Investigation of MT-based ASR Confusion Models for Semi-Supervised Discriminative Language Modeling

Erinç Dikici[1], *Emily Prud'hommeaux*[2], *Brian Roark*[3], *Murat Saraçlar*[1]

[1]Department of Electrical and Electronics Engineering, Boğaziçi University, Istanbul, Turkey
[2]Department of Computer Science, University of Rochester, Rochester, NY, USA
[3]Center for Spoken Language Understanding, Oregon Health & Science University, Portland, OR, USA
{erinc.dikici,murat.saraclar}@boun.edu.tr, emilypx@cs.rochester.edu, roarkbr@gmail.com

10.21437/Interspeech.2013-330

Abstract

Semi-supervised discriminative language modeling uses simulated N-best lists instead of real ASR outputs as its training examples. In this study we apply two techniques in which artificial examples are generated using a WFST and an MT system trained on pairs of reference text and ASR output. We compare the performance of these techniques with the structured prediction and ranking variants of the WER-sensitive perceptron algorithm, and contrast with the supervised case where real ASR outputs are given as input. Choosing Turkish statistical morphs as n-gram features, we analyze the similarities between the hypotheses of these three setups and the number of utilized features. We show that the MT-based system yields the lowest WER, not only because the examples generated by this technique are more effective, but also because the ranking perceptron generalizes better with this setup. When trained on a combination of artificial WFST and MT data, the structured perceptron performs as well on an unseen test set as it does when trained on real ASR output.

Index Terms: discriminative training, semi-supervised learning, language modeling, ranking perceptron

1. Introduction

As a final stage in automatic speech recognition (ASR), discriminative language modeling (DLM) aims to choose the most accurate transcription of a spoken utterance among candidate hypotheses returned by the recognizer, usually in the form of an N-best list. Supervised DLM training methods use these candidate hypotheses and their corresponding reference transcriptions as input examples. The downside of this approach is that a large amount of training data is needed to train a DLM, which can be a problem if sufficient in-domain speech data are not available or transcribing them manually is costly. Semi-supervised discriminative training removes this necessity by using simulated hypotheses instead of real ones, via a confusion model which models possible confusions made by the recognizer. Given some reference text, we can generate artificial training examples that resemble outputs of a real ASR system, using the confusion model.

Semi-supervised DLM training has recently been popular in the literature, and there are a number of approaches to construct an appropriate confusion model (CM) for this task. One of the approaches uses a weighted finite-state transducer (WFST) to represent the CM. An example is Kurata et al. [1, 2], where phoneme similarities estimated from an acoustic model are specified in the CM by a process called Pseudo-ASR. Jyothi et al. [3] follow a similar method by modeling the phonetic confusions with a WFST. Another approach makes use of a machine translation (MT) system to learn these confusions. For instance, Tan et al. [4] use a phrase-based MT system to learn error models between parallel corpora of ASR output and reference text represented by phonemes. Li et al. [5] use translation alternatives of source phrase sequences to simulate confusions that could be made by an MT system. In a third approach, Xu et al. [6] make use of the competing words (cohorts) which occur in the ASR outputs of untranscribed speech to train their CM. A comparison of these three approaches is given in Sagae et al. [7]. Although the authors use the same dataset (English n-gram features) for all experiments, the language unit they utilize for training different CMs is different (a phone-based WFST model and word phrase-based MT and cohort models).

The aim of this study is to investigate the use of MT-based CMs for generating artificial ASR hypotheses, and to compare them with the WFST-based methods. Unlike [7], we use a Turkish statistical morph based model for both of the techniques. Furthermore, we consider not just the first candidate but all of the N-best candidates for training the CM, which was the main difference between the MT and cohort conditions in [7]. Finally, we employ pre-trained generative language models, which were shown to aid in obtaining linguistically plausible word sequences [8]. In such a way, we will carefully control the comparison between these confusion modeling methods, under similar best-performing conditions.

Canonical DLM methods view the task as a structured prediction problem, where the goal is to pick the best hypothesis out of the N-best list. Another approach is the ranking methodology, where the list is reorganized such that better hypotheses are shifted to the top. The perceptron algorithm has been popularly applied in both forms in the literature [9, 10]. The ranking perceptron has been shown to outperform the structured perceptron for the supervised case [11], but in [12] where various versions of the algorithm were applied on a WFST-based CM setting, the differences between the algorithms were not as pronounced. The second objective of our study is to show how the two algorithms will behave under the MT-based CM setting. For training we use a version known as the WER-sensitive perceptron, first proposed in [13] and then applied in [8, 12].

The outline of the paper is as follows: In Section 2 we present the two methods for generating artificial hypotheses, based on WFST and MT CMs. Section 3 explains the algorithmic procedure to make use of this data in discriminative model training and how the results are evaluated. The data and experimental setup are given in Section 4 which are followed by the experimental results in Section 5. We conclude the paper with a discussion in Section 6 and a summary in Section 7.

2. Artificial hypothesis generation

In semi-supervised discriminative modeling, we use simulated hypotheses as training examples. These examples need to be formed in such a way that they resemble the outputs of a real ASR system. One way to do that is to first learn some kind of a confusion model which represents the variability in real outputs, and then apply this model on some reference text to generate examples with similar variability. In this study we use two different techniques to obtain such a confusion model, one is based on weighted finite-state transducers whereas the other is based on statistical machine translation. Note that in both cases, we train the confusion models by aligning all N-best hypotheses to the reference transcription.

2.1. WFST-based confusion modeling

WFST-based confusion modeling technique analyzes ASR N-best hypotheses to determine which language units are confused with which others. Choosing morphs as the language unit, we follow a similar hypothesis generation procedure as in [12].

The procedure starts with representing each N-best hypothesis by morphs and aligning this morph sequence to the corresponding known reference sequence using the Levenshtein (edit) distance. This alignment gives a list of morph pairs that are confused by the ASR. The probability of confusion for a specific morph pair can be computed as the frequency of their match-ups in the list. The confusion model, denoted by $\mathcal{CM}$, is constructed as a WFST, the paired morphs being its input-output pairs, and their probabilities being its weights. In implementation, to reduce computational costs, arcs having probability less than 0.01 are discarded. Once the CM is learned, generation of artificial hypotheses given an input string $\mathcal{W}$ is straightforward, and can be summarized with the following composition sequence:

$$N\text{-best}(\text{prune}(\mathcal{W} \circ \mathcal{L}_W \circ \mathcal{CM}) \circ \mathcal{G}_M)$$

First, $\mathcal{W}$ is converted into morphs by composing with the lexicon $\mathcal{L}_W$. Composing this result with $\mathcal{CM}$ gives the alternative hypotheses in the form of a graph. Depending on the length of the input string and the number of possible confusions available in $\mathcal{CM}$, the resulting confusion graph can be so large to prevent efficient processing, and some of the confusions may even be unfeasible or unmeaningful. In order to circumvent this we prune this graph to the most probable 1000-best paths, and then reweight its arcs by composing with a language model (LM) transducer, $\mathcal{G}_M$. Two different LMs are used in our implementation: the GEN-LM is estimated from Turkish newswire data collected from the Internet, represented by 5-grams with a vocabulary of 76K morphs, and the ASR-LM is derived from the ASR's real outputs, represented by 4-grams out of 40K morphs. Finally, N most probable paths are selected as the artificial hypotheses.

2.2. MT-based confusion modeling

We will compare the WFST-based confusion modeling technique with an approach that generates confusions using a statistical phrase-based machine translation framework. The MT system used here, Moses [14], requires a word-aligned bilingual parallel corpus of training data from which it builds the grammars used to translate from one language to the other. In our case, this parallel corpus consists of ASR N-best hypotheses and their reference sequences, both of which are segmented into morphs, just like the WFST-based setup. Because there is no variation in the order of morphs in such data, we perform morph alignment using the Levenshtein algorithm as in the WFST setup, rather than using a word alignment package such as Giza++ [15].

These morph-level alignments serve as training data for Moses, which extracts the phrase grammar, tunes the weights of the feature functions for the phrase translation rules, and decodes additional reference transcriptions into ASR-like output that can serve as training data for the DLM. Default settings were used in Moses throughout with just a few exceptions that enabled us to increase efficiency. Since we expect there to be no reordering during translation, it was not necessary to build a reordering model or to allow any distortion during decoding. The language models used during tuning and decoding are the same as the ones used in the WFST-based confusion modeling.

3. Training methods

In the training step, the training data represented by morph sequences are converted into numerical feature vectors, which are then used to learn a discriminative model. It was denoted in Section 1 that learning can be done with either a structured prediction or a ranking setting. In our setup we use the linear modeling framework to represent the simulated N-best lists as features and apply a variant of the perceptron algorithm in both settings to train the DLM.

3.1. Linear modeling framework

In this paper, we adopt the same linear modeling framework as in [16]. Let x be the acoustic input and y its reference transcription. Also let $\mathbf{GEN}(x)$ be a function which generates a set of candidate hypotheses $\tilde{y}$, given x. The pairs $(x, \tilde{y})$ are mapped by a representation $\mathbf{\Phi}$ into a feature vector $\mathbf{\Phi}(x, \tilde{y})$. The aim of discriminative training is to optimize the model vector $\mathbf{w}$ which contains the relative weights of the features in $\mathbf{\Phi}$. The model score is defined by the inner product of $\mathbf{w}$ and $\mathbf{\Phi}$.

In semi-supervised training the acoustic input x does not exist, therefore $\mathbf{GEN}(\cdot)$ acts as the simulated hypothesis generator, taking the reference sentence y as its input. The output of $\mathbf{GEN}(y)$ is expected to resemble the N-best list of an ASR system which would have processed the acoustic utterance of that sentence.

3.2. Structured WER-sensitive perceptron (*WPer*)

The perceptron is a popular algorithm applied to solve structured prediction problems. Its goal is to minimize the number of misclassifications by picking the hypothesis among $\mathbf{GEN}(y)$ which has the least number of word errors with respect to the reference y. In this study we adopt a variant of this algorithm called the WER-sensitive perceptron, which tries to minimize a loss function related to the number of word errors rather than the number of misclassifications [17]. The new loss function is defined in terms of the edit distances between the reference transcription and the hypotheses and is denoted by $\Delta(y, \tilde{y})$.

Figure 1 shows a pseudocode of the algorithm. It passes over the training data multiple times. For each training example i, y_i (not to be confused with the y above) is the *oracle* hypothesis which has the lowest WER, and z_i is the hypothesis which gives the highest score under the current model weights. The model updates itself by favoring the features in y_i and penalizing the ones in z_i, with a sensitivity multiplier defined by $\Delta(y_i, z_i)$. In the end, the final weights are averaged for robustness.

input set of training examples $\{y_i : 1 \leq i \leq I\}$,
number of iterations T
$\mathbf{w} = 0$, $\mathbf{w}_{sum} = 0$
for $t = 1 \dots T$, $i = 1 \dots I$ **do**
  $z_i = \text{argmax}_{z \in \mathbf{GEN}(y_i)} \langle \mathbf{w}, \mathbf{\Phi}(z) \rangle$
  $\mathbf{w} = \mathbf{w} + \Delta(y_i, z_i)(\mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i))$
  $\mathbf{w}_{sum} = \mathbf{w}_{sum} + \mathbf{w}$
return $\mathbf{w}_{avg} = \mathbf{w}_{sum}/(IT)$

Figure 1: The *WPer* algorithm

3.3. Ranking WER-sensitive perceptron (*WPerRank*)

The structured perceptron algorithm defines the DLM task as the separation of better examples from worse, by considering only two of the hypotheses in the N-best list. It has been shown in earlier studies ([11, 12]) that it is possible to make use of the other hypotheses by considering this as a list ranking problem, where the number of word errors provides the desired (target) ranking.

The ranking approach states that for any two hypotheses a and b from the same N-best list, if a has fewer word errors, hence a higher rank (is closer to the top of the list) than b, then the model score differences should be greater than some separation threshold $\lambda > 0$:

$$r_a \succ r_b \iff \langle \mathbf{w}, \mathbf{\Phi}(a) - \mathbf{\Phi}(b) \rangle > \lambda \quad (1)$$

In this study we define λ as $\tau\Delta(r_a, r_b)$ where τ is a positive margin multiplier and r denotes the rank of the hypothesis, and update the model in a similar iterative fashion as shown in the pseudocode in Figure 2.

3.4. Testing

Once training is completed, the final averaged model $\mathbf{w}_{avg}$ can be used to select the best scoring hypothesis among ASR outputs ($\mathbf{GEN}(x)$) of some unseen acoustic data x via the following expression:

$$y^* = \underset{\tilde{y} \in \mathbf{GEN}(x)}{\text{argmax}} \{w_0 \log P(\tilde{y}|x) + \langle \mathbf{w}_{avg}, \mathbf{\Phi}(\tilde{y}) \rangle\} \quad (2)$$

Here we also include the recognition score now available from the baseline recognizer, $\log P(\tilde{y}|x)$, in the decision process. Its scaling factor, w_0, is optimized on a held-out set. Considering all y^*, the overall WER represents the system performance.

4. Data and experimental setup

This study utilizes DLM in a Turkish LVCSR system which is used to transcribe broadcast news. Our dataset consists of 194 hours of speech divided into 188 hours of training, 3.1 hours of held-out (validation) and 3.3 hours of test subsets. These subsets contain 105355, 1947 and 1784 utterances, respectively. The recognizer outputs are organized in 50-best lists and represented by morph unigrams.

In our experiments, the training subset is divided into two equal parts. The first part (t_1) is used to construct the confusion models, which are then applied on the reference transcriptions of the second part (t_2) to generate artificial N-best lists. These lists constitute the training examples of the DLM system.

The feature vector $\mathbf{\Phi}$ consists of morph unigram counts. In the real ASR N-best lists of t_1 there are about 38K unique morphs, which sets the upper limit on the number of morphs included in the confusion model.

input set of training examples $\{y_i : 1 \leq i \leq I\}$,
number of iterations T, a positive margin multiplier τ,
a positive learning rate η, a positive decay rate γ
$\mathbf{w} = 0$, $\mathbf{w}_{sum} = 0$
for $t = 1 \dots T$ **do**
  for $i = 1 \dots I$ **do**
    for $(a, b) \in \mathbf{GEN}(y_i)$ **do**
      if $r_a \succ r_b$ & $\langle \mathbf{w}, \mathbf{\Phi}(a) - \mathbf{\Phi}(b) \rangle < \tau\Delta(r_a, r_b)$ **then**
        $\mathbf{w} = \mathbf{w} + \eta\Delta(r_a, r_b)(\mathbf{\Phi}(a) - \mathbf{\Phi}(b))$
    $\mathbf{w}_{sum} = \mathbf{w}_{sum} + \mathbf{w}$
  $\eta = \eta \cdot \gamma$
return $\mathbf{w}_{avg} = \mathbf{w}_{sum}/(IT)$

Figure 2: The *WPerRank* algorithm

The SRILM toolkit [18] is used for building the language models. The WFST-based confusion system is implemented by the OpenFST library [19] while the MT-based system is implemented by using the Moses SMT tool [14]. The parameters τ, η and γ are optimized on the held-out set. The generative baseline and oracle rates are 22.9% and 14.2% for the held-out set, and 22.4% and 13.9% for the test set, respectively.

5. Experimental results

In our first set of experiments, we investigate the effectiveness of the artificial data generated by the WFST- and MT-based confusion models, presented in Section 2. For both techniques, the CM is trained by aligning the ASR N-best outputs from t_1 with their reference transcriptions using the Levenshtein distance, and language model reweighting is applied using ASR-LM and GEN-LM. 100 sentences from the training corpus are selected as the development set for the MT-based model. Finally, the CMs are applied on the reference transcriptions of t_2 to generate 50-best lists.

We train the discriminative models with the WPer and WPerRank algorithms explained in Section 3. The algorithms make 20 and 10 passes over the training data, respectively. The parameter w_0 is optimized on the held-out set. Table 1 reports the system performances in terms of WER on the held-out set, with respect to the confusion modeling technique and the language model employed.

Table 1: *Held-out WER(%), Baseline: 22.9%*

Confusion Model	Language Model	WPer	WPerRank
WFST	ASR-LM	22.8	22.7
	GEN-LM	22.7	22.7
MT	ASR-LM	22.5	22.3
	GEN-LM	22.4	22.3

The interpretation of Table 1 is threefold: First of all, regardless of the language model or the algorithm, the WERs of the MT-based CM technique are lower than those of the WFST, which suggests that the artificial examples generated by the MT model are more appropriate for semi-supervised training. The level of improvement is about 0.3% for WPer and even more for WPerRank (the latter being statistically significant at $p < 0.05$). Second, WPerRank provides remarkably lower WER than WPer with MT, on the contrary to WFST where the difference is insignificant (This latter observation is consistent with [12]). Finally, for both choices of the confusion model or the training algorithm, GEN-LM seems to yield slightly better WER with respect to ASR-LM.

As a second experiment, we consider whether combining the WFST and MT training examples will provide any further system gains. Table 2 shows the error rates of individual and combined results with the GEN-LM language model, this time also including the test set performance. The supervised case in which real ASR outputs of t_2 are used for training the DLM is also given for comparison.

Table 2: *Held-out and test WER(%) with GEN-LM*

Confusion Model	WPer		WPerRank	
	hld	tst	hld	tst
WFST	22.7	22.1	22.7	22.3
MT	22.4	22.3	22.3	21.8
WFST + MT	22.3	22.0	22.2	21.9
Real ASR	22.2	22.0	21.9	21.6

We see from Table 2 that combining simulated training data of two CMs does not result in a significant decrease in WER on the held-out set. On the other hand, there is a 0.3% decrease with WPer on the test set, which suggests that the learned discriminative model is more generalizable to unseen data. Furthermore, the WER is as low as the one achieved using the real ASR hypotheses for training.

Please note that the combination scheme used in this experiment was to simply concatenate the training examples of both sources. Other complicated methods like fusion strategies based on the model (constructing an intermediate model by averaging the weights of two models), score (choosing the hypothesis which is more confidently selected by any of the two models), or outputs (doubling the N-best lists) have also been tried, and were observed to yield very similar test set WER as the one reported.

6. Discussion

In order to understand why the examples generated by the MT-based confusions are a better match for semi-supervised DLM training and why WPerRank provides lower WER than WPer in general, we look at the variability of artificially generated examples and the number of utilized features.

It was noted earlier that there are about 38K unique morphs in the N-best lists of t_1, which is used for training the confusion model. The N-best lists generated by the MT confusions contain more than 28K morphs whereas the ones of the WFST-technique contain only about 22K. The number of unique morphs in real ASR outputs of t_2 is also 38K (not all of the features are the same as t_1). Considering occurrence frequencies of these features, the cosine similarity between the N-best sets can be seen in Table 3.

Table 3: *Cosine similarities between simulated hypotheses*

	WFST	Real
MT	0.994	0.998
Real	0.996	

More than 20K features are shared by the WFST and MT systems. There are about 7K unique morphs in the MT simulations which do not exist in WFST's, as opposed to only about 1K for vice versa. Based on these evaluations we understand that the MT-based artificial examples have more variability than the WFST-based, and are much closer to what the real ASR outputs would look like for the same reference text.

We now look at the number of utilized features after training with both of the algorithms, which is summarized in Table 4.

Table 4: *Number of utilized features (GEN-LM)*

CM	WPer	WPerRank
WFST	10,922	14,227
MT	15,320	24,597
WFST + MT	14,914	24,887
Real ASR	20,469	37,373

Table 4 suggests that there is a positive correlation between the system performance and the number of features utilized. More features are utilized by WPerRank than by WPer since the former considers each and every hypothesis of the N-best list, rather than only two.

7. Conclusions

In this paper we applied semi-supervised discriminative language modeling techniques to improve Turkish LVCSR performance, and compared two artificial data generation techniques, one based on the WFSTs and the other on MT confusions. Using the artificial data as training examples, we trained our models with the structured prediction and ranking variants of the perceptron algorithm which is sensitive to the number of word errors. We showed that the MT simulations provide a better basis for training the DLM, and that a WER reduction of more than 0.3% can be obtained using both of the algorithms, the ranking version performing slightly better. We also found out that fusing the WFST and MT confusions under different strategies yields a small improvement, closer to what the system would give if real ASR data were used.

The variability of the features seems to be an important factor in system performance, and in the future we intend to investigate more on the ways which improve variability within the artificially generated N-best lists. As another direction, we would like to follow a similar approach as in [20] to explore the performance of our system in completely unsupervised conditions, where we do not even have the reference text to train the confusion models.

8. Acknowledgements

The authors would like to thank Ebru Arısoy for the baseline DLM setup, to Haşim Sak for the WER-sensitive perceptron algorithm, to Arda Çelebi for the ASR-based artificial hypothesis generation baseline, to Murat Semerci for the ranking perceptron algorithm, and to Ethem Alpaydın for valuable comments and suggestions. This research is supported in part by TÜBİTAK under the project number 109E142 and by the Turkish State Planning Organization (DPT) under the TAM project number 2007K120610. This research is also partially supported by the National Science Foundation Grant #0964102.

9. References

[1] G. Kurata, N. Itoh, and M. Nishimura, "Acoustically discriminative training for language models," in *Proc. ICASSP*, 2009, pp. 4717–4720.

[2] G.Kurata, N.Itoh, and M.Nishimura, "Training of error-corrective model for ASR without using audio data," in *Proc. ICASSP*, 2011, pp. 5576–5579.

[3] P. Jyothi and E. Fosler-Lussier, "Discriminative language modeling using simulated ASR errors," in *Proc. Interspeech*, 2010, pp. 1049–1052.

[4] Q. Tan, K. Audhkhasi, P. Georgiou, E. Ettelaie, and S. Narayanan, "Automatic speech recognition system channel modeling," in *Proc. Interspeech*, 2010, pp. 2442–2445.

[5] Z. Li, Z. Wang, S. Khudanpur, and J. Eisner, "Unsupervised Discriminative Language Model Training for Machine Translation using Simulated Confusion Sets," in *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Committee, Aug., pp. 656–664.

[6] P. Xu, D. Karakos, and S. Khudanpur, "Self-supervised discriminative training of statistical language models," in *Proc. ASRU*, 2009, pp. 317–322.

[7] K. Sagae, M. Lehr, E. T. Prud'hommeaux, P. Xu, N. Glenn, D. Karakos, S. Khudanpur, B. Roark, M. Saralar, I. Shafran, D. Bikel, C. Callison-Burch, Y. Cao, K. Hall, E. Hasler, P. Koehn, A. Lopez, M. Post, and D. Riley, "Hallucinated N-best lists for discriminative language modeling," in *Proc. ICASSP*, 2012.

[8] A. Çelebi, H. Sak, E. Dikici, M. Saraçlar, M. Lehr, E. Prud'hommeaux, P. Xu, N. Glenn, D. Karakos, S. Khudanpur, B. Roark, K. Sagae, I. Shafran, D. Bikel, C. Callison-Burch, Y. Cao, K. Hall, E. Hasler, P. Koehn, A. Lopez, M. Post, and D. Riley, "Semi-supervised discriminative language modeling for Turkish ASR," in *Proc. ICASSP*, pp. 5025–5028.

[9] 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.

[10] L. Shen and A. K. Joshi, "Ranking and reranking with perceptron," *Machine Learning*, vol. 60, pp. 73–96, September 2005.

[11] E. Dikici, M. Semerci, M. Saraçlar, and E. Alpaydın, "Classification and ranking approaches to discriminative language modeling for ASR," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 291–300, 2013.

[12] E. Dikici, A. Çelebi, and M. Saraçlar, "Performance comparison of training algorithms for semi-supervised discriminative language modeling," in *Proc. Interspeech*, 2012.

[13] H. Sak, M. Saraçlar, and T. Güngör, "Discriminative reranking of ASR hypotheses with morpholexical and N-best-list features," in *Proc. ASRU*, 2011, pp. 202–207.

[14] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbsts, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL Interactive Poster and Demonstration Sessions*, 2007, pp. 177–180.

[15] F. J. Och and H. Ney, "A comparison of alignment models for statistical machine translation," in *Proceedings of the 18th Conference on Computational Linguistics*, 2000, pp. 1086–1090.

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

[17] H. Sak, M. Saraçlar, and T. Gungor, "Morpholexical and discriminative language models for Turkish automatic speech recognition," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 8, pp. 2341–2351, 2012.

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

[19] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *CIAA 2007*, ser. LNCS, vol. 4783. Springer, 2007, pp. 11–23, `http://www.openfst.org`.

[20] P. Xu, B. Roark, and S. Khudanpur, "Phrasal cohort based unsupervised discriminative language modeling," in *Proc. Interspeech*, 2012.