789, 1000, 1000, 4546 ...” as a label sequence, state n -gram features are extracted in a similar way with word n -gram features. Duration features also take into account the consecutive occurrences of the same state in the feature definition. An example 4-gram duration feature extracted from this state sequence is as follows:

$\Phi_d(x, y)$ = the number of times “4546 4789 1000₂ 4546” is seen in state sequence of (x, y)

State and duration n -gram features together with word n -gram features were shown to give a significant gain on a GALE Arabic transcription task [23]. Additionally, combining word cluster features with duration features yielded a significant gain on top of the generative language model on an English BN system [1].

3 Algorithms

In this section we first explain how the DLM is used at test time and then review the popular DLM training algorithms.

In the testing phase, the estimated model vector \mathbf{w} is used to reweight the ASR hypotheses of a test utterance x . The final output is the hypothesis with the highest evaluation score: $y^* = \operatorname{argmax}_{y \in \mathbf{GEN}(\cdot)} \left\{ w_0 \log P(y|x) + \langle \mathbf{w}, \Phi(y) \rangle \right\}$. Here, $\log P(y|x)$ is the recognition score assigned to y by the baseline recognizer for x , and w_0 is a scaling factor optimized on a held-out set. The overall system performance is represented by the WER of all y^* s.

Discriminative language modeling for ASR can be viewed as a structured prediction or a reranking task. Structured prediction is a classification type of approach in which the aim is to pick the most accurate example (hypothesis) in the N -best list, in terms of the number of word errors (WE) with respect to the reference. Generally, this means training with one positive example against a representative or collection of negative examples. The reranking approach, on the other hand, learns from pairwise relationships between the examples. In this section, after reviewing the traditional training algorithms for DLM, namely the structured perceptron and global conditional log-linear model (GCLM), we will summarize some of the recently proposed algorithms for both structured prediction and reranking tasks.

Structured Perceptron (Per) [29] is an adaptation of the canonical perceptron algorithm for solving structured prediction problems. For each utterance in the training set $x_i, i = 1..I$, Per uses two hypotheses for training: y_i is the *oracle* hypothesis which has the least WE, and z_i is the *current-best* hypothesis which yields the highest inner product score, $\langle \mathbf{w}, \Phi \rangle$, under the current model \mathbf{w} . Taking into account the fact that the oracle needs to have the highest inner product score in order to minimize the overall WER, the model weights are updated by favoring the features which occur in y_i and penalizing the ones which occur in z_i .

Global Conditional Log-Linear Model (GCLM) [29] aims to maximize the conditional log-likelihood of the training data under the parameters \mathbf{w} , given by

$$F(\mathbf{w}) = \sum_i \log p_{\mathbf{w}}(y_i|x_i) = \sum_i \log \frac{e^{\langle \mathbf{w}, \Phi(y_i) \rangle}}{\sum_{y \in \text{GEN}(\cdot)} e^{\langle \mathbf{w}, \Phi(y) \rangle}}. \quad (1)$$

The numerator can be thought of as the score of the correct hypothesis while the denominator is a sum of the scores of all hypotheses. F is a convex function so the optimal parameters can be found using a simple gradient update $\mathbf{w} = \mathbf{w} + \eta \nabla F$, where

$$\nabla F = \Phi(y_i) - \sum_{y \in \text{GEN}(\cdot)} p_{\mathbf{w}}(y|x_i) \Phi(y). \quad (2)$$

Thus, GCLM considers all alternative hypotheses in the parameter estimation, while the perceptron algorithm compares only the reference hypothesis and the current best hypothesis under the current model. The perceptron algorithm is typically used for feature selection and model initialization, and from that starting point, GCLM training with a zero mean Gaussian prior on the model parameters has been shown to further improve the model [29].

Ranking Perceptron (PerRank) [37] also considers all alternative hypotheses by comparing each and every pair of hypotheses a and b with the intention that if a has fewer word errors, it must have an inner product score significantly higher than that of b .

Figure 1 shows pseudocodes of the Per and PerRank algorithms as applied in this study. Both algorithms make several passes (T) over the data ($\{1 \leq i \leq I\}$) and in the end, the model weights obtained at each update step are averaged for robustness. The significance multiplier $g(\cdot, \cdot)$ in the update rule is implied by the optimization criterion. Defining this parameter as the edit distance between the two hypotheses leads to a word error rate sensitive update rule [32]. Some other parameters such as the margin constraint multiplier (τ), learning (η) and decay (γ) rates facilitate the convergence of the optimization procedure, and are determined by grid search on a held-out set. For more information on the selection of these parameters, the reader is referred to [12].

<pre> w = 0, w_{sum} = 0 for $t = 1 \dots T$, $i = 1 \dots I$ do $z_i = \operatorname{argmax}_{z \in \text{GEN}(\cdot)} \langle \mathbf{w}, \Phi(z) \rangle$ w += $g(y_i, z_i)(\Phi(y_i) - \Phi(z_i))$ w_{sum} = w_{sum} + w return w_{avg} = w_{sum} / (IT) </pre>	<pre> w = 0, w_{sum} = 0 for $t = 1 \dots T$ do for $i = 1 \dots I$ do for $(a, b) \in \text{GEN}(\cdot)$ do if $r_a \succ r_b$ & $\langle \mathbf{w}, \Phi(a) - \Phi(b) \rangle < \tau g(a, b)$ then w += $\eta g(a, b)(\Phi(a) - \Phi(b))$ w_{sum} = w_{sum} + w $\eta = \eta \cdot \gamma$ return w_{avg} = w_{sum} / (IT) </pre>
---	--

Fig. 1. Per and PerRank algorithms

Margin Infused Relaxed Algorithm (MIRA) [10] is an algorithm which trains a model (so called the *prototype*) for each class such that the inner product score of an instance with its class prototype, $\langle \mathbf{w}_{c_i}, \Phi \rangle$, is higher than the score with any other class prototype. For a two-class problem with $c_i \in \{\pm 1\}$, the binary MIRA iteratively updates a single prototype \mathbf{w} , just like the perceptron. The update rule is $\mathbf{w} = \mathbf{w} + \tau_i c_i (\Phi(y_i) - \Phi(z_i))$, where, unlike the perceptron, the learning rates τ_i are hypothesis-specific, and are found by solving a quadratic constrained optimization problem. More information on the application of the MIRA algorithm can be found in [13].

Ranking MIRA (MIRArank) [13] follows a similar procedure as in PerRank by updating for each pair of hypotheses that satisfy the margin criterion.

Support Vector Machine (SVM) is a binary linear classifier which aims to find a separating hyperplane that maximizes the margin between the nearest samples of two classes. The constrained optimization problem which covers all training examples j is defined as $\min_{\mathbf{w}} \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_j \xi_j$ subject to $c_j \langle \mathbf{w}, \Phi(y_j) \rangle \geq 1 - \xi_j$, where $\xi_j \geq 0$ are the slack variables for violations of the margin constraints and C is a user-defined trade-off parameter for its smoothness. The labeling of training examples in an SVM setup is not straightforward. In our implementation, the positive class is composed of the hypotheses having the lowest error rate of their N -best list.

Ranking SVM (SVMrank) [16] is a modification of the classical SVM setup to handle the reranking problem. Here the optimization can be viewed as an SVM classification problem on pairwise difference vectors, $\Phi(a) - \Phi(b)$. In that sense, the algorithm tries to find a large margin linear function which minimizes the number of pairs of training examples that need to be swapped to achieve the desired ranking.

4 Training Approaches

The training approaches that will be explained in this section are evaluated on a broadcast news transcription task. Our Turkish Broadcast News Speech and Transcripts Database [2, 34] is a collection of around 195 h of Turkish TV and radio channel recordings that are manually transcribed. We use 188 h of this collection for training, and about 3 h each as the held-out (parameter optimization) and test data.

The ASR hypotheses are organized in 50-best lists and represented in morphs as the language modeling unit. There are around 100k sentences and 46k unique morphs in our data set. The feature vector of the linear model, Φ , consists of morph unigram counts and therefore it is high dimensional but sparse. More information on our baseline system can be found in [2]. On the test set, the generative baseline WER is 22.4% and the corresponding oracle rate is 13.9%.

4.1 Supervised Training

The standard way to train a DLM is to use the supervised approach, where all acoustic training data are manually transcribed. In other words, the references

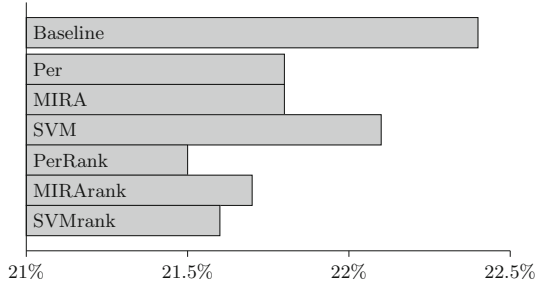


Fig. 2. Supervised training WER (%) for different training algorithms

of the ASR N-best hypotheses must be known beforehand. This allows us to determine their target ranks, which is the key factor in training.

We first examine the performance of supervised training, where DLM makes use of the 188 h of transcribed training data. Figure 2 shows the test set WERs of the six algorithms mentioned in Sect. 3, namely the classification and reranking versions of the perceptron, MIRA, and SVM. More information on how the algorithms are applied can be found in [13].

Figure 2 shows that the ranking variants of all three algorithms perform better than their classification variants, and that PerRank provides the lowest WER, with an improvement of around 1 % over the test baseline. SVMrank comes next, followed by MIRArank. We also see that the accuracies of Per and MIRA are very similar.

4.2 Semi-supervised Training

It is possible to increase the amount of training data by generating new artificial examples from some source text through confusion modeling. This approach as a whole is named semi-supervised training.

A confusion model (CM) is a model which represents the confusions (errors) made by the ASR system. In our study, we train the CM on pairs of the available ASR N-best hypotheses and their reference transcriptions [5]. Each hypothesis is aligned to its reference, resulting in a list of pairs confused by the ASR system together with the probability of confusion.

The artificial examples are generated from a text source which do not need to be associated with any acoustic component, thus is easier to obtain than manually transcribed recordings. Application of the CM yields a lattice of possible hypotheses that could be output by the ASR system if that sentence were to be uttered. The most probable N paths can be used as training data for the DLM training, along with their associated source text as the reference.

4.3 Unsupervised Training

In the case of having a large amount of untranscribed acoustic data, the class/rank of the hypotheses cannot be determined as there is no reference to be

compared against. Unsupervised training approach helps DLM training in such a case.

The technique we employ in unsupervised training is to choose a *target output* sequence that will take over the missing reference, by observing the candidates in the N-best list. We explore three ways to generate or choose this word sequence. The first choice is to select the 1-best as the target output, and determining the WE of other hypotheses by aligning each to the 1-best. A second approach is to choose the target output by the Minimum Bayes Risk (MBR) formulation [14, 26], which basically aims to determine the hypothesis which is closest to all the others in the N-best list [21]. As a third choice, the MBR approach can also be used on segments of each hypothesis instead of the whole hypothesis, which is named Segmental MBR after [15].

The target output selection approach can be used either to train a DLM or a CM. We refer to these two cases as Unsupervised DLM and Unsupervised CM, which are analogous to the supervised and semi-supervised training, respectively.

4.4 Summary of Experiments on Training Approaches

Table 1 is a summary of possible training approaches explained so far and their test set performances. The second and third columns show what kind of data is used to train the DLM, and if applicable, the CM. Here, A stands for acoustic data which are passed through the ASR system to obtain real hypotheses and T stands for their reference transcriptions. In our experiments, the artificial hypotheses are derived from the manual transcriptions instead of some other source text to be able to compare them to the real ones. The PerRank, which has shown the best performance in Fig. 2, is used for training and the training data is divided into two equal pieces, denoted by the numerical subscripts (A_1, T_2 , etc.).

The first part of Table 1 presents the four basic scenarios whereas the second part shows their combinations. We see that the first three scenarios provide significant improvements over the baseline. This suggests that both the derived artificial hypotheses and the chosen target output are effective for training, if

Table 1. PerRank test WER (%) for different training scenarios.

Scenario	CM	DLM	
Supervised		$(AT)_2$	Baseline
Semi-Supervised	$(AT)_1$	T_2	
Unsupervised DLM		A_2	
Unsupervised CM	A_1	T_2	
Sup + Semi-Sup	$(AT)_1$	$(AT)_1 + T_2$	
Sup + Unsup DLM		$(AT)_1 + A_2$	
Unsup DLM + Unsup CM	A_1	$A_1 + T_2$	

not as effective as the real hypotheses and manual transcriptions. Unsupervised CM approach, on the other hand, does not provide a significant improvement.

The first two combination experiments suggest reusing the CM training data in DLM training, or combining transcribed and untranscribed data, and sets the WER below the 22.0 % line. The advantage of the third experiment is to be able to combine totally unmatched acoustic and textual sources, which still gives an improvement in WER.

5 Conclusion

In this paper we review a decade of discriminative language modeling and summarize the framework, training algorithms and possible training approaches. Discriminative language modeling outperforms the conventional approaches, partly due to the improved parameter estimates with discriminative training and partly due to using features that can reflect complex language characteristics, such as morphology, syntax and semantics. We present the classification and reranking variants of popular training algorithms in the literature, and discuss their advantages and disadvantages. There are several approaches of DLM training with respect to the availability of different data sources, and we investigate the supervised, semi-supervised and unsupervised cases. The results show that reranking techniques outperform classification techniques, and that it is possible to obtain improvements in WER even when the acoustic and text data are coming from different sources, without any manual transcriptions.

Acknowledgments. This research is supported in part by TUBITAK Project numbers 105E102, 109E142 and the Bogazici University Research Fund (BU-BAP) projects 07HA201D, 14A02D3 (D-7948).

References

1. Arisoy, E., Ramabhadran, B., Kuo, H.K.J.: Feature combination approaches for discriminative language models. In: Proceedings of Interspeech, Florence, Italy (2011)
2. Arisoy, E., Can, D., Parlak, S., Sak, H., Saraçlar, M.: Turkish broadcast news transcription and retrieval. *IEEE Trans. Audio Speech Lang. Process.* **17**(5), 874–883 (2009)
3. Arisoy, E., Saraçlar, M., Roark, B., Shafran, I.: Discriminative language modeling with linguistic and statistically derived features. *IEEE Trans. Audio Speech Lang. Process.* **20**(2), 540–550 (2012)
4. Bergsma, S., Lin, D., Schuurmans, D.: Improved natural language learning via variance-regularization support vector machines. In: Proceedings of CoNLL, pp. 172–181. CoNLL, Association for Computational Linguistics, Stroudsburg (2010)
5. Çelebi, A., Sak, H., Dikici, E., Saraçlar, M., Lehr, M., Prud’hommeaux, E., Xu, P., Glenn, N., Karakos, D., Khudanpur, S., Roark, B., Sagae, K., Shafran, I., Bikel, D., Callison-Burch, C., Cao, Y., Hall, K., Hasler, E., Koehn, P., Lopez, A., Post, M., Riley, D.: Semi-supervised discriminative language modeling for Turkish ASR. In: Proceedings of ICASSP, pp. 5025–5028 (2012)

6. Chelba, C., Jelinek, F.: Structured language modeling. *Comput. Speech Lang.* **14**(4), 283–332 (2000)
7. Cherry, C., Quirk, C.: Discriminative, syntactic language modeling through latent SVMs. In: *Proceedings of the 8th AMTA Conference, Hawaii*, pp. 65–74, October 2008
8. Collins, M.: Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In: *Proceedings of EMNLP*, pp. 1–8 (2002)
9. Collins, M., Roark, B., Saraçlar, M.: Discriminative syntactic language modeling for speech recognition. In: *ACL*, pp. 507–514 (2005)
10. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.* **3**, 951–991 (2003)
11. Creutz, M., Lagus, K.: Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. Technical report, Helsinki University of Technology, Palo Alto, CA, Publications in Computer and Information Science Report A81, March 2005
12. Dikici, E., Çelebi, A., Saraçlar, M.: Performance comparison of training algorithms for semi-supervised discriminative language modeling. In: *Proceedings of Interspeech, Portland, Oregon, September 2012*
13. Dikici, E., Semerci, M., Saraçlar, M., Alpaydın, E.: Classification and ranking approaches to discriminative language modeling for ASR. *IEEE Trans. Audio Speech Lang. Process.* **21**(2), 291–300 (2013)
14. Goel, V., Byrne, W.: Minimum bayes-risk automatic speech recognition. *Comput. Speech Lang.* **14**, 115–135 (2000)
15. Goel, V., Kumar, S., Byrne, W.: Segmental minimum bayes-risk ASR voting strategies. In: *Proceedings of Interspeech*, pp. 139–142 (2000)
16. Joachims, T.: Optimizing search engines using clickthrough data. In: *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 133–142 (2002)
17. Jyothi, P., Fosler-Lussier, E.: Discriminative language modeling using simulated ASR errors. In: *Proceedings of Interspeech*, pp. 1049–1052 (2010)
18. Jyothi, P., Johnson, L., Chelba, C., Strope, B.: Distributed discriminative language models for Google voice search. In: *Proceedings of ICASSP*, pp. 5017–5021 (2012)
19. Khudanpur, S., Wu, J.: Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. *Comput. Speech Lang.* **14**, 355–372 (2000)
20. Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., Stolcke, A.: Morphology-based language modeling for conversational Arabic speech recognition. *Comput. Speech Lang.* **20**(4), 589–608 (2006)
21. Kuo, H.K.J., Arisoy, E., Mangu, L., Saon, G.: Minimum bayes risk discriminative language models for Arabic speech recognition. In: *Proceedings of ASRU*, pp. 208–213 (2011)
22. Kurata, G., Sethy, A., Ramabhadran, B., Rastrow, A., Itoh, N., Nishimura, M.: Acoustically discriminative language model training with pseudo-hypothesis. *Speech Commun.* **54**(2), 219–228 (2012)
23. Lehr, M., Shafran, I.: Learning a discriminative weighted finite-state transducer for speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **19**(5), 1360–1367 (2011)
24. Li, Z., Khudanpur, S.: Large-scale discriminative n-gram language models for statistical machine translation. In: *Proceedings of the 8th AMTA Conference, Hawaii*, pp. 133–142, October 2008

25. Li, Z., Wang, Z., Khudanpur, S., Eisner, J.: Unsupervised discriminative language model training for machine translation using simulated confusion sets. In: *Coling 2010, Posters*, Beijing, China, pp. 656–664, August 2010
26. Mangu, L., Brill, E., Stolcke, A.: Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Comput. Speech Lang.* **14**, 373–400 (2000)
27. McDonald, R., Crammer, K., Pereira, F.: Online large-margin training of dependency parsers. In: *Proceedings of ACL*, pp. 91–98. ACL, Association for Computational Linguistics, Stroudsburg (2005)
28. Roark, B.: Probabilistic top-down parsing and language modeling. *Comput. Linguist.* **27**(2), 249–276 (2001)
29. Roark, B., Saraclar, M., Collins, M.: Discriminative n-gram language modeling. *Comput. Speech Lang.* **21**(2), 373–392 (2007)
30. Rosenfeld, R., Chen, S.F., Zhu, X.: Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. *Comput. Speech Lang.* **15**(1), 55–73 (2001)
31. Sagae, K., Lehr, M., Prud’hommeaux, E.T., Xu, P., Glenn, N., Karakos, D., Khudanpur, S., Roark, B., Saralar, M., Shafran, I., Bikel, D., Callison-Burch, C., Cao, Y., Hall, K., Hasler, E., Koehn, P., Lopez, A., Post, M., Riley, D.: Hallucinated N-best lists for discriminative language modeling. In: *Proceedings of ICASSP* (2012)
32. Sak, H., Saraclar, M., Gungor, T.: Morpholexical and discriminative language models for Turkish automatic speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **20**(8), 2341–2351 (2012)
33. Saraclar, M., Roark, B.: Joint discriminative language modeling and utterance classification. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 561–564, 18–23 March 2005
34. Saraclar, M.: Turkish broadcast news speech and transcripts LDC2012S06, Philadelphia, Linguistic Data Consortium, Web Download (2012)
35. Sarikaya, R., Affy, M., Deng, Y., Erdogan, H., Gao, Y.: Joint morphological-lexical language modeling for processing morphologically rich languages with application to dialectal Arabic. *IEEE Trans. Audio Speech Lang. Process.* **16**(7), 1330–1339 (2008)
36. Shafran, I., Hall, K.: Corrective models for speech recognition of inflected languages. In: *Proceedings of EMNLP*, Sydney, Australia, pp. 390–398 (2006)
37. Shen, L., Joshi, A.K.: Ranking and reranking with perceptron. *Mach. Learn.* **60**, 73–96 (2005)
38. Singh-Miller, N., Collins, C.: Trigger-based language modeling using a loss-sensitive perceptron algorithm. In: *Proceedings of ICASSP*, vol. 4, pp. IV-25–IV-28, April 2007
39. Tan, Q., Audhkhasi, K., Georgiou, P., Ettelaie, E., Narayanan, S.: Automatic speech recognition system channel modeling. In: *Proceedings of Interspeech*, pp. 2442–2445 (2010)
40. Watanabe, T., Suzuki, J., Tsukada, H., Isozaki, H.: Online large-margin training for statistical machine translation. In: *Proceedings of EMNLP-CoNLL*, pp. 764–773, June 2007
41. Xu, P., Karakos, D., Khudanpur, S.: Self-supervised discriminative training of statistical language models. In: *Proceedings of ASRU*, pp. 317–322 (2009)
42. Zhou, Z., Gao, J., Soong, F., Meng, H.: A comparative study of discriminative methods for reranking LVCSR n-best hypotheses in domain adaptation and generalization. In: *Proceedings of ICASSP*, pp. 141–144 (2006)

Speech and Computer

17th International Conference, SPECOM 2015, Athens,

Greece, September 20-24, 2015, Proceedings

Ronzhin, A.; Potapova, R.; Fakotakis, N. (Eds.)

2015, XVI, 506 p. 135 illus., Softcover

ISBN: 978-3-319-23131-0