VARIATIONAL APPROXIMATION OF LONG-SPAN LANGUAGE MODELS FOR LVCSR

Anoop Deoras†, Tomáš Mikolov‡, Stefan Kombrink‡, Martin Karafiát‡, Sanjeev Khudanpur†

†HLTCOE and CLSP, Johns Hopkins University, Baltimore MD 21218, USA
‡Brno University of Technology, Speech@FIT, Czech Republic
{*adeoras, khudanpur*}*@jhu.edu*, {*imikolov, kombrink, karafiat*}*@fit.vutbr.cz*

ABSTRACT

Long-span language models that capture syntax and semantics are seldom used in the first pass of large vocabulary continuous speech recognition systems due to the prohibitive search-space of sentence-hypotheses. Instead, an N-best list of hypotheses is created using tractable n-gram models, and rescored using the long-span models. It is shown in this paper that computationally tractable *variational approximations* of the long-span models are a better choice than standard n-gram models for first pass decoding. They not only result in a better first pass output, but also produce a lattice with a lower oracle word error rate, and rescoring the N-best list from such lattices with the long-span models requires a smaller N to attain the same accuracy. Empirical results on the WSJ, MIT Lectures, NIST 2007 Meeting Recognition and NIST 2001 Conversational Telephone Recognition data sets are presented to support these claims.

Index Terms— Recurrent Neural Network, Language Model, Variational Inference

1. INTEGRATING LANGUAGE MODELS INTO LVCSR

The language model (LM) in most state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems is still the n-gram, which assigns probability to the next word based on only the $n-1$ preceding words. A major reason for using such simple LMs, besides the ease of estimating them from text, is computational complexity. The search space (time) in LVCSR decoding is governed by the number of distinct "equivalent" histories, i.e. the number of unique probability distributions needed to account for all possible histories, and it grows nearly exponentially with n-gram order. So it is customary to limit n to 3 or 4

It is also true, however, that *long-span* LMs, be they due to a higher n-gram order, or because they take syntactic, semantic, discourse and other long-distance dependencies into account, are much more accurate than low-order n-grams. Long-span LMs therefore are employed when accuracy is a priority. The standard practice is to carry out a first pass of decoding using, say, a 3-gram LM to generate a *lattice*, and to rescore only the hypotheses in the lattice with a higher order LM, such as a 4- or 5-gram. But even the search space defined by a lattice is intractable for many long-span LMs. In such cases, only the N-best full-utterance hypotheses from the lattice are extracted for evaluation by the long-span LM. Typically, N is a few thousand, if not a few hundred.

This work was partially supported by National Science Foundation Grant No 0530118 (OISE/PIRE). BUT researchers were partly supported by European project DIRAC (FP6-027787), Grant Agency of Czech Republic project No. 102/08/0707, Czech Ministry of Education project No. MSM0021630528 and by BUT FIT grant No. FIT-10-S-2.

Using an n-gram for generating the N-best list however comes at a price. For one, acoustic model adaptation based on the first pass output may suffer due to its lower accuracy. More pertinently to this work, the n-gram LM may assign such a low score to good hypotheses that they fail to appear among the N-best. If such hypotheses would have eventually surfaced to the top due to the long-span LM, their loss is attributable to the "bias" of the N-best list towards the first pass LM. For both reasons, we seek ways to incorporate information from long-span LMs into first pass decoding.

We propose to do so using **variational inference** techniques. Given a long-span model P, possibly a sophisticated LM with complex statistical dependencies, we will seek a simple and computationally tractable model Q^* that will be a good surrogate for P. Specifically, among all models Q of a chosen family $\mathcal{Q}$ of tractable models, we will seek the one that minimizes the Kullback-Leibler divergence from P. We will use this model for first pass decoding. Examples of P include computationally powerful LMs outside the family of finite state machines, such as recurrent neural networks [1], random forests [2] and structured language models [3, 4]. We will approximate P with a Q^* from the family $\mathcal{Q}$ of finite state machines. The choice of $\mathcal{Q}$ is driven by decoding capabilities.

Section 2 provides details about the proposed variational approximation. Section 3 briefly describes our long-span LM, a recurrent neural network. Section 4 describes our experimental setup and presents a number of results. We conclude with a summary and some remarks in Section 5.

2. VARIATIONAL APPROXIMATION OF A MODEL

There are many popular methods of approximate inference, among which variational inference has gained popularity due to its simplicity [5]. Such methods are necessary when exact inference is intractable. In variational inference, a surrogate model characterized by the distribution $Q \in \mathcal{Q}$ is chosen to replace a complex model, characterized by the distribution P, such that inference under Q becomes much more tractable. The surrogate[1] model Q is chosen such that among all the distributions in the family of the chosen parameterization, $\mathcal{Q}$, it has the minimum Kullback-Leibler divergence with the complex distribution P. Thus if we decide on a family of distributions, $\mathcal{Q}$, then the surrogate distribution is found out by solving the following optimization problem:

$$\begin{aligned} Q^* &= \arg\min_{Q\in\mathcal{Q}} KL(P\|Q) \\ &= \arg\max_{Q\in\mathcal{Q}} \sum_{\cdot} P(\cdot)\log Q(\cdot) \end{aligned} \quad (1)$$

[1]We use P and Q interchangingly for the model or the distribution.

In this work, we choose $\mathcal{Q}$ to be the family of distributions parameterized by n-grams. i.e. we want to learn a model, Q, parameterized by n-grams, that is closest to P in the sense of Kullback Leibler divergence. Under some mild conditions, Q^* is the n-dimensional marginal of P.

A natural question is whether Q^* is simply the n-gram model $\hat{Q}$ estimated from the same (LM training) text that P was estimated from. Not surprisingly, the answer is negative. For one, even if both P and $\hat{Q}$ were estimated from the same text, P may have been estimated with a different criterion than maximum likelihood (ML), so that its n-gram marginals may not agree with $\hat{Q}$, even after differences due to smoothing are ignored. Even if P is the ML estimate from a rich family $\mathcal{P}$ that contain $\mathcal{Q}$ as a subset, additional assumptions must hold about $\mathcal{P}$ for Q^* to be the same as $\hat{Q}$.

But if P is indeed a long-span LM, then computing its n-dimensional marginal could also be computationally prohibitive. Often, and for our choice of P in Section 3, it is impossible. So how does one proceed? For any P that is a generative model of text, the minimizer of (1) may be approximated via Gibbs sampling [5] !

We *simulate* text using the distribution P. Given the start-of-sentence symbol <s>, we sample the next word from the probability distribution conditioned on <s>, and continue generating words conditioned on already generated words, i.e. given the sequence $w_1\, w_2\, \ldots\, w_{l-1}$ of words so far, the l^{th} word is sampled from $P(\cdot|w_1, \ldots, w_{l-1})$, conditioned on the entire past. Our estimate of Q^* is simply an n-gram model $\hat{Q}^*$ based on this synthetically generated text.

If P is stationary and ergodic, then from the consistency of the maximum likelihood estimator [6], we know that by solving (1), we essentially find out the maximum likelihood estimate, $\hat{Q}^*$, in the n-gram family, based on the simulated corpus. Thus

$$\lim_{n\to\infty} \lim_{L\to\infty} KL(P||\hat{Q}^*) = 0, \tag{2}$$

where L is the size of the simulated corpus and n the order of the approximated model, $\hat{Q}^*$.

In practice, we choose n such that first pass decoding is tractable; L is chosen as large as possible, subject to memory constraints and diminishing returns in LVCSR performance. Since L is finite, for pragmatic purposes, smoothing of n-gram probability distribution (for modeling Q) allows us to approximate the maximum likelihood probability had we seen an infinite corpus ($L \to \infty$) generated by P.

3. A RECURRENT NEURAL NET LANGUAGE MODEL

It is well known that humans can exploit longer context with great success in guessing the next word in a sentence. It seems natural therefore to construct LMs that implicitly capture temporal information of arbitrary length. Our recent work with a recurrent neural network language model (RNN LM) does so [1], and has shown remarkable improvements in perplexity over n-gram LMs, along with improvement in recognition accuracy. RNN LMs also outperform some combinations of syntactic and n-gram models [7]. We therefore use the RNN LM of [1] as our long-span model, which we will try to approximate via n-grams.

The network has an input layer x, a hidden layer s (also called state or context layer) and an output layer y. Input to the network at time t is denoted $x(t)$, output $y(t)$, and the hidden state $s(t)$. The input $x(t)$ is formed by concatenating a vector $w(t)$, which represents the current word, with output from the context layer $s(t-1)$ to capture long-span dependencies. We refer the readers to [1] for details.

4. EXPERIMENTS, RESULTS AND DISCUSSION

We report experimental results on four corpora. Perplexity measurements on the Penn WSJ Tree-Bank show that a variational 5-gram is competitive with the best reported results for syntactic LMs. Word error rate (WER) reduction in adapting a Broadcast News language model to the MIT Lectures data is shown next. Finally, WER reductions are demonstrated on the NIST 2007 Meeting Recognition (rt07s) and the NIST 2001 Conversational Telephone Recognition (eval01) test sets.

4.1. Perplexity Experiments on WSJ

We trained n-gram and RNN LMs on Sections 0-20 (1M words) of the Penn Tree-Bank corpus, and measured their perplexity on Sections 23-24 (0.1M words). Sections 21-22 were used as a held out set for parameter tuning.

Baselines: We used interpolated Kneser Ney smoothing to build 3-gram and 5-gram LMs; we will call them the KN models. We also trained an RNN LM, which we will call RNN-Full. To obtain an alternative long-span model we also trained a *cache* LM from the same training data.

For all models, the vocabulary comprised the $10K$ most frequent words in Sections 0-20.

Variational Approximations: We sampled about $230M$ word tokens using RNN-Full as a generative model. From this sampled corpus, we estimated a 3-gram and 5-gram Kneser Ney smoothed LMs. We will call them the VarApxRNN models. Each of these n-gram approximations was also interpolated with the corresponding n-gram LM estimated from (only) the original LM training corpus; these interpolated LMs will be called the VarApx+KN models.

Setup	PPL	Setup	PPL
KN (3g)	148	Random Forest (Xu)	132
VarApxRNN (3g)	152	-	-
VarApx+KN (3g)	**124**	-	-
KN (5g)	141	SLM (Chelba)	149
VarApxRNN (5g)	140	SLM (Roark)	137
VarApx+KN (5g)	**120**	SLM (Filimonov)	125
VarApx+KN + Cache	111	X-Sent (Momtazi)	118
RNN-Full	102	-	-

Table 1. LM Perplexity on Penn Tree-Bank Sections 23-24.

The first column of Table 1 shows that VarApxRNN performs as well as the KN model of the same n-gram order, and their interpolation, VarApx+KN, outperforms both of them. Since the VarApxRNN model is trained on only the *simulated* text, interpolating it with KN introduces the knowledge present in the original training data (sections $0-20$) bringing the simulated statistics closer to the true distribution. To our knowledge, the perplexity of the RNN-full model is significantly lower than any n-gram model reported in the literature.

Figure 1 empirically supports the asymptotic validity of (2) in the size L of the simulated corpus and model order n.

Comparison with Other Long-Span LMs: An advantage of choosing the Penn Tree-Bank corpus and the particular training/test partition is that several other researchers have reported preplexity

on this setup using various long-span LMs. The second column of Table 1 collects a few such results.

- The random forest language model (RFLM) of Xu [2] asks questions of only the trigram history, and is therefore comparable with VarApxRNN (3g) and VarApx+KN (3g). The RFLM estimates a better 3-gram model from existing text; by contrast, VarApxRNN performs simple estimation from simulated text. It appears that VarApx+KN (3g), which combines simulation with the original text, is better.
- Structured language models have been proposed by Chelba and Jelinek [3], Roark [4] and Filimonov and Harper [8] to exploit *within sentence* long-span dependencies. Table 1 suggests that they are outperformed by VarApx+KN (5g), i.e. by simulating text with RNN-Full and estimating KN 5-gram models.
- Across-sentence dependencies are exploited in the model of Momtazi et al [9]. This performance is nearly matched by VarApx+KN (5g), which only uses the 5-gram context. Moreover, the across-sentence model is a complex interpolation of many word and class models with regular and skip n-grams. The interpolation of VarApx+KN (5g) with another tractable long-span LM, namely the cache LM, outperforms the across-sentence model.

These results suggest that RNN-Full is actually a good approximation to the true distribution of the WSJ text, and the reduction in variance by simulating 300M words of text offsets the bias of the n-gram LM estimated from it. As a result, VarApx+KN (5g) outperforms more sophisticated models that have smaller bias, but suffer higher variance due to the limited (1M words) text corpus.

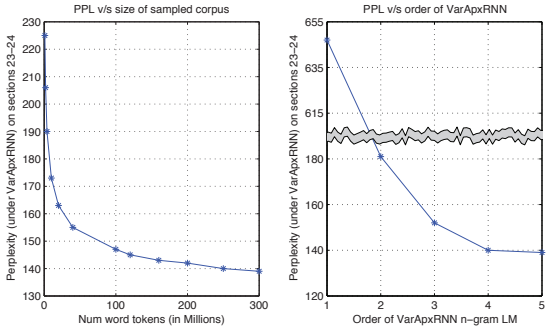

Fig. 1. The perplexity of Sections 23-24 as a function of (left) the size L of the simulated corpus for model order n = 5 and (right) the order n of the model for corpus size L = 300M. These results support (2), but also suggest that VarApxRNN (5g) is still far from the RNN LM, whose perplexity is 102.

4.2. Domain Adaptation Experiments on MIT Lectures

We performed recognition on the MIT lectures corpus [10] using state-of-the-art acoustic models trained on the English Broadcast News (BN) corpus (430 hours of audio), provided to us by IBM [11]. IBM also provided us its state-of-the-art speech recognizer, Attila [12] and an LM containing 4.7M n-grams ($n \leq 4$) that was trained on BN text (335M word tokens). Another 150K words of MIT lecture transcripts were provided as in-domain text.

Interpolated Kneser Ney smoothed n-gram models build with the 150K word in-domain corpus were interpolated with corresponding n-gram LMs from IBM. We will call these models KN:MIT+BN. The RNN LM trained on the 150K words (only) will be called RNN-Full as before.

We simulated text (30M word tokens) using RNN-Full, and estimated n-gram LMs from it, which we will again call VarApxRNN. Models resulting from the interpolation of VarApxRNN and KN:MIT+BN n-gram LMs of the same order will be called VarApx+KN.

We followed IBM's multi-pass decoding recipe [12] using 3- and 4-gram LMs, generated an N-best list from the resulting lattice, and rescored it with RNN-Full. Table 2 shows the WER for different decoding configurations, contrasting standard n-gram LMs with the corresponding VarApx+KN LMs.

We used two sets (2.1 hours each) for decoding. Set 1 was used as a development set for tuning various parameters (acoustic model scaling parameter, language model interpolation weight etc), while Set 2 was used for evaluation.

Setup	**Set 1**	**Set 2**
KN:MIT+BN (4g) decoding	24.7	22.4
+ RNN-Full rescoring (**100** best)	24.1	22.4
+ RNN-Full rescoring (**2000** best)	23.8	21.6
Oracle (**2000** best)	17.9	15.5
VarApx+KN (4g) decoding	24.3	22.2
+ RNN-Full rescoring (**100** best)	23.8	21.7
+ RNN-Full rescoring (**2000** best)	23.6	21.5
Oracle (**2000** best)	17.5	15.1

Table 2. Performance (%WER) on the MIT Lectures data set.

4.3. Conversational Speech Recognition Experiments

We demonstrate WER improvements in two conversational speech recognition tasks: the transcription of multiparty meetings, and of conversational telephone speech. Brno's variant of the AMI system, developed for the NIST Meeting Transcription evaluation [13], was used for the former, and the Brno conversational telephone speech (CTS) system for the latter.

The AMI recognizer used fast speaker adaptation (HLDA, CMLLR and VTLN); it processed PLP+NN-posterior features extracted from 16kHz audio with SAT models trained on 200 hours of meeting data. The CTS recognizer used an initial decoding pass for for VTLN and MLLR, and processed PLP features extracted from 8kHz audio with SAT models trained on 270 hours of telephone speech. All acoustic models were trained using the MPE criterion and used cross-word tied-state triphones, and both setups produced bigram lattices using a 2-gram LM trained using Good-Turing discounting, which were subsequently expanded to 5-gram lattices using a modified Kneser-Ney smoothed LM.

5M words of Fisher CTS transcripts were used as training text for three LMs: two n-grams and an RNN. We call the 2-gram model with Good-Turning discounting GT (2g). The 5-gram model and the RNN model are called KN (5g) and RNN-Full, as before. 400M words of text generated from RNN-Full LM via Gibbs sampling were used to estimate additional n-gram LMs, which we again call VarApxRNN. Altogether, this resulted in a total of four LM configurations, 2-gram vs 5-gram $\times$ standard n-gram vs variational approximation. Since the LMs were trained on CTS transcripts, they

are in-domain for conversation telephone recognition (eval01), but out-of-domain for meeting recognition (rt07s).

The four LMs were applied to rt07s and eval01, and the WERs are reported in Table 3. The table also illustrates the WER when N-best list rescoring with RNN-Full is performed.

Setup	eval01	rt07s
GT (2g) Decoding	30.3	33.7
+ KN (5g) Lattice Rescoring	**28.0**	**32.4**
+ RNN-Full rescoring (**100** best)	27.1	30.8
+ RNN-Full rescoring (**1000** best)	26.5	30.5
Oracle (**1000** best)	19.5	21.3
VarApx+GT (2g) Decoding	30.1	33.3
+ VarApx+KN (5g) Lattice Rescoring	**27.2**	**31.7**
+ RNN-Full rescoring (**100** best)	27.0	30.6
+ RNN-Full rescoring (**1000** best)	**26.5**	**30.4**
Oracle (**1000** best)	19.5	21.0

Table 3. Performance (%WER) on conversational speech data sets.

4.4. Analysis and Discussion of LVCSR Results

From Table 2 it is clear that using VarApx+KN during decoding consistently produces lattices with a 0.5% lower *oracle* WER compared to lattices produced by standard n gram models. The first pass output from decoding with VarApx+KN also has 0.2% to 0.4% lower WER than from decoding with their standard n-gram counterparts. It seems fair to conclude that VarApx+KN is a better n-gram model than a standard n-gram model estimated with Kneser Ney smoothing. Unlike RNN-Full, it can be incorporated into the decoder, bringing some of the benefits of RNN-Full to first pass decoding.

Note further from the upper half of Table 2 that 2000-best rescoring with RNN-Full reduces WER over a standard 4-gram by 0.8% to 0.9%. In the lower half, using VarApx+KN in decoding shows a different benefit: if VarApx+KN is used for generating the N-best list, the same WER reduction is available at 100-best rescoring! If 2000-best rescoring is undertaken, an additional small gain of 0.2% is obtained.

The benefits of decoding & lattice rescoring with the variational approximation of RNN-Full are even more evident from Table 3, where VarApx+KN reduces WER by 0.7%-0.8% over a 5-gram on both CTS and meeting transcription.

A final observation from Table 3 is that there still remains a gap between decoding with VarApx+KN and rescoring with RNN-Full. The latter reduces WER by almost 2% (absolute) over the standard 5-gram, compared to 0.7%-0.8% by the former. This suggests that when the RNN is trained on more data (5M words in Table 3 vs 1M words in Table 1), it improves even further over a 4- or 5-gram model. One may need to investigate further increasing the amount L of simulated data and/or the order n of the approximation in (2), or perhaps consider other tractable model families $\mathcal{Q}$ in (1).

5. CONCLUSION AND FUTURE WORK

We have presented experimental evidence that (n-gram) variational approximations of long-span LMs yield greater accuracy in LVCSR than standard n-gram models estimated from the same training text. The evidence further suggests that the approximated LMs also yield higher quality lattices in terms of the *oracle* WER, and result in more efficient N-best rescoring with the long-span LMs. Both these results advocate for *early integration* of long-span LMs during decoding, even if only in their approximated forms. Finally, there is preliminary evidence that the RNN LM improves significantly over n-grams with increasing training data, calling for an investigation of more powerful tractable approximations.

6. ACKNOWLEDGEMENT

We would like to acknowledge the contribution of Frederick Jelinek towards this work. He would be a co-author if he were available and willing to give his consent. We would also like to thank Jan "Honza" Černocký for his many helpful comments and suggestions. We are grateful to Bhuvana Ramabhadran and Brian Kingsbury for sharing with us state of the art IBM speech recognition system (Attila) and statistical models.

7. REFERENCES

[1] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan "Honza" Černocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proc. of the ICSLP-Interspeech*, 2010.

[2] Peng Xu, *Random forests and the data sparseness problem in language modeling*, Ph.D. thesis, Johns Hopkins University, 2005.

[3] Ciprian Chelba and Frederick Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, no. 4, pp. 283–332, 2000.

[4] Brian Roark, "Probabilistic top-down parsing and language modeling," *Computational Linguistics*, vol. 27, no. 2, pp. 249–276, 2001.

[5] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[6] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, Holden-Day Inc, Oakland, CA, 1977.

[7] Anoop Deoras, Denis Filimonov, Mary Harper, and Fred Jelinek, "Model combination for speech recognition using empirical bayes risk minimization," in *Proc. of IEEE-Spoken Language Technologies*, 2010, to appear.

[8] Denis Filimonov and Mary Harper, "A joint language model with fine-grain syntactic tags," in *Proc. of 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[9] Saeedeh Momtazi, Friedrich Faubel, and Dietrich Klakow, "Within and across sentence boundary language model," in *Proc. of ICSLP-Interspeech*, 2010.

[10] J. Glass, T. Hazen, S. Cyphers, I Malioutov, D. Huynh, and R. Barzilay, "Recent progress in MIT spoken lecture processing project," in *Proc. of ICSLP-Interspeech*, 2007.

[11] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy, "Scaling shrinkage-based language models," in *Proc. ASRU*, 2009, pp. 299–304.

[12] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Workshop on Spoken Language Technology*, 2010, to appear.

[13] T. Hain et. al., "The 2005 AMI system for the transcription of speech in meetings," in *Proc. of Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop, UK*, 2005.