Large-Scale Language Modeling with Random Forests for Mandarin Chinese Speech-to-Text

Conference Paper *in* Lecture Notes in Computer Science · August 2010

DOI: 10.1007/978-3-642-14770-8_31 · Source: DBLP

Ilya Oparin
Apple Inc.
28 PUBLICATIONS **560** CITATIONS

Lori Lamel
French National Centre for Scientific Research
423 PUBLICATIONS **14,664** CITATIONS

Jean-Luc Gauvain
Computer Science Laboratory for Mechanics and Engineering Sciences
316 PUBLICATIONS **12,978** CITATIONS

Improving Mandarin Chinese STT System With Random Forests Language Models

Ilya OPARIN, Lori LAMEL and Jean-Luc GAUVAIN
LIMSI CNRS, Spoken Language Processing Group
B.P. 133, 91403 Orsay, cedex, France

Email: {oparin,lamel,gauvain}@limsi.fr

***Abstract*—The goal of this work is to assess the capacity of random forest language models estimated on a very large text corpus to improve the performance of an STT system. Previous experiments with random forests were mainly concerned with small or medium size data tasks. In this work the development version of the 2009 LIMSI Mandarin Chinese STT system was chosen as a challenging baseline to improve upon. This system is characterized by a language model trained on a very large text corpus (over 3.2 billion segmented words) making the baseline 4-gram estimates particularly robust. We observed moderate perplexity and CER improvements when this model is interpolated with a random forest language model. In order to attain the goal we tried different strategies to build random forests on the available data and introduced a Forest of Random Forests language modeling scheme. However, the improvements we get for large data over a well-tuned baseline N-gram model are less impressive than those reported for smaller data tasks.**

I. Introduction

This paper is concerned with large-scale experiments with random forest (RF) language models (LMs). A random forest is a collection of decision trees (DTs) that include randomization in the tree-growing algorithm. Our earlier experiments with RFLMs on small data showed improvements over the N-gram LM [1]. In the current work we investigate performance of RFLMs on a large data set. Random forest LMs were shown to consistently outperform word-based N-gram LMs for relatively small-scale tasks [2], [3], [4]. There were attempts to use RFLMs on a larger dataset of about 600-700 million words for Mandarin Chinese GALE task [4], [5]. An absolute 0.6% reduction in character error rate (CER) over a 18.9% N-gram baseline CER was shown. However, it is not fully clear how the 4-gram LM that serves as a baseline in these experiments was trained and tuned.

The setup we use is similar to the one used in the above-mentioned experiments but our work is characterized by much larger training data size (3.2 vs. 0.6-0.7 billion words) and twice higher recognition accuracy baseline (9.8% vs. 18.9% CER) attained with a fine-tuned baseline 4-gram model trained without any pruning and cutoff.

Our goal was to improve the performance of a competitive speech-to-text (STT) system with RFLMs trained on several billion words of data. This imposes several types of peculiarities we had to deal with:

1) The brute-force baseline 4-gram LM is trained on very large amounts of text data (over 3 billion of word tokens) without any pruning and cut-off. This model is thus robust and challenging to improve upon.
2) The speech recognizer we use is a development version of the LIMSI STT system component used in the AGILE participation in the GALE'09 evaluation. It provides us with a high recognition accuracy baseline over 90%. Improving over such a baseline is challenging and every small gain is welcome.
3) Training of random forest models is a very computationally expensive process. It is hardly feasible to perform RFLM training in the same fashion it is done for N-gram models for the amounts of data we deal with. Reasonable simplifications should be figured out.

II. Decision Trees and Random Forests

The decision tree mechanism for estimating probabilities of words given contexts has long been known as an alternative to the N-gram approach [6].

With the help of DTs it is possible to cluster together similar histories (i.e. possible previous words regarding the one being predicted) at the leaves of a tree. Each leaf forms an equivalence class of the histories that share the same probability distribution over words to predict. Usually binary decision trees are implemented with sets of possible histories split at every node with a *yes/no* question. If the predictor (i.e. position in N-gram history we ask questions about) is the previous word, a question looks like "Is the previous word in the set S?"

A DT is constructed in a way to reduce the uncertainty about the event predicted. Thus, entropy can be used to measure how good the question that splits the data at each node is. Entropy of a decision tree can always be decreased by increasing the number of leaves. However, such a tree will not be able to generalize well on unseen data. Stopping criteria should be used for a reasonable termination of the branching. There are a number of possible criteria, for example the minimum entropy reduction threshold. An alternative to such empirical thresholds may be measuring the entropy reduction on heldout data under the same split as for training data. Thus, one might grow a tree on training data and then check the entropy on heldout data in order to prune the tree.

The whole process of DT construction in its simplest and unrestricted form can be formulated in several steps:

ISBN 978-1-4244-6245-2/10/$26.00 ©2010 IEEE

1) Propagate training data down the nodes starting from the root of the tree.
2) At each node for each predictor variable find the set of values which minimizes the average conditional entropy of training data at a given node. This set of values constitutes the question that splits the data in two parts at a given node.
3) Find the predictor-question pair that leads to the lowest entropy and calculate the entropy reduction.
4) Check the entropy reduction on heldout data with that question and make decision if this split is retained in the tree or the node is not branched and turned into a leaf.

Just as N-gram probabilities, DT probabilities should be smoothed. This can be done both by means of the techniques developed for language modeling or with a recursive in-tree smoothing with parent node probabilities as proposed in [7].

We do not present thorough mathematical formulations for all the issues concerned with DT-based LMs due to lack of space in this paper and advise to refer to [2] for the the details.

Despite the appealing idea of DT language modeling, several studies showed that stand-alone DTs do not outperform traditional smoothed N-gram models [3]. However, with the recent advances in language modeling that extended the use of decision trees to that of random forests, this direction was brought back in the research spotlight.

The underlying assumption of RFs is that while one DT does not generalize well on unseen data, a set of randomized DTs interpolated together might perform better. First, greedy algorithms are used at the stage of DT construction for choosing the best questions to split data. Second, questions in other nodes are not taken into consideration when we try to find the best question at a given node. As a result, trees are not globally optimal. A collection of trees with randomization introduced at the phase of tree construction may be - and actually is - closer to the global optimum.

Different schemes may be used to randomize DTs in order to form a RF. The most commonly used are the random predictor selection and the random initialization of greedy algorithms used to find the “best” question at a node.

It should also be noted that the RF approach is a promising framework to incorporate different sources of information such as syntax and morphology into a language model [2], [8]. Random forest LMs that take account of morphological features were shown to improve the recognition performance for inflectional languages [1].

III. Experiments

A. Chinese Mandarin STT system

1) Recognition vocabulary and acoustic models: Words are not separated by white spaces in Chinese. The solution is thus either to make use of character-based LMs or perform word segmentation as a pre-processing step. The former was shown to be inferior to the latter thus the segmentation approach was taken in this work [9]. We make use of the simple longest-match segmentation algorithm based on 56052 word vocabulary used in previous LIMSI Mandarin Chinese STT systems [9]. However, character error rate is conventionally used to evaluate final recognition performance.

Word recognition has one decoding chain with three passes. The first decoding pass generates a word lattice with cross-word, position-dependent, gender-independent acoustic models, followed by consensus decoding with 4-gram and pronunciation probabilities [10], [11]. Unsupervised acoustic model adaptation is performed for each segment cluster using the CMLLR and MLLR techniques prior to the next decoding pass. The first decoding pass is done with an MLP+PLP+f0 acoustic model, the second uses a PLP+F0 based model, and the third pass also uses an MLP+PLP+f0 acoustic model. The acoustic models all use speaker-adaptive (SAT) and Maximum Mutual Information Estimation (MMIE) training.

Models were trained on 1400 hours of manually transcribed broadcast news and broadcast conversation data distributed by LDC for use in the GALE program, using both standard PLP and concatenated MLP+PLP features. For the PLP models, a maximum-likelihood linear transform (MLLT) is also used. The model sets cover about 49k phone contexts, with 11.5k tied states and 32 Gaussians per state. Silence is modeled by a single state with 2048 Gaussians.

2) Training data: The language model of 2009 LIMSI Mandarin STT system is trained on large amounts of Mandarin Chinese data thus providing the system with robust LM estimates. This makes improvement of the results attained with this system a challenging task. The language model training data consists of 48 different text sources in Mandarin Chinese available by the end of 2009. These sources are collected by different institutions and are diverse in size, genre and internal structure. The total amount of data available for training is 3.2 billion word tokens (after segmentation).

3) Baseline LM: The baseline LM is a word-based 4-gram LM. Individual LMs are first build for each of the 48 corpora. These models are smoothed according to the unmodified interpolated Kneser-Ney discount scheme. No cut-offs and pruning is imposed thus making the LMs to take account of all possible information. These individual models are subsequently linearly interpolated with the interpolation weights tuned on *dev09* data.

4) Test data: The GALE Phase 4 *dev09* sets were used in this study to evaluate the performance of different models. A subset of *dev09* called *dev09s* was also defined for this evaluation. It constitutes about half of *dev09* data. The *dev09* was used to compute the interpolation weights for individual 4-gram LMs while constructing the baseline N-gram LM. As the number of individual models is 48 (one model is trained for each available corpus), this small number of parameters do not result in a bias towards this data. This is supported by the consistent improvements on the previous GALE evaluations dev and test sets. In our experiments *dev09* is used in the same way i.e. we use it only to tune the LM interpolation weights but never use it for RF tuning (e.g. as heldout data at the DT growing phase). This is done in order to keep the same test conditions and to not introduce any biases.

TABLE I
Perplexity of different RF configurations on dev09

DT depth	*50 DT*	*100 DT*	*50 DT int*	*100 DT int*
fully grown	279.4	276.1	206.8	206.4
10000 nodes	299.1	295.7	207.9	207.7
1000 nodes	358.1	356.1	210.7	210.7

B. Training of RF Models

We used the SRILM-compatible RF toolkit in the current experiments [12].

1) RF on restricted data: Decision tree training may be performed on restricted data to evaluate RF parameters. The crucial point is thus to choose the training and heldout data that is likely to be representative regarding the test data. For current task this is broadcast news (bnm) and broadcast conversations (bcm) Mandarin transcribed data as it constitutes the target type of data in the evaluation. Training data were chosen to contain all available *bnm* and *bcm* transcribed data except for the recent *bcm* and *bnm* data released during previous year. The latter were chosen as the heldout data used as a stopping criterion at the DT training phase.

After the structures of constituent DTs are defined, the training data together with additional data are poured down to the leaves to get more robust probability estimates. Additional data contain remaining top four (according to the interpolation weights assigned to component N-gram LMs) text sources. These text sources are characterized by large sizes and thus were downsampled in such a manner that the resulting size corresponds to the weights inferred during the interpolation. The total size of the restricted data are 30M word tokens.

Heldout data are usually used as a stopping criterion (or for post-pruning) during the DT training phase. However, it was shown that shallow RFs that contain DTs of limited "depth" have performance close to the RFs consisting of "fully grown" DTs [5]. We thus first compare the performance of RFs consisting of fully grown and shallow DTs. Another issue that needs evaluation is the number of DTs to form a RF. Usually 100 or 50 randomized DTs are sufficient.

The perplexity results for different RF configurations are presented in Table I. The numbers 50 and 100 correspond to the number of randomized DTs that constitute a RF. The second and the third columns correspond to the performance of RFs as stand-alone models while the last two columns show perplexity of RFs interpolated with the baseline 4-gram LM. As can be seen from this table while the RFs with DTs of maximum 1000 nodes appear to be too shallow, the ones with 10000 nodes perform close to fully grown (and subsequently pruned) trees. There is also no really significant difference between RFs consisting of 50 and 100 trees trained on restricted data.

2) RF trained on all available data: The baseline 4-gram LM is obtained as a result of interpolation of many sub-LMs each being trained on one of 48 available Mandarin text sources. The interpolation weights are tuned on *dev09*. Applying the same strategy to RF construction seems a natural thing to do. The problem that occurred to us is the size of the corpora. There are corpora that contain hundreds million words we found infeasible to train RFs straightforwardly. Training RFs for specific sources consumes a lot of computational time and puts high demands on memory usage. Training about 50 full-grown RFs consisting of hundred DTs on large data may keep busy a modern computational cluster with several dozens of nodes for months. The performance of DTs with maximum 10000 nodes was shown to be close to that of full-grown DTs on restricted data. At the same time such trees are much faster to train. As a result we train maximum 10000 nodes shallow DTs for each individual corpus and consider 50 randomized DTs enough to form a RF. The final RF is obtained by interpolation of the RFs corresponding to different sources. We call such a RF a *Forest of Random Forests* (FRF).

We found it feasible to directly train RFs for 34 sources. Thus the largest corpora were subdivided into 2-4 smaller parts and a 25 DTs random forest is trained for each subpart. RFs for all the subparts corresponding to a given source form the final RF for this source.

Modified Kneser-Ney 4-gram model is trained for each data source. This model serves as a backoff and bailout model for a corresponding RFLM. Modified Kneser-Ney discounting parameters are calculated for each of the models independently. The models were slightly pruned to fit into memory in 32-bit framework. A threshold of 1e-8 was used (similar to -prune parameter in SRILM).

IV. Results

Decision tree probabilities were discounted according to the modified Kneser-Ney scheme. Tree-based LMs are used together with the corresponding lower-order Kneser-Ney smoothed N-gram LMs that serve as backoff models.

The perplexity on the *dev09* and *dev09s* data sets with the baseline 4-gram LM are 211 and 207 respectively. For *dev09s* set, the perplexity with the RF trained on restricted data is 293, which is higher than that with the baseline interpolated N-gram LM trained on all available data. When these two are interpolated together the perplexity on this data set decreases to 201, that corresponds to a 3% relative improvement. Different perplexity results on the whole *dev09* set are presented in Table II.

The *RF* corresponds to the RF trained on restricted data as described in Section III-B1. According to earlier results with RFLMs on smaller setups one would expect a perplexity reduction over the N-gram baseline with standalone RF models. However, in this restricted data experiment we use much less data to train the RFLM and also do not make use of interpolation of LMs corresponding to different data sources.

The *FRF* in Table II stands for the forest of random forests trained on all available data. The stand-alone perplexity of *FRF* is 15% lower than that of the *RF* but still higher than the N-gram baseline. No further improvement was observed after interpolation of the FRF with the N-gram LM.

Perplexity results on earlier GALE development and evaluations sets presented in Table III show similar small improvements with the *RF* model described above. The *Baseline* stands

TABLE II
PERPLEXITY OF RFLMS OF DIFFERENT KINDS ON DEV09

RF type	*Stand-alone*	*Interp*
RF	299	207
FRF	256	208

TABLE III
PERPLEXITY ON PREVIOUS GALE EVALUATION SETS

Set	*Baseline*	*RF*	*Interp*
dev07	184	237	179
eval07	206	289	203
dev08	192	260	187
dev07+eval07+dev08	194	260	190

TABLE IV
CER WITH RFLM ON DEV09S SET

RF weight	*0.0*	*0.1*	*0.2*	*0.3*	*0.5*	*1.0*
CER	9.81	9.77	**9.72**	9.75	9.80	10.41

for the perplexity attained with the baseline 4-gram LM, *RF* corresponds to the stand-alone RF and the final *Interp* column reports on the interpolation of these two kinds of models.

It should be noted that the 2009 LIMSI Mandarin STT also includes the Neural Network (NN) LM [13]. The interpolation of NNLMs with the baseline 4-gram model leads to the perplexity reduction down to 186 on *dev09*. However, if we try to interpolate all three sources, namely baseline 4-gram, NNLM and RFLM, no further improvement is gained over the interpolated N-gram/NN language model.

Speech recognition experiments were carried out in order to evaluate performance of the RF model within the STT system. The lattices generated by the 2009 LIMSI Mandarin STT system with the baseline 4-gram LM were rescored with the best RFLM (the first one from the Table II). These results are presented in Table IV. The column with the zero weight corresponds to the baseline CER attained with the 4-gram model. Small but significant improvement in CER over the baseline N-gram model is observed with the RFLM.

V. CONCLUSION AND FUTURE WORK

Improving over a robust state-of-the-art STT system trained on large amounts of data is a challenging task. Many approaches that perform well on small and medium-size tasks do not scale well to experiments on large data.

In this paper we presented the results on using random forest language models to improve upon a well-tuned competitive Mandarin STT system trained on large data. We proposed and tested Forest of Random Forests scenario for RFLM training that take account of all available data in a manner similar to the one used to train the baseline 4-gram LM. The moderate improvements both in perplexity and character error rate were observed. However, these improvements we observed are less impressive as compared to the gains reported for smaller scale tasks. Moreover, the improvement gained with the RFLM is outpowered by the application of neural network language models that provide larger improvement over the baseline when interpolated with the N-gram LM. One can argue that the RF approach can be viewed as a sophisticated smoothing technique. At the same time a baseline 4-gram LM with a comparatively compact vocabulary of 56k words is trained on very large corpora of 3.2 billion words without any pruning and cutoff. That makes the estimates provided by this model robust and rather reluctant to enhancements. Moreover, the baseline N-gram LM consisted of 48 sub-LMs with properly tuned interpolation weights.

It should be noted that the results presented here are still prone to improvement. Due to very large size of training data and high computational demands imposed by the random forest construction process several simplifications had to be made. E.g. the number of nodes was forced to be not larger than 10000 and backoff LMs were slightly pruned. These simplifications may result in losing some of the potential gain that can be attained with RFLMs. Thus, the major direction of future work is performing efficient straightforward training of RF language models on the same amounts of data available for N-gram LM training. Another direction is the use of different sources of information e.g. part-of-speech or morphology in DTs trained on large data.

ACKNOWLEDGMENTS

This work has been partially supported by OSEO under the Quaero program and by the GALE program. Any opinions, findings or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] I. Oparin, O. Glembek, L. Burget and J. Černocký, "Morphological Random Forests for Language Modeling of Inflectional Languages", Proc. of IEEE Spoken Language Technology Workshop, SLT08, Goa, 189-192, 2008.

[2] P. Xu, "Random Forests and the Data Sparseness Problem in Language Modeling", PhD Thesis, Johns Hopkins University, 2005.

[3] P. Xu, and F. Jelinek, "Random Forests in Language Modeling", Proc. of EMNLP'04, Barcelona, 325-332, 2004.

[4] Y. Su, F. Jelinek and S. Khudanpur, "Large-Scale Random Forest Language Models for Speech Recognition", Proc. of Interspeech'07, Antwerp, 598-601, 2007.

[5] Y. Su, "Knowledge Integration Into Language Models: A Random Forest Approach", PhD thesis, Johns Hopkins University, 2009.

[6] L.R. Bahl, P.F. Brown, P.V. de Souza and R.L. Mercer, "A Tree-Based Statistical Language Model for Natural Language Speech Recognition", CSL, 37:1001-1008, 1989.

[7] J. Navratil, Q. Jin, W. Andrews and J.P. Campbell, "Phonetic Speaker Recognition Using Maximum-Likelihood Binary Decision Tree Models", ICASSP03, Hon Kong, 796-799, 2003.

[8] I. Oparin, "Language Models for Automatic Speech Recognition of Inflectional Languages", PhD Thesis, University of West Bohemia, Plzen, Czech Republic, 2009.

[9] J. Luo, L. Lamel and J.-L. Gauvain, "Modeling Characters Versus Words for Mandarin Speech Recognition", Proc. of ICASSP'09, Taipei, 4325-4328, 2009.

[10] J.-L. Gauvain, L. Lamel and G. Adda, "The LIMSI Broadcast News Transcription System", Speech Communication, 37(1-2):89-108, 2002.

[11] L. Lamel, A. Messaoudi and J.-L. Gauvain, "Improved Acoustic Modeling for Transcribing Arabic Broadcast Data", Proc. of Interspeech'07, Antwerp, 2077-2800, 2007.

[12] Y. Su, Random Forest Language Model Toolkit, http://www.clsp.jhu.edu/ yisu/rflm.html

[13] H. Schwenk and J.-L. Gauvain, "Training Neural Network Language Models On Very Large Corpora", Proc. of JHLT/EMNLP, Vancouver, 201-208, 2005.