EFFICACY OF A CONSTANTLY ADAPTIVE LANGUAGE MODELING TECHNIQUE FOR WEB-SCALE APPLICATIONS

Kuansan Wang and Xiaolong Li

Internet Service Research Center (ISRC), Microsoft Research
Microsoft Corporation
Redmond, WA 98052, USA

ABSTRACT

In this paper, we describe CALM, a method for building statistical language models for the Web. CALM addresses several unique challenges dealing with the Web contents. First, CALM does not rely on the whole corpus to be available to build the language model. Instead, we design CALM to progressively adapt itself as Web chunks are made available by the crawler. Second, given the dynamic and dramatic changes in the Web contents, CALM is designed to quickly enrich its lexicon and N-grams as new vocabulary and phrases are discovered. To reduce the amount of heuristics and human interventions typically needed for model adaptation, we derive an information theoretical formula for CALM to facilitate the optimal adaptation in the maximum a posteriori (MAP) sense. Testing against a collection of Web chunks where new vocabulary and phrases are dominant, we show CALM can achieve comparable and satisfactory model measured in perplexity. We also show CALM is robust against over training and the initial condition, suggesting that any assumptions made in obtaining the initial model can gradually see their impacts diminished as CALM runs its full course and adapt to more data.

Index Terms— N-gram, Statistical language model, MAP adaptation, CALM, Web applications

1. INTRODUCTION

Ever since appearing in Shannon's original work on information theory [1], statistical language models (SLM), often in the form of N-grams, have successfully found many natural language applications, ranging from unsupervised linguistic unit acquisition, speech recognition, information retrieval, to more recently machine translation [2]. All these applications share an amazing characteristic that the SLM takes little advantage of the fact that what is being modeled is a human language. The impoverished use of linguistic knowledge seems largely compensated by the large amount of training data and rigorous statistical techniques. Over the decades, many of these techniques have been widely validated and made available as freely available software toolkits [3, 4, 5].

The massive data available on the Web make it an enticing source for building large scale SLMs. The sheer size of the Web, however, also poses new challenges that have reinvigorated a close look at the well accepted modeling techniques and engineering strategies. Researchers, for example, recently proposed an approach called Stupid Backoff that is tailored to Google Inc.'s Map-Reduce infrastructure [6]. Recognizing that faithfully implementing conventional smoothing methods [7] is a taxing feat, the designers made a radical design choice in forgoing the basic probabilistic properties that all the probabilities of possible events must sum up to 1. By adopting a "probability-like" scoring mechanism, Stupid Backoff uses a single backoff weight whose value is determined manually. Stupid Backoff is technically no longer a statistical model for which various useful properties are applicable. Nevertheless, it is reported that such a tradeoff enables Stupid Backoff to scale up to a size 60 times larger than otherwise possible [7].

In contrast, MSRLM [8] achieves large scale capability by introducing an ingenious data structure to store N-gram in backorder trees. The new data structure, similar to those employed in [5], enables MSRLM to contain the memory use and provides an efficient way in computing backoff. As a result, MSRLM can implement widely known algorithms. Models based on either Stupid Backoff or MSRLM have yielded respectable outcomes in NIST MT evaluations.

In this paper, we describe a technique called constantly adaptive language modeling (CALM) for building SLMs. Although CALM can be applied to unified language model [9] as well, we focus our discussion on N-grams in this paper. Our applications, mostly aiming at Web search related tasks, are different from speech recognition or machine translation such that issues deemed minor in these applications become more prominent and impactful to ours. Specifically, the issues CALM is designed to address include the following. First, our applications need to rely on a properly normalized scoring mechanism to compare various hypotheses, for which probabilistic measures remain an ideal choice and "probability-like" scoring is just not sufficient. Secondly, many of our applications need to be able to quickly reflect the dynamic changes on the Web. As a result, our method needs to be able to keep our SLM as fresh as possible. Third, typically the Web documents are not available to

us all at once. Instead, they often arrive in chunks based on the Web crawler's schedule. Accordingly, smoothing techniques that require the raw statistics of the *entire* corpus to be available, as in MSRLM for example, no longer apply. Finally, the Web documents are very noisy and often full of ill-formed contents. Techniques that use held-out data to fine tune parameters are therefore not robust as they are very sensitive to the quality of the held-out data. In many deployments, we have found that parameters fine-tuned in a lab environment can lead to very brittle outcomes in deployments because data from the field are drastically different from the lab data. CALM is designed to avoid as much as possible any empirical heuristics that may appear engineering appealing but eventually hurt the application performance.

2. ITERATIVE A POSTERIORI ADAPTATION

The central idea of CALM is to constantly adapt the SLM whenever a new Web data chunk becomes available using maximum a posteriori (MAP) adaptation, a technique that has been widely studied [10-13]. Because CALM is building the SLM without seeing the entire corpus, we first address how CALM assimilates new vocabulary and N-grams in this section. Also, a critical issue in MAP adaptation is how to determine the prior probability for adaptation. A key contribution of CALM is to propose a formula to acquire this prior easily.

2.1. Annexation of new vocabulary and N-grams

Let $P^{(i)}(w|h)$ denote a *smoothed* model at time i for a given history h. When a new data chunk is available, CALM updates the smoothed model using

$$P^{(i+1)}(w|h) = \pi^{(i)}P^{(i)}(w|h) + (1 - \pi^{(i)})P_{ML}^{(i)}(w|h) \quad (1)$$

Here, $P_{ML}^{(i)}(w|h)$ denotes the maximum likelihood (ML) estimation purely based on the raw counts of the new data chunk at time i. $P_{ML}^{(i)}(w|h)$ is therefore *not* a smoothed model but nevertheless contains all the information in the data chunk, including the statistics of the new vocabulary (special case of N-grams where N=1) and phrases that are absorbed into the adapted model in the manner of (1). In our implementation, CALM stores N-gram probabilities in a large hash table that allows efficient additions of new vocabulary and N-grams. Note that we do not employ conventional cutoff and discount strategies in CALM. To ensure the quality of parameter estimation and manageable model size, CALM uses a compression algorithm (Sec. 2.3) that can be incorporated into the iterations described by (1). We note that the interpolation technique is actually a mixture distribution concept as expanding the recursion in (1) leads to

$$P^{(\infty)}(w|h) = \sum_{i=-\infty}^{\infty}(1 - \pi^{(i)})P_{ML}^{(i)}(w|h) \quad (2)$$

The choice of the mixture weights is MAP optimal if each mixture weight is the prior probability of the individual component $P_{ML}^{(i)}(w|h)$. A MAP optimal choice of $\pi^{(i)}$ in (1) should therefore reflect how $P^{(i)}(w|h)$ is close to the eventual model $P^{(\infty)}(w|h)$, or how well $P^{(i)}(w|h)$ can already predict the statistics of the new data. This insight underlies the CALM's approach to the prior estimation.

2.2. Prior estimation and Stirling's approximation

Let n_k denotes the count of the k^{th} N-gram token (h, w), $M = \sum_k n_k$ be the total token counts in the data chunk, and $p_k = P^{(i)}(w|h)P^{(i)}(h)$ be the N-gram probability predicted by the smoothed model. Given the fundamental assumption of N-gram is that *non-overlapping* tokens are statistically independent from one another, we can compute how well the smoothed model can already explain the data as

$$\Pr = \frac{M!}{\prod_k n_k!}\prod_k p_k^{n_k}$$

Applying Stirling's approximation $\ln M! \approx M \ln M - M$, one can show that

$$\ln \Pr = \sum_k n_k \ln \frac{p_k}{n_k/M} = M \sum_k \frac{n_k}{M} \ln \frac{p_k}{n_k/M}$$

Since $P_{ML}^{(i)}(w|h) = n_k/M$, we obtain

$$\ln \Pr \approx -MD_{KL}(P_{ML}^{(i)} \| P^{(i)}) \quad (3)$$

where $D_{KL}(P\|Q)$ denotes Kullback-Leibler (KL) divergence between the distributions P and Q. In terms of information theory, KL divergence describes the *per-token* differences of the information in the two distributions. A succinct way to interpret (3) is how well the smoothed model can account for the new data can be computed by how much new information is discovered over all the M tokens in the newly observed data. We note that the adaptation in (1) is conducted on a per-token basis while (3) computes the probability for all M statistically independent tokens. As a result, the per-token interpolation weight can be obtained by

$$\ln \pi^{(i)} = \frac{1}{M}\ln \Pr \approx -D_{KL}(P_{ML}^{(i)} \| P^{(i)}) \quad (4)$$

In the extreme case where $P^{(i)}$ can fully predict the statistics of the new data, the KL divergence in (4) is 0 and $\pi^{(i)} = 1$, leading to the expected outcome that the model does not need to be updated.

Since KL divergence is non-commutative, it is worth noting that (4) is evaluating how the model at hand is different from the newly observed data, rather than how the data is different from the existing N-gram model. In other words, the prior estimated in (4) is indeed a *posterior* statistics.

2.3 Model compression using tied N-grams

Engineering realities dictate that a model's complexity has to be regulated with practical concerns and principled approaches, even though (1) appears to suggest CALM can grow the model indefinitely. Conventional approaches [3,4,7] typically control the model size by excluding N-grams from the model that, in our experience, can lead to undesirable results (Sec. 1). To compress the model, CALM uses a source coding approach in which similar N-grams are identified and forced to share a single distribution. In other words, CALM uses a "*tied*" distribution instead of an N-grams removal approach. Through CALM iterations, tied N-grams can be untied and vice versa, totally driven by data. The compression algorithm in CALM bears strong resemblance to that described in [14] and will be omitted here.

3. EXPERIMENTAL RESULTS

We evaluate the proposed method by comparing the uncompressed SLMs obtained using CALM and MSRLM toolkit that implements modern language modeling techniques. In this article, we report the results for N-gram of order 3 and use perplexity as the general metric for comparison since the models are used in a wide variety of applications.

3.1 Corpus description

As observed in [6] that conventional SLM techniques can still run into formidable obstacles in handling large corpus, we limit for this study to the corpus of Microsoft Support US English web sites that consist of 179,904 documents, 11.2 million sentences and 105.2 million words in 9 web chunks crawled on May 22, 2008. We use the first 8 chunks for training and the last chunk for testing. Both the baseline language model and the eventual CALM model have 1,160,961 unigrams, 5,955,511 bigrams, and 15,643,795 trigrams, i.e., the model complexities are controlled. Based on this corpus, the perplexity measurements for the baseline model with Kneser-Ney smoothing are 486.13, 73.88, and 42.13 for unigram, bigram, and trigram, respectively.

The rationale behind choosing this corpus is that, though being very small by the Web standards, it highlights the challenge in building language models for the Web. Figure 1 depicts how the model grows under CALM (Sec. 2.1) as more data chunks are included in the adaptation. As can be seen, the portion of new N-grams encountered in each chunk is high. Among all the trigrams in the last training chunk (Chunk 8), for example, 54% of them are new. This high ratio is especially worth noting as all these data chunks, which are all in-domain data, can often be misclassified as out-of-domain by techniques that use Web documents for language modeling [15]. We note that a major source of the new N-grams is from the diverse Microsoft product or service names as well as the specific technical terminologies and resource locators that describe the product features and solutions to various end user issues.

3.2 Perplexity measurement against the baseline

Figure 2 shows the perplexity measurements of the CALM models relative to the baseline. Here, we first use MSRLM on Chunk 1 to obtain the initial smoothed model $P^{(0)}$. The priors $\pi^{(i)}$ computed using (4) for the following adaptation chunks are also shown in Figure 2. As can be seen, CALM progressively reduces the perplexity as more data are used to adapt the model. At the end of the first pass, i.e., when all the training chunks are encountered, the CALM model reaches a comparable, if not a better, perplexity than the baseline. The model perplexity remains roughly unchanged going into the second pass where all the training data have been seen before, suggesting CALM is robust against over training. This desirable outcome can be attributed to (4) that shows the model will not be changed much if a high prior estimation indicates the statistics of the data chunk can already be predicted by the model.

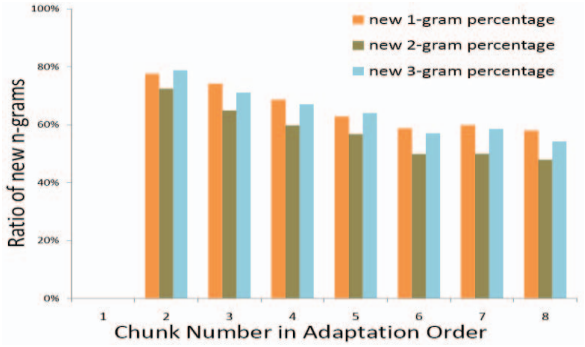

Fig.1. Percentage of new N-grams in each incoming chunk.

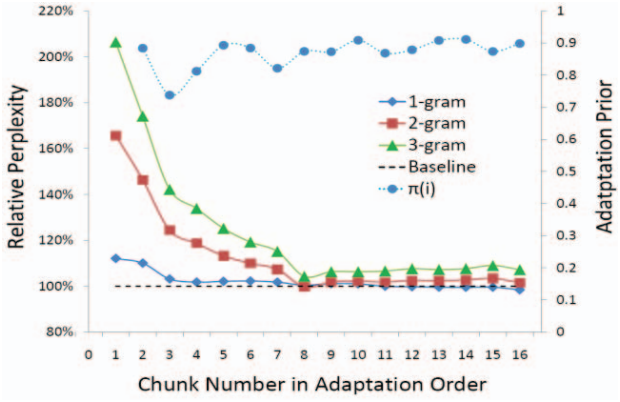

Fig.2. CALM model perplexities, relative to the baseline in percentage, after adapting each data chunk. The second pass results suggest CALM model does not change much after repeated chunk exposure. Also shown is the adaptation prior computed using (4).

3.3 Sensitivity to initial conditions

CALM being an iterative algorithm, a key question to ask is whether the proposed method is sensitive to the choice of the initial condition. Using different chunks as the starting point, we show in Figure 3 that the eventual model perplexi-

ties do not vary much, demonstrating that CALM is not sensitive to the initial conditions.

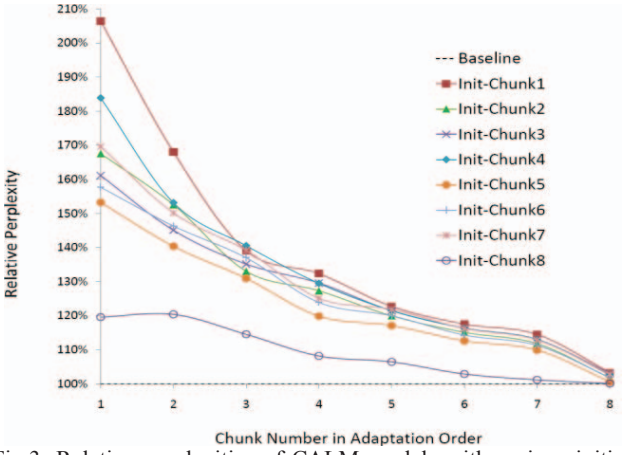

Fig.3. Relative perplexities of CALM models with various initial conditions. After a full pass of adaptation, all reach similar results.

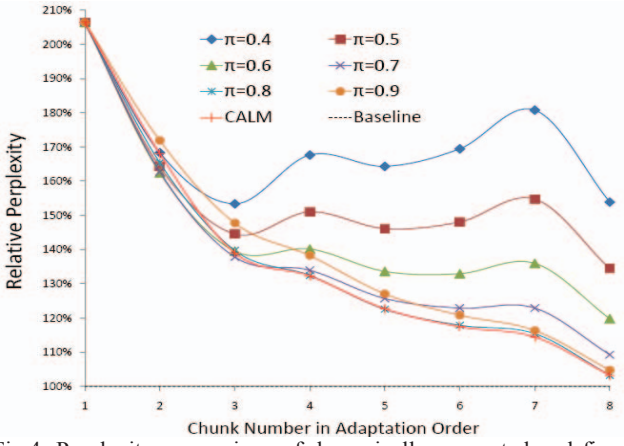

Fig.4. Perplexity comparison of dynamically computed and fixed priors (from 0.4 to 0.9). Dynamically computed prior reaches the lowest perplexity.

3.4 Effects of fixed priors

A key step of CALM is to estimate for each new data chunk the interpolation prior using (4) without using any held-out data. We studied the importance of the dynamically estimated priors by comparing the results from models adapted with fixed priors whose values range from 0.4 to 0.9. As shown in Figure 4, all fixed prior cases result in inferior models, and only when values close to the dynamically computed $\pi^{(i)}$ are used can the final model reach the comparable perplexity.

4. SUMMARY

In this paper we describe CALM, a novel approach for building SLMs for Web-scale corpus. CALM is designed to iteratively adapt to partial data without needing the entire corpus and some held-out data to be available. The iteratively adapting nature of CALM is highly desirable for Web applications that require the SLMs to reflect the fresh contents available on the Web. Even though CALM builds the model without having the global lexicon and N-gram statistics, we show CALM yields comparable model quality as obtained from the state-of-the-art techniques. In addition, we show CALM is resilient to over training and insensitive to initial condition. Finally, we show the method used in CALM to dynamically compute the adaptation priors plays a key role in the model quality.

5. ACKNOWLEDGEMENTS

The authors would like to thank Patrick Nguyen, Jianfeng Gao, and Alex Acero for their invaluable feedback and helpful discussion.

6. REFERENCES

[1] C. E. Shannon, A mathematical theory of communication. *Bell Systems Technical Journal*, Vol. 27, 1948.
[2] R. Rosenfeld, Two decades of statistical language modeling: where do we go from here? In *Proc. IEEE*, Vol. 88(8), 2000.
[3] P. Clarkson and R. Rosenfeld, Statistical language modeling using the CMU-Cambridge toolkit. In *Proc. EuroSpeech-97*, Vol. 1, Rhodes, Greece, 1997.
[4] A. Stockle, SRILM: the SRI language modeling toolkit. In *Proc. ICSLP-2002*, Vol. 2, Denver CO, 2002.
[5] B.-J Hsu and J. Glass, Iterative language model estimation: efficient data structure and algorithms, in *Proc. InterSpeech-2008*, Brisbane, Australia, September 2008.
[6] T. Brants, A. C. Popat, P. Xu, F. Och, and J. Dean, Large language models in machine translation, In *Proc. EMNLP-2007*, pp. 858-867, Prague, June 2007.
[7] S. Chen and J. T. Goodman, An empirical study of smoothing techniques for language modeling. *Technical Report TR-10-98*, Computer Science Group, Harvard University, 1998.
[8] P. Nguyen, J. Gao, and M. Mahajan, MSRLM: a scalable language modeling toolkit. *MSR Technical Report TR-2007-144*, Microsoft Corporation, Redmond, WA, 2007.
[9] K. Wang, Semantic synchronous understanding for robust spoken language applications. In *Proc. ASRU-2003*, pp. 640-645, Virgin Island, December 2003.
[10] J. R. Bellegarda, Statistical language model adaptation: review and perspective. Speech Communications, vol. 42, 2004.
[11] H. Suzuki and J. Gao, A comparative study on language model adaptation techniques using new evaluation metrics. In *Proc. HLT/EMNLP 2005*, Vancouver, BC, October 2005.
[12] G. Tur and A. Stolcke, Unsupervised language model adaptation for meeting recognition. In *Proc. ICASSP-2007*, Honolulu HI, April 2007.
[13] X. Liu, M. Gales, and P. C. Woodland, Context dependent language model adaptation. In *Proc. InterSpeech-2008*, Brisbane, Australia, September 2008.
[14] J. Goodman and J. Gao, Language model size reduction by pruning and clustering, in *Proc. ICSLP-2000*, pp. 110-113, Beijing, China, October 2000.
[15] I. Bulyko, M. Ostendorf, M. Siu, T. Ng, A. Stolcke, O. Cetin, Web resources for language modeling in conversational speech recognition. ACM Trans. Speech and Language Processing, 5(1), pp. 1-25, December 2007.