INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Effectively Building Tera Scale MaxEnt Language Models Incorporating Non-Linguistic Signals

Fadi Biadsy, Mohammadreza Ghodsi, Diamantino Caseiro

Google, Inc., USA

{biadsy,ghodsi,dcaseiro}@google.com

Abstract

Maximum Entropy (MaxEnt) language models are powerful models that can incorporate linguistic and non-linguistic contextual signals in a unified framework with a convex loss. MaxEnt models also have the advantage of scaling to large model and training data sizes We present the following two contributions to MaxEnt training: (1) By leveraging smaller amounts of transcribed data, we demonstrate that a MaxEnt LM trained on various types of corpora can be easily adapted to better match the test distribution of Automatic Speech Recognition (ASR); (2) A novel *adaptive-training* approach that efficiently models multiple types of non-linguistic features in a universal model. We evaluate the impact of these approaches on Google's state-of-the-art ASR for the task of voice-search transcription and dictation. Training 10B parameter models utilizing a corpus of up to 1T words, we show large reductions in word error rate from adaptation across multiple languages. Also, human evaluations show significant improvements on a wide range of domains from using non-linguistic features. For example, adapting to geographical domains (e.g., US States and cities) affects about 4% of test utterances, with 2:1 win to loss ratio.

Index Terms: speech recognition, language modeling, maximum entropy, model adaptation, contextual adaptation

1. Introduction

State-of-the-art Automatic Speech Recognition (ASR) systems rely on n-gram Language Models (LMs) during *first-pass decoding*. Typically, these models have to be small enough to be able to fit in RAM, and fast enough to perform real-time transcription. At Google, for example, these *first-pass* LMs consist of at most 200 million n-grams, depending on the language. The output of first-pass decoding is a word-lattice. This lattice, in a two-pass system, is then rescored using larger or more complex LM(s) to capture a various and a wider range of contextual features, to potentially improve the long tail of possible hypotheses. This step is called *second-pass rescoring*. Traditionally, the second-pass LM is simply a significantly larger n-gram LM, trained on a large pool of textual corpora. While n-gram models can scale to billions of parameters and 10's of billions of training word tokens [1], they suffer from two problems:

(1) **Model adaptation for in-domain data:** Since most textual data available to train LMs are not speech transcripts (e.g., web documents, news article, books, or typed queries), they may not necessary reflect the test distribution of ASR. To address this problem, the LM is typically adapted on a given *in-domain* manually transcribed data, aiming to better fit this type of data. A well-known adaptation technique for n-gram modeling is linear interpolation of k n-gram models, by optimizing the perplexity on the in-domain data [2]. Although this technique may be adopted by the research community due its simplicity, the learned k interpolation weights are at the corpus level, hence context-independent. Alternatively, Allauzen and Riley [3] have introduced Bayesian LM interpolation which works at the context-dependent level. Bayesian interpolation of large n-gram models can be expensive due to the need to provide estimates from each domain of the probability of each n-gram in the union.

(2) **Domain Modeling:** A central problem in language modeling is how to build flexible and scalable models that can combine information/signals from various domains. Examples of these signal might be the gender, dialect, geographical location of the user, is it weekend?, is it winter?, etc. N-gram models are not flexible enough to straightforwardly incorporate these types of knowledge in the model. We would like an efficient and effective approach that allows us to add these type of signals to the model without impacting the general model when the feature is not observed. Also, in practice, a method that learns such feature weights without the need of retraining the entire model might be preferable.

This paper introduces solutions to the above two problems for log-linear based LM. Log-linear LMs provide an alternative to n-gram backoff. Instead of defining a specific model structure with backoff costs and/or mixing parameters, these models combine multiple features into a single feature vector. Learning can be via locally normalized likelihood objective functions, as in Maximum Entropy (MaxEnt) models [4, 5, 6, 7, 8] or global "whole sentence" objectives [9, 10, 11].

Although in the past few years the research community has been focusing on Neural Network LMs (NN), we propose to use MaxEnt models for rescoring due to, in part, these two reasons: **(I)** We need a flexible model, which not only allows us to incorporate various number of signals, but also scales to the amount of data we have at Google. Our textual corpora for American English, for example, is about 1 trillion word tokens. Although NN-based LMs can make use of arbitrary features, as of today, they do not yet scale to these data sizes. **(II)** Our main goal is to optimize our ASR's performance for *short queries* for voice search. The average number of words in our voice search query is about 4 words.[1] We found based on our preliminary research that LSTM, for example, is not effective for this task.[2]

In this work, we test our approaches using some of the largest reported MaxEnt models. The next section describes background work about MaxEnt LMs; Section 3 describes our experimental setup. Adapting our large MaxEnt model using our adaptation technique, we observe in Section 4 large gains over two baselines: unadapted MaxEnt and n-gram models. Afterwards, in Sections 5, we introduce our *MaxEnt adaptive-training* to train non-linguistic signals and show our results. We finally conclude in Section 7.

[1] Computed from a sub-sample of 3 million voice-search queries.

[2] Our future work will focus on making NN-models perform well on short queries.

http://dx.doi.org/10.21437/Interspeech.2017-1203

2. Background

In this section, we briefly describe MaxEnt language modeling. Let $h = w_{i-k}^{i-1}$ be the immediate context before word w_i, $\Phi(h, w_i)$ be a d-dimensional feature vector, θ a d-dimensional parameter vector, and V a vocabulary. Then

$$\mathrm{P}(w_i \mid h) \quad = \quad \frac{\exp(\Phi(h, w_i) \cdot \theta)}{Z(h, \theta)}$$

where Z is a partition function to normalize the model.:

$$Z(h, \theta) \quad = \quad \sum_{v \in V} \exp(\Phi(h, v) \cdot \theta)$$

Training with a likelihood objective function is a convex optimization problem, with well-studied efficient estimation techniques, such as Stochastic Gradient Descent (SGD). The most expensive part of this optimization is the calculation of the normalizer term Z, since it requires summing over the entire vocabulary, which can be very large. This term also needs to be computed during inference, which can be problematic for real-time systems of large vocabulary. To mitigate this problem, we use hierarchical modeling [12] in which the vocabulary is hard clustered into *word-clusters* $c(w)$. Hence, the model becomes: $\mathrm{P}(w_i \mid h) = \mathrm{P}(c(w_i) \mid h) \cdot \mathrm{P}(w_i \mid h, c(w_i))$. Submodels $\mathrm{P}(c(w_i) \mid h)$ and $\mathrm{P}(w_i \mid h, c(w_i))$ are MaxEnt models with a much reduced vocabulary. This technique can speedup model predictions by up to $\sqrt{|V|}$.

Besides improving speed, hierarchical modeling can also improve modeling quality [13]. Our approach differs from [13] in that we do not limit the feature set to n-grams and cluster n-grams, and in that we do not use regularization. For vocabulary clustering, we use the distributed algorithm described in [14].

For all our experiments, we make use of the Iterative Parameter Mixture (IPM) method [15] to distribute the training process, using hundreds of machines.[3]

3. Experimental Setup

We evaluate the impact of our MaxEnt adaptation and domain modeling ideas on ASR on multiple languages. We make use of Google's state-of-the-art ASR system with an LSTM RNN acoustic model [16], and a 5-gram Bayesian interpolated first pass LM. The models described in this paper are used in the second-pass to rescore either lattices, for n-gram models, or lists of 150-best hypotheses, for MaxEnt models. During rescoring, the first-pass LM's log-likelihood is log-linearly interpolated with the second-pass model score.

We rank the vocabulary according to the distribution in machine transcribed ASR logs. The top million words are partitioned in 1000 clusters. The remaining words are assigned to a special cluster <TAIL>. For efficiency, its cluster conditional submodel $P(w|h, c(w) =$ <TAIL>$)$ is estimated using unigram relative frequencies instead of a MaxEnt model. Out of vocabulary words are also assigned to the <TAIL> cluster and receive the lowest probability of all <TAIL> words.

3.1. Feature templates

We organize our feature vector Φ in *feature templates*, each responsible for a particular type of features. Let y be the token being predicted, the feature templates used are: Word n-grams, $< w_{i-k}, \cdots, w_{i-1}, y >$ up to 5-gram; Cluster n-grams, $< c(w_{i-k}), \cdots, c(w_{i-1}), y >$ from 3 to 5-gram; Skip bigrams, $< w_{i-k}, *, y >$ up to 5 word gap; Left and Right skip trigrams, $< w_{i-k}, w_{i-k+1}, *, y >$, and $< w_{i-k}, *, w_{i-1}, y >$, up to 3 word gap. We also use *PrefixBackoff*$_0$ features as described in [17]. These features are shared between contexts h in the same feature template and trigger when a regular feature is missing for a given y.

[3] We have developed our own optimized IPM algorithm that doesn't rely on the MapReduce framework. It avoids writing models to disk to better scale for larger models and datasets. This algorithm is out of the scope of this paper, and will be described in our future work.

Table 1: *Number of words in the training data. B = Billions, M = Millions.*

Language	Train	Adapt	Domain	Vocab
American English *(en-us)*	943B	47M	3.02B	3.63M
French *(fr-fr)*	245B	14M	476M	1.96M
Italian *(it-it)*	139B	16M	119M	3.92M
Russian *(ru-ru)*	182B	66M	238M	2.00M
Turkish *(tr-tr)*	143B	12M	252M	1.99M

Table 2: *Comparing Voice Search (V) and Dictation (D) WER (%) across, 1st-pass only vs. 100-best oracle, rescoring with n-gram LM, unadapted and adapted MaxEnt LMs.*

	fr-fr		tr-tr		ru-ru		it-it	
	V	D	V	D	V	D	V	D
1st Pass	15.9	9.6	15.5	17.5	16.5	18.8	13.0	6.4
Oracle	8.1	3.2	8.1	8.0	8.6	7.0	3.3	2.4
N-Gram	15.6	9.1	14.8	17.8	16.1	16.9	12.6	6.4
MaxEnt	15.6	9.1	14.9	17.5	16.1	16.8	12.6	6.3
+ adapt	**14.8**	**8.7**	**14.7**	**16.7**	**15.7**	**16.0**	**12.4**	**6.1**

The model is initialized by selecting the top most frequent features in the training data. Model sizes are 10 billion parameters for American English, and 5 billion for the other languages.

Our distributed training algorithm runs on 500 machines with exponential decay learning rate [18]. Depending on the language, it takes 4-20 hours of training time for 4-15 epochs.

4. MaxEnt Model Adaptation

The vast majority of the training data available to train our LMs consist of typed text: web documents, anonymized typed query logs, news articles and books. Unfortunately, models trained on such data are unlikely to perform well on our task, which is the transcription of voice search, spoken questions, voice commands, dictation and speech inputs for 3rd party apps.

To alleviate this mismatch, we also make use of unsupervised data (automatically transcribed voice search and dictation anonymized logs) in our training data. However, this type of data may contain errors, especially for languages with high WER. To fine tune our system, we make use of a sample of manually transcribed data. We call this data *adapt dataset*. Table 1 shows the amount of data we use in this paper.

We first train our MaxEnt models, as described in Section 3, on the pool of data described above (train + adapt). Then, upon model convergence (tested on a held-out dev set), using the distributed SGD training, we present the shuffled adapt data *only*, to the training algorithm, and run it for four iterations with a step function learning rate: 0.2, 0.15, 0.1, 0.05. We use 30-50 machines, depending on the language. These learning rates have been chosen empirically optimizing perplexity of a held-out dev set. We update all parameters of all active features during training – thus, all features that share the same context will be updated. The learning rate of the first epoch we use for the adaptation step is typically higher than the minimum learning rate reached in the first training phase, which is about 0.1.

To evaluate this adaptation technique, we train both traditional 5-gram and MaxEnt models, as shown in Section 3 for 4 different languages. We select 5 billion parameters for the MaxEnt models using feature frequency, while 5-gram models are pruned to 5 billion parameters using entropy pruning [19].

We observe that MaxEnt LM, even without adaptation, is

generally competitive with n-gram, and often better (see Table 2). However, once the MaxEnt LM is adapted using our approach, it obtains significant reduction in WER across all languages and across both types of data sets. We achieve up to 0.9 WER reduction from the n-gram LM baseline and up to 0.8 compared to the unadapted MaxEnt, and up to 2.8 WER reduction relative to no second-pass rescoring.

5. Domain Adaptation

A domain corresponds to a subset of queries that are defined using a non-linguistic signal that is available during both training and prediction. For example, we define a series of "GEO" domains, such as "California", "New York City", or "Canada". Similarly, knowing the App-ID sending the request, we define App domains – e.g. "YouTube", "Maps", etc. Domains may overlap; for example, the same utterance may belong to both "California" and "YouTube" domains. Given a set of *(domain key, value)* pairs D associated with each utterance, we formalize a domain conditional language model as:

$$\mathrm{P}(w_i \mid h, D) \quad = \quad \frac{\exp(\Phi(h, w_i) \cdot \theta + \Phi_D(h, w_i) \cdot \theta_D)}{Z(h, D, \theta, \theta_D)}$$

where $\Phi_D(h, w_i)$ and θ_D are, respectively, domain dependent feature and parameter vectors.

Note that in this formula, we have two sets of parameters, one for the original background model (θ) and another for the domains (θ_D), to represent the non-linguistic signals. A common technique to train this model is simply to train all parameters jointly on a mix of data with and without domain annotations. Unfortunately, this method introduces multiple problems:

1. Since the overwhelming majority of our textual data do not have these annotations, the training algorithm may not robustly estimate these parameters (θ_D). During SGD, these parameters will be far less active than the domain-independent ones.
2. We want the background model not to change even if we add extra training data for some specific domains. For example, we want to continue getting the exact predictions on voice-search queries, even if we add YouTube App training data (annotated with the App signal).
3. The model is not easily extendable: supporting a new signal may require retraining the model from scratch.

Although this *joint* training approach has these limitations, evaluating it, we demonstrate that it, in fact, negatively impacts WER and it performs poorly on a domain task (as shown in Section 6). We refer to the joint training as **BASE-I**.

Aiming to address problem 1, one might simply present the domain-dependent examples last at the training process. But it is not clear what learning rate to use in this case, since if large learning rate is used, we may greatly affect the background model's parameters; and small ones may not robustly train the domain-dependent parameters. Similarly, we observe that this approach negatively impacts WER. We refer to this approach **BASE-II**. To address these challenges, we propose and test our *adaptive-training* approach:

We first start with a trained and adapted MaxEnt model. Then, for each domain, we add a set of domain specific parameters to the model (θ_D). Recall that the features corresponding to these parameters are triggered during both training and prediction only if the utterance belongs to that domain. The domain parameters are initialized to zero (i.e., $\theta_D = 0$), so at this point the new model is equivalent to the trained background model.

Table 4: *WER of different domain adaptation methods.*

Method	V	D
BASE-I	15.2%	9%
BASE-II	15.2%	9%
Adaptive Training	14.8%	8.7%

We have observed that adding domain-specific unigram and bigram parameters (θ_D) is sufficient for our tested domain tasks. That is, for California, for example, we select the frequent unigrams and bigrams from utterances that are annotated with California. This is important because we want to support multiple domains while maintating a model as small as possible.

After adding all domain specific features, we train these parameters (θ_D) using SGD on *only* annotated data while keeping the background model parameters (θ) *frozen*. We should stress out that even though these parameters are frozen they are still used during gradient computation, but are not updated. Therefore, this approach can be viewed as learning these domain-specific parameters (θ_D) given the background LM predictions, or these are domain-specific biases from the background model. Note, as a result, the model performance is unaffected for utterances that do not belong to a domain, addressing item 2, above.

Since we need an annotated training set for domain adaptation, we must have the corresponding signals in the training data. We use automatically transcribed, speech logs as the source of our domain training data. These sets contain anonymized transcripts, along with additional signals. Some signals, such as GEO location, are only kept at a coarse level.

6. Domain Adaptation Results

To evaluate our domain modeling approach, we have run several "side-by-side" (SxS) experiments, in which each utterance is automatically transcribed by two systems. If the two transcripts are different, they are sent for rating. Each pair of results is rated by two humans. We use SxS experiments because of the following reasons. They can accurately measure semantic changes as opposed to minor lexical differences. Also, we can do a SxS experiment on a specific domain, which only focuses on the fraction of the traffic affected by adapting to that domain. Plus, in SxS experiments, we are able to show additional information to the human raters (such as approximate location of the origin of the query) which allows them to rate more accurately.

For each of the SxS experiments, we present the following results: **Change:** The percentage of utterances for which the two systems produced different transcripts. **Wins/Losses:** The ratio of wins to losses in the experimental system vs. the baseline. We also report the *p-value* for statistical significance. We use $\star\star\star$, $\star\star$, $\star$ and no-star to respectively represent p-value ranges of $< .1\%$, $[.1\%, 1\%)$, $[1\%, 5\%)$ and $\geq 5\%$.

6.1. Domain training method

We use the fr-fr system to evaluate the three alternative domain training methods in section 5: **BASE-I**, **BASE-II**, and our proposed method. We use the same training recipe for all methods. The results are presented in Table 4. We observe that both **BASE-I** and **BASE-II** achieve worse WER than our method, that preserves the WER obtained by the domain independent system, since we do not change the domain-independent features. SxS experiments on the Canadian domain also show that using **BASE-I** or **BASE-II** the domains adapted model is significantly worse than the model before adaptation, both for domain-independent utterances (19/54 Win/Loss) and for domain-dependent (30/58), whereas our approach achieves positive SxS results (49/24) (see Section 6.3).

Table 3: *Examples of wins in the SxS experiment for three English GEO domains*

Domain	Device location	Transcript without GEO signal	Transcript with GEO signal
Country = Canada	Peterborough, ON Oshawa, ON	*the Baroque Era. Pets* *janitorial jobs offshore*	*Peterborough Canada pets* *janitorial jobs Oshawa*
US State = Texas	Irving, TX Baytown, TX	*Urban Police Department* *Ashanti hold it down*	*Irving Police Department* *Russian Depot Baytown*
US State = Louisiana	LaPlace, LA New Orleans, LA	*Aptos weather* *Arlene's arrest*	*LaPlace weather* *Orleans arrest*
City = NYC	NYC NYC	*gypsy* *126 Lake Street in Iceland*	*JFK* *126 Lake Street in Islip*
City = San Francisco	San Francisco San Francisco	*Puccini's local number* *what's the drive time to Penn.*	*PG&E local number* *what's the drive time to Pinole*

Table 5: *App domain SxS results*

	YouTube			Maps			Play Store		
	Win/Loss	%Change	Significance	Win/Loss	%Change	Significance	Win/Loss	%Change	Significance
en-us	$54/29$	7.2%	$\star\star$	$63/52$	7.0%		$69/43$	4.7%	$\star$
fr-fr	$46/32$	16.3%	$\star$	$66/51$	10.0%	$\star\star$	$68/28$	15.9%	$\star\star\star$
it-it	$76/28$	4.7%	$\star\star\star$	$86/49$	4.5%	$\star$	$90/30$	11.4%	$\star\star\star$
ru-ru	$42/51$	4.5%		$92/41$	6.6%	$\star\star\star$	$69/41$	10.0%	$\star\star\star$
tr-tr	$45/33$	11.0%		$65/57$	6.9%		$68/22$	11.7%	$\star\star\star$

Table 6: *French GEO (country) based SxSs.*

French queries from:	Win/Loss	%Change	p-value
Canada	$49/24$	14.0%	$.1\% - .5\%$
Tunisia	$34/20$	5.7%	$5\% - 10\%$
Algeria	$24/18$	12.9%	$10\% - 20\%$
Belgium	$37/21$	2.4%	$10\% - 20\%$

Table 7: *US English GEO based SxSs.*

	Win/Loss	%Change	Significance
Overall	$60/33$	4.4%	$\star\star\star$
Canada	$75/44$	3.4%	$\star\star\star$
U.A.E.	$48/32$	7.5%	$\star\star$
Texas	$82/36$	2.1%	$\star\star\star$
California	$69/45$	1.9%	$\star\star\star$
Florida	$59/42$	1.6%	$\star\star$
Louisiana:	$71/41$	2.0%	$\star\star\star$
Los Angeles	$67/48$	1.8%	$\star\star$
Philadelphia	$65/44$	2.0%	$\star\star\star$
New York City	$64/30$	2.6%	$\star\star\star$

6.2. Application domains

An App domain corresponds to the set of speech queries that originated from a particular App on the user device. We test our adaptive-training approach across five languages for this domain. First, we make use of the models described in Section 4 as our background models. Using the Domains sets, in Table 1, which is ASR speech query logs that are annotated with App and GEO signals, we adapt our models for 4-8 SGD adaptive iterations, as described in Section 5.

Table 5 shows the results of our approach against the background model (no domains). In all cases except for YouTube in ru-ru, we observe improvements, and in the majority of the cases (11 out of 15) the improvements are statistically significant (p-value $< 5\%$). We observe that PlayStore domain performs the best – perhaps due to being a more restricted domain.

6.3. GEO location domains

A GEO domain corresponds to all speech queries originating from within a specific geographical area. (Voice queries may contain approximate location information, if enabled by the user. The geographical features are logged only if they have a user population ≥ 1000, *and* an area $\geq 1\text{km}^2$.) We test our approach on the GEO domain for two ASR systems: American English system (en-us) and French system (fr-fr).

In the fr-fr system, similar to the App experimental setup, we have trained four country specific domains to recognize French speech in Algeria, Belgium, Canada, and Tunisia. Table 6 shows that the use of the country signal improves the quality of our transcripts, but the results are statistically significant only for Canadian French, and approaching significance for Tunisia. We speculate that system tuning is likely to help achieve significant results for the other countries.

For en-us GEO domains, we define domains for each US state, the top 30 most populated US cities, and the top 20 countries using the "en-us" system. As shown in Table 7, we observe significant reduction in errors for all tested domains. We also run an overall SxS that shows that the overall effect of GEO domains is about $2/1$ Win/Loss ratio, changing 4.4% of the queries, with strong statistical significance. Table 3 shows a few representative examples of our "wins".

7. Discussion

We have discussed the performance of a large scale MaxEnt Language Models (LMs) when used as second-pass rescoring for Automatic Speech Recognition (ASR). As a first contribution, we have described a simple model adaptation approach for MaxEnt LM which exerts significant reduction in word error rate when compared to both 5B n-gram and unadapted MaxEnt second-pass LMs, across four languages, on the task of voice-search and dictation transcription. Our adaptation approach consists of a few iterations of Stochastic Gradient Decent (SGD) on the adaptation data. Our method is not only effective, since it affects all competing parameters that share same history, but it also scales on large models; it is efficient, and easily distributed, using the standard distributed SGD training algorithms.

Another main contribution of this paper is introducing and evaluating thoroughly our new *adaptive-training* method. This method allows us to incorporate and efficiently train various non-linguistic signals into MaxEnt without jointly training all the parameters. It has multiple advantages: (1) the original (baseline) MaxEnt model is not affected if no signals are available; (2) Adding new signals to the model can be done without retraining the full model; (3) It scales well and it is efficient since only new parameters of the newly added signals are trained. (4) Our approach significantly outperforms traditional joint training methods; (5) Relying on human evaluation, we have seen that our ASR becomes significantly more accurate across multiple domains – GEO domain: countries, US states, and US cities and/or App domain: YouTube, Maps, and PlayStore – for our American English speech recognizer.

8. References

[1] C. Chelba, D. Bikel, M. Shugrina, P. Nguyen, and S. Kumar, "Large scale language modeling in automatic speech recognition," Google, Tech. Rep., 2012.

[2] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proceedings, Workshop on Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal, Eds. Amsterdam: North Holland, 1980, pp. 381–397.

[3] C. Allauzen and M. Riley, "Bayesian language model interpolation for mobile speech input," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*, 2011, pp. 1429–1432.

[4] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: a maximum entropy approach," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1993, pp. 45–48.

[5] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," *Computer Speech and Language*, vol. 10, pp. 187–228, 1996.

[6] S. F. Chen and R. Rosenfeld, "A survey of smoothing techniques for ME models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 37–50, 2000.

[7] J. Wu and S. Khudanpur, "Efficient training methods for maximum entropy language modeling." in *INTERSPEECH*, 2000, pp. 114–118.

[8] T. Alumäe and M. Kurimo, "Efficient estimation of maximum entropy language models with n-gram features: an srilm extension." in *INTERSPEECH*, 2010, pp. 1820–1823.

[9] R. Rosenfeld, "A whole sentence maximum entropy language model," in *Proceedings of IEEE Workshop on Speech Recognition and Understanding*, 1997, pp. 230–237.

[10] R. Rosenfeld, S. F. Chen, and X. Zhu, "Whole-sentence exponential language models: a vehicle for linguistic-statistical integration," *Computer Speech and Language*, vol. 15, no. 1, pp. 55–73, Jan. 2001.

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

[12] J. Goodman, "Classes for fast maximum entropy training," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7-11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings*. IEEE, 2001, pp. 561–564.

[13] S. F. Chen, "Shrinking exponential language models," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. NAACL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 468–476.

[14] J. Uszkoreit and T. Brants, "Distributed word clustering for large scale class-based language modeling in machine translation," in *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, 2008, pp. 755–762.

[15] K. Hall, S. Gilpin, and G. Mann, "Mapreduce/bigtable for distributed optimization," in *Neural Information Processing Systems Workshop on Leaning on Cores, Clusters, and Clouds*, 2010.

[16] H. Sak, A. W. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *CoRR*, vol. abs/1507.06947, 2015. [Online]. Available: http://arxiv.org/abs/1507.06947

[17] F. Biadsy, K. B. Hall, P. J. Moreno, and B. Roark, "Backoff inspired features for maximum entropy language models," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, 2014, pp. 2645–2649.

[18] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL*, 2009, pp. 477–485.

[19] A. Stolcke, "Entropy-based pruning of backoff language models," in *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 2000, pp. 8–11.