ENTROPY BASED PRUNING OF BACKOFF MAXENT LANGUAGE MODELS WITH CONTEXTUAL FEATURES

Tongzhou Chen[⋆∗] *Diamantino Caseiro*[†] *Pat Rondon*[†]

[⋆] Georgia Institute of Technology, Atlanta, Georgia, USA
[†] Google, Inc. New York, New York, USA

ABSTRACT

In this paper, we present a pruning technique for maximum entropy (MaxEnt) language models. It is based on computing the exact entropy loss when removing each feature from the model, and it explicitly supports backoff features by replacing each removed feature with its backoff. The algorithm computes the loss on the training data, so it is not restricted to models with n-gram like features, allowing models with any feature, including long range skips, triggers, and contextual features such as device location.

Results on the 1-billion word corpus show large perplexity improvements relative for frequency pruned models of comparable size. Automatic speech recognition (ASR) experiments show word error rate improvements in a large-scale cloud based mobile ASR system for Italian.

***Index Terms*—** entropy based pruning, language modeling, maximum entropy modeling, contextual features, geo-domain features

1. INTRODUCTION

Despite the recent popularity of neural network-based language models (NNLM), sparse language modeling techniques are still the state of the art for very large second pass language models used in tasks such as speech recognition for mobile ASR. A likely reason for the better quality of sparse models is that NNLM model training does not yet scale to the amount of data available (often as high as 1 trillion tokens), and the data consist of short queries where the long context modeling capacity of long short term memory language models (LSTMLM) is of little use.

A drawback of sparse modeling techniques, such as conventional n-grams, MaxEnt, or sparse non-negative matrix language models (SNM) [1], is that the number of possible features grows very fast when we increase the modeling context history (how far back the model looks for features).

The most popular solution to this problem is to select only features that occur with some minimum frequency. With conventional n-gram models, this technique is often complemented by training a larger model and then pruning it to the required size [2, 3].

Although there has some been research in pruning techniques for more complex models, such as MaxEnt [4], or SNM [5, 6], most techniques assume that features are either n-grams or extracted from short histories. In this paper we propose a relative entropy pruning technique that allows for arbitrary features.

In [7], backoff features were proposed for MaxEnt models. These features trigger when a regular feature (for example, a high order n-gram) is missing from the model, and they greatly improve the quality of pruned models. Our proposed technique is designed to work with backoff-MaxEnt models.

In the next section we describe related work. In Sections 3 and 4 we introduce our notation and describe the pruning algorithm. In Section 5, we compare our algorithm to frequency pruning in terms of perplexity (PPL) and word error rate (WER) on various models. We conclude in Section 6.

2. PREVIOUS WORK

Pruning is an effective and common feature selection technique in sparse language modeling. In particular, it is often used in classical n-gram models because the total number of possible features grows exponentially with the n-gram order, and even when restricted to features seen in the training data, for large orders, the number of features per order is of the same magnitude as the number of tokens in the training data.

The simplest form of pruning is to select only features that appear a minimum number of times in the training data. This is very effective, yet more sophisticated techniques have been proposed. Seymore and Rosenfeld [2] showed that pruning the n-gram model according to the estimated conditional probability and frequency of features is superior to the traditional frequency counts. Later, Stolcke [3] proposed a criterion based on the relative entropy difference after removing a feature from the original n-gram model with backoffs. Seymore and Rosenfeld's technique can be seen as an approximation to Stolcke's entropy pruning technique. Oguz et al. [6] extended entropy pruning to n-gram SNM models with contextual features. Besides frequency and entropy, other criteria such as mutual information [5], probability and rank [8] have also been proposed.

Compared to conventional n-gram language models, there has been comparably less work on pruning maximum entropy models. The most exhaustive work is Chen at al. [4], where the authors propose and compare a diversity of pruning techniques, including various approximations to relative entropy pruning. Our method differs in various respects: we compute the exact entropy loss for each feature in the training data instead of approximating it; we support more feature types than word or class n-grams; and our method supports backoff features.

3. BACKGROUND

A maximum entropy language model with backoff-inspired features consists of the following parts:

- A set X of word contexts such as previous history or contextual information.
- A set Y denotes the vocabulary.

[∗]This author performed the work while at Google, Inc.

- A set of functions $f^t : X \times Y \mapsto \mathbb{R}^d$ mapping any (x, y) pair to a sparse $0-1$ feature vector in $\mathbb{R}^d$. $f^t(x, y) = 1$ if some property of x with y is true; and 0 otherwise. Each function corresponds to a particular feature template t, for example: 2-gram, skip-5, etc.
- A parameter vector v in $\mathbb{R}^d$.
- Optionally, a set of functions $bo^t : X \times Y \mapsto \mathbb{R}$ assigning backoff weights.

For any $x \in X$ and $y \in Y$, a MaxEnt model gives the posterior as:

$$p(y|x; v) = \frac{N(y|x; v)}{Z(x; v)}.$$

The numerator $N(y|x; v)$ is the exponential of the dot product between a parameter vector v and the sparse feature vectors $f^t(x, y)$, summed with backoffs $bo^t(x, y)$ if backoff features are enabled.

$$N(y|x; v) = \exp(\sum_t v \cdot f^t(x, y) + bo^t(x, y))$$

The denominator $Z(x; v)$ gives a normalization constant.

$$Z(x; v) = \sum_{y' \in Y} N(y|x; v).$$

The backoff functions $bo^t(x, y)$ are non-zero only when a corresponding feature $f^t(x, y)$ does not exist. In this work, the value of the backoff is shared across contexts x, and only dependent on the feature template t and the predicted word y.

4. ENTROPY-BASED MODEL PRUNING

The natural criterion of entropy-based pruning is to minimize the relative entropy between the original and pruned models. As relative entropy is inversely proportional to the difference of the log-likelihoods of the pruned model and the original model, we would like to maximize the difference of the log-likehoods between the pruned and original models.

4.1. Loss function

We define the loss $L_{-f_i^t(x,\tilde{y})}$ as the log-likelihood difference of model without feature $f_i^t(x, \tilde{y})$ [1] and the original model, in the training data $D \subset X \times Y$:

$$L_{-f_i^t(x,\tilde{y})} = \sum_{(x,y) \in D} \log P_{-f_i^t(x,\tilde{y})}(y|x; v) - \log P(y|x; v).$$

For a fixed y, after pruning the feature f_i^t, the posterior becomes

$$p_{-f_i^t(x,\tilde{y})}(y|x; v) = \frac{N_{-f_i^t(x,\tilde{y})}(y|x; v)}{Z_{-f_i^t(x,\tilde{y})}(x; v)},$$

where

$$N_{-f_i^t(x,\tilde{y})}(y|x; v) = \begin{cases} \exp(\log N(y|x; v) - v_i + bo^t(x, \tilde{y})) & \text{if } y = \tilde{y}; \\ N(y|x; v) & \text{otherwise,} \end{cases}$$

and

$$Z_{-f_i^t(x,\tilde{y})}(x; v) = Z(x; v) - N(\tilde{y}|x; v) + N_{-f_i^t(x,\tilde{y})}(\tilde{y}|x; v).$$

The term $bo^t(x, \tilde{y})$ is zero when backoff features are not used.

4.2. Efficiently computing the loss

We compute the loss for all features by iterating once though the training data D: For every example (x, y) we emit the partial loss for every feature used in computing $P(y|x)$. As shown in section 4.1, each partial loss is computed in constant time by reusing $N(x, y)$ and $Z(x)$. These partial losses are then accumulated in the final loss for each feature. This algorithm can be easily distributed using MapReduce [9].

5. EXPERIMENTS

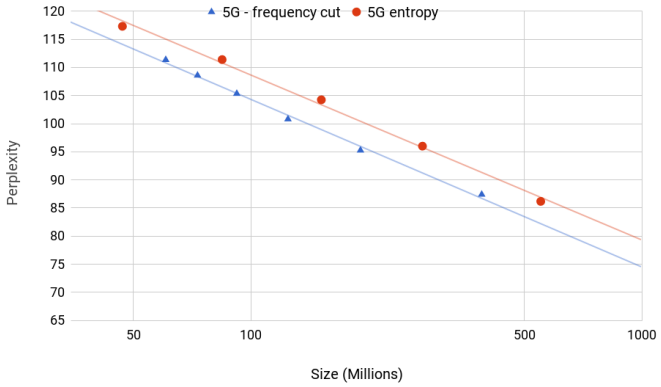
We performed our experiments as follows. For a trained MaxEnt model, we first estimated the entropy loss per feature. Then we pruned the features with loss no less than threshold. Finally, we retrained the pruned model with weights starting from 0 and evaluated the PPL and WER on the pruned, retrained model.

5.1. Experiments on the one-billion-word corpus

We run our perplexity experiments on the one-billion-word corpus [10]. All our MaxEnt models are hierarchical, i.e., $P(w|h) = P(c(w)|h)P(w|x, h)$, and use 1000 clusters. We compare our pruned models to a counts-cutoff baseline, where we prune based on feature frequency. Unigram features, and backoff features, are never pruned.

In the first experiments we used n-gram features only (up to 5-gram), to simplify the analysis. Starting with a model with no pruning, we used various thresholds to obtain models of various sizes and perplexity. The loss for entropy pruning was computed on the training data. Figure 1 shows the results. We observe that entropy pruning is significantly worse than frequency pruning. According to the trendlines, entropy pruning perplexity is about 5 points worse across a wide range of model sizes.

Fig. 1. Model size vs. perplexity of entropy and frequency pruned 5-gram models.



[1] And without its corresponding weight v_i.

Table 1. Percentage of features kept when pruning with threshold -1 using entropy-based pruning alone.

	2-gram	3-gram	4-gram	5-gram
% kept	9.09%	7.64%	19.67%	39.75%

Table 2. Percentage of features kept when pruning at various thresholds after removing features with fewer than 3 occurrences.

Threshold	2-gram	3-gram	4-gram	5-gram
0	93.71%	94.53%	94.63%	95.35%
-0.3	69.89%	63.05%	59.94%	58.91%
-0.8	55.03%	47.29%	44.82%	45.77%
-1.3	46.93%	38.49%	35.76%	37.70%
-1.9	40.77%	31.71%	28.46%	30.94%

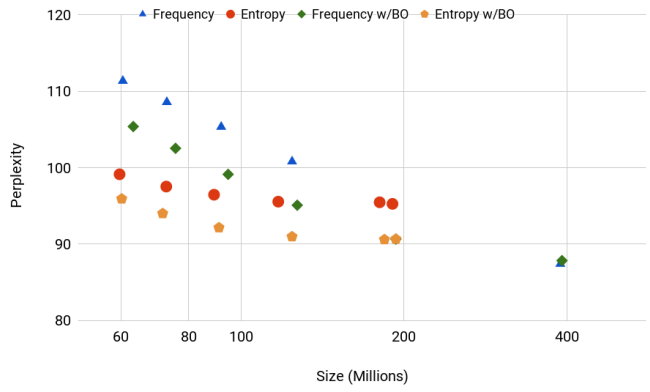
Looking at the number of features selected by template in table 1 we notice that entropy pruning is selecting mostly singleton features, thus overfitting to the training data.

To overcome this problem, we decide to filter out features with low frequencies before applying entropy pruning. Figure 2 compares entropy and frequency pruning on a model with features occurring at least 3 times in the training data.

We see that now entropy pruning achieves much better perplexity for a given model size. Note also that backoff features greatly improve perplexity with minimal increase in model size, and the proposed entropy pruning method is also effective for models incorporating backoff features.

When we break down the pruning results by n-gram order, as in table 2, we see that the pruning algorithm is less aggressive when pruning more general features such as 2-grams and 3-grams than when pruning features with longer, more-specific contexts.

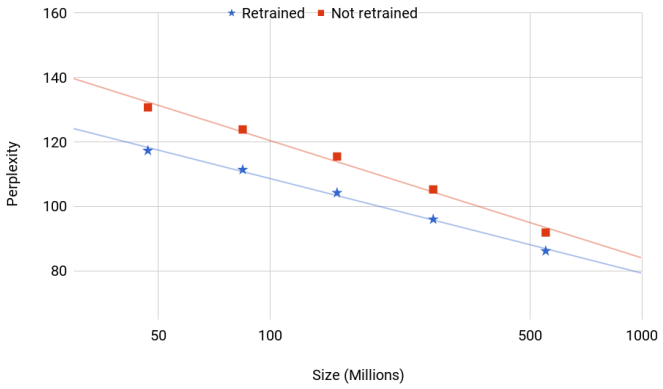
Fig. 2. Model size vs. perplexity of entropy and frequency pruned 5-gram models.



5.1.1. Is retraining necessary?

In the previous experiments, after pruning, we set the weight of remaining features to 0 and retrain the model. To verify if retraining is really necessary, after pruning, we kept the original feature weights and did not retrain the model. Figure 3 compares perplexity with and without retraining and shows that retraining after pruning is indeed important, improving perplexity by 6% to 10%.

Fig. 3. Perplexity of re-trained vs not re-trained models.

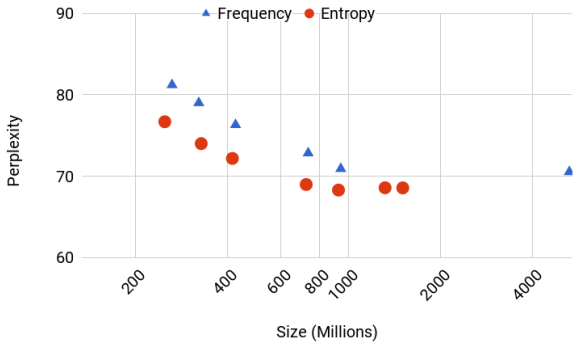


5.1.2. Beyond n-grams

To verify if the proposed entropy pruning model is also effective in models with a larger variety of features types, we trained a MaxEnt model with the following feature templates: word n-grams, $y, x_{i-1}, \cdots, x_{i-k}$ up to 5-gram; word cluster n-grams, $y, c(x_{i-1}), \cdots, c(x_{i-k})$ from 3 to 5-gram; skip 2-grams, $y, *, x_{i-k}$ up to 5 word gap; left and right skip 3-grams, $y, *, x_{i-k+1}, x_{i-k}$, and $y, x_{i-1}, *, x_{i-k}$ up to 3 word gap. To avoid overfitting, we only included features with frequency 2 or above.

Figure 4 shows the size and perplexity of various models. We observe that the algorithm is effective for models with diverse feature types. (Note that perplexity plateaus around 800M features; the difference in perplexity of models beyond this size is a fraction of a point and can be attributed to noise in the distributed training.)

Fig. 4. Model Size vs Perplexity of entropy and frequency pruned models.



5.2. Geographic adaptation experiments

One distinguishing feature of our MaxEnt models is the ability to adapt models to the speaker's context — for example, to their geographic location [11]. An adapted model comprises a base language model, which is unchanged by the adaptation, and a smaller context-specific model adaptation which is trained against context-specific data and applied during prediction only when the user is in the appropriate context. By pruning our model adaptations, we can focus

Table 3. In-domain perplexity before and after pruning.

Pruning Method	PPL Before	PPL After
Locale-Agnostic	74.68	74.62
Locale-Specific		
New York	100.33	96.87
Los Angeles	69.21	80.29
Chicago	75.69	74.52

Table 4. Unigrams with the five largest losses in each region.

New York	Los Angeles	Chicago
kcbs	los	what
brooklyn	what	how
staten	how	what's
bronx	burbank	chicago
manhattan	northridge	cubs

our limited feature budget on contextual features that have the most impact relative to the non-context-adapted model.

In the following experiments, we trained a model for US English with geographic adaptations for the top 30 US cities by traffic, all 50 US states, and top 14 English-speaking countries by traffic; see [11] for model and data details. We then used our entropy-based pruning method to prune the geographic adaptation model, with loss measured against the context's training set. Finally, we retrained the adaptation. We performed two variants of the pruning experiment: an overall pruning which just keeps unigrams and bigrams in the geo-adapted model whose loss is in the 95th percentile (counted separately for unigrams and bigrams), and city-specific prunings which keep the features *within each city* whose losses are in the 95th percentile of the features in that city. We used the largest three US cities for our experiment. For each experiment, we report the difference in perplexity between the unpruned and pruned model (table 3). Table 4 lists the top five unigrams by loss in each city. In each case, there are unigrams with clear city-specific importance (e.g., the boroughs of New York City), though there are also some terms with large losses which are not city specific. These non-city-specific terms likely have large losses because of distributional differences between our adaptation data, which is heavily skewed toward voice queries, and our baseline training data, which uses a larger variety of text sources.

5.3. ASR experiments

We conducted automatic speech recognition (ASR) experiments to gauge the impact of model pruning on automatic speech recognition quality. All experiments were based on Google's cloud based mobile ASR system for Italian.[2] This is a state of the art system with an LSTM acoustic model [12] and a 15 million n-gram LM for the first pass, estimated from various data sources using Bayesian interpolation [13]. In the second pass, a very large MaxEnt language model is used to rescore n-best lists generated by the first pass system.

The corpus used to train the second pass models consists of 35 billion words of mobile written sentences and a small subset of 15 million transcribed words. All data were anonymized and stripped of personally-identifying items. The vocabulary contains 3.9 million words. We rank the vocabulary according to frequency in automatically recognized ASR logs. The most frequent million words are clustered in 1000 clusters. The remaining words are assigned to a special cluster, <TAIL>. For efficiency, we estimate its cluster conditional sub-model $P(w|\Phi(W), c(w) = \texttt{<TAIL>})$ is estimated via unigram relative frequencies instead of MaxEnt.

For training, we use stochastic gradient descent and the distributed IMI algorithm [14, 15]. Our training procedure consists of training on the training data for 5 IMI epochs using 500 worker machines, followed by 3 epochs of adaptation on the transcribed data subset [11]. This adaptation procedure only modifies features in contexts seen in the adaptation data.

We prune the initial 2 billion parameter MaxEnt 2nd pass language model to sizes varying from 25 million to 1 billion parameters. Pruning is performed before the adaptation epochs; we observed that, if we prune the final adapted model, the pruning algorithm discards the adapted features, leading to large quality losses. Figures 5 and 6 show the word error rate and size of models pruned using frequency and entropy pruning. We see that in the voice search test set WER is basically the same as using frequency pruning, but in the dictation test set entropy pruned models achieve up to 0.1% WER improvements.

Fig. 5. Model size vs. WER of pruned models on Voice Search task.

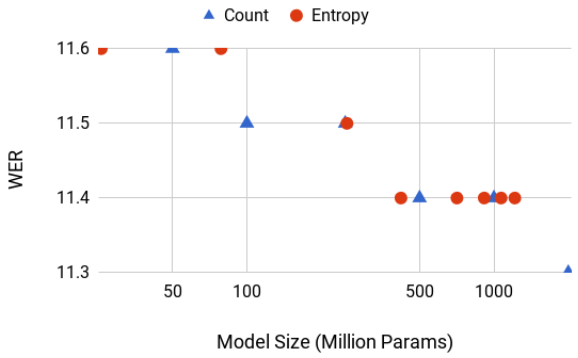
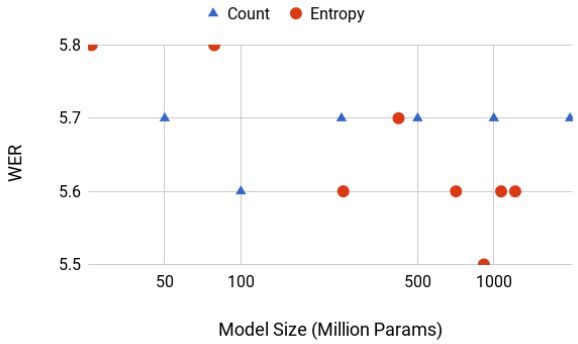


Fig. 6. Model size vs. WER of pruned models on Dictation task.



6. CONCLUSIONS AND FUTURE WORK

We showed that the proposed pruning algorithm for MaxEnt models leads to significantly better models, in terms of perplexity and WER, than frequency pruning. Because we compute the losses on a training dataset we are able to prune models with features such as skips. However, the algorithm is susceptible to overfiting. As future work, we plan to investigate using explicit regularization techniques to address the problem. The proposed algorithm showed promising perplexity results as a feature selector for geo contextual features. Our next steps will be researching the impact on ASR quality.

[2]We switch from English to Italian because we are benchmarking a full ASR system with our technique integrated, rather than just evaluating an LM trained on the 1 billion-word corpus, for which we do not have audio.

7. REFERENCES

[1] Noam M. Shazeer, Joris Pelemans, and Ciprian Chelba, "Sparse non-negative matrix language modeling for skip-grams," in *Proceedings of Interspeech*, 2015, pp. 1428–1432.

[2] Kristie Seymore and Ronald Rosenfeld, "Scalable backoff language models," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. IEEE, 1996, vol. 1, pp. 232–235.

[3] Andreas Stolcke, "Entropy-based pruning of backoff language models," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[4] Stanley F. Chen, Abhinav Sethy, and Bhuvana Ramabhadran, "Pruning exponential language models," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2011, Waikoloa, HI, USA, December 11-15, 2011*, David Nahamoo and Michael Picheny, Eds. 2011, pp. 237–242, IEEE.

[5] Joris Pelemans, Noam M. Shazeer, and Ciprian Chelba, "Pruning sparse non-negative matrix n-gram language models," in *Proceedings of Interspeech*, 2015, pp. 1433–1437.

[6] Barlas Oguz, Issac Alphonso, and Shuangyu Chang, "Entropy based pruning for non-negative matrix based language models with contextual features," in *Proceedings of Interspeech*, 2016, pp. 2328–2332.

[7] Fadi Biadsy, Keith Hall, Pedro J Moreno, and Brian Roark, "Backoff inspired features for maximum entropy language models," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[8] Jianfeng Gao and Min Zhang, "Improving language model size reduction using better pruning criteria," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2002, ACL '02, pp. 176–182, Association for Computational Linguistics.

[9] Jeffrey Dean and Sanjay Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[10] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn, "One billion word benchmark for measuring progress in statistical language modeling," *CoRR*, vol. abs/1312.3005, 2013.

[11] Fadi Biadsy, Mohammadreza Ghodsi, and Diamantino Caseiro, "Effectively building tera scale maxent language models incorporating non-linguistic signals," in *INTERSPEECH*, 2017.

[12] Hasim Sak, Andrew W. Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," *CoRR*, vol. abs/1507.06947, 2015.

[13] Cyril Allauzen and Michael 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.

[14] Keith Hall, Scott Gilpin, and Gideon Mann, "Mapreduce/bigtable for distributed optimization," in *Neural Information Processing Systems Workshop on Leaning on Cores, Clusters, and Clouds*, 2010.

[15] Ryan McDonald, Keith Hall, and Gideon Mann, "Distributed training strategies for the structured perceptron," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 456–464.