K-Component Adaptive Recurrent Neural Network Language Models

Yangyang Shi[1], Martha Larson[1], Pascal Wiggers[2], and Catholijn M. Jonker[1]

[1] Intelligent System Department, Delft University of Technology
Mekelweg 4, 2628CD, Netherlands
{yangyang.shi,m.larson,c.m.jonker}@tudelft.nl
[2] CREATE-IT Applied Research, Amsterdam University of Applied Sciences (HvA)
p.wiggers@hva.nl

Abstract. Conventional n-gram language models for automatic speech recognition are insufficient in capturing long-distance dependencies and brittle with respect to changes in the input domain. We propose a k-component recurrent neural network language model (KARNNLM) that addresses these limitations by exploiting the long-distance modeling ability of recurrent neural networks and by making use of k different sub-models trained on different contextual domains. Our approach uses Latent Dirichlet Allocation to automatically discover k subsets of the training data, that are used to train k component models. Our experiments first use a Dutch-language corpus to confirm the ability of KARNNLM to automatically choose the appropriate component. Then, we use a standard benchmark set (Wall Street Journal) to perform N-best list rescoring experiments. Results show that KARNNLM improves performance over the RNNLM baseline; the best performance is achieved when KARNNLM is combined with the general model using a novel iterative alternating N-best rescoring strategy.

Keywords: Recurrent Neural Networks, Latent Dirichlet Allocation, N-best rescoring.

1 Introduction

The language model plays a crucial role in automatic speech recognition. It is responsible for constraining the sequence of recognized words to sequences occurring in natural language. Conventional n-gram language models calculate the probability of each word based on a history of the preceding $n-1$ words. The length of the history is limited by number of times individual $n-1$ word sequences appear in the training data; as the history grows in length, the n-gram language model faces a quickly increasing data sparseness challenge. In addition to their inadequacy in modeling long-distance dependencies, n-gram models are known for cross-domain brittleness. Within a single domain this shortcoming manifest itself as lack of robustness to variations in the input speech [1].

Recent studies have shown that a recurrent neural network language model (RNNLM) can outperform n-grams [2]. One claim about the performance of the RNNLM is that it projects the high dimensional vocabulary into a low dimensional continuous space.

I. Habernal and V. Matousek (Eds.): TSD 2013, LNAI 8082, pp. 311–318, 2013.

In this way, an RNNLM helps to relieve data sparseness issues. Simultaneously, the recurrent procedure in the RNNLM equips the language models with the memory to address the long-distance dependency insufficiency.

The k-component adaptive recurrent neural network language model (KARNNLM) proposed in this paper adopts the framework of RNNLMs with their capability to model long-distance word dependencies, and extends them to tackle brittleness to variation. Rather than using one single model as in conventional RNNLMs, the KARNNLM contains k component models, each trained on a specific domain that has been discovered in the training data. KARNNLM first uses Latent Dirichlet Location to automatically construct k contextual domains in the training data and divide the data into partitions corresponding to the k-components. Domain models are trained on the individual partitions and combined with a general model trained on the entire corpus. The resulting model is used to rescore N-best lists, the standard procedure for applying the RNNLMs in speech recognition. We propose three methods for combining domain models: KARNNLM, that makes a hard decision for the correct k-component model at the word level, MIX, that creates a linear combination of the sentence-level probabilities of all k-component models and an alternative rescoring strategy.

Taking the fact that word usage pattern varies among contextual domains, a novel approach for exploiting context domains within the RNNLM framework is proposed in this paper. We demonstrate that the proposed method can improve performance over both conventional n-gram language models and generic RNNLM models.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we introduce our KARNNLM approach, including the construction of the contextual domains and the k-component adaptation strategy. Section 4 presents experimental results. The final section gives the conclusion and outlook.

2 Related Work

Our work is related to previous work that has been carried out in three topic areas: recurrent neural networks language modeling, topics based language modeling and two-pass rescoring, each of which is covered in this section in turn.

The recurrent neural network language model RNNLM, originally proposed by [2,3], incorporates the time dimension by expanding the input layer, which represents the current input word, with the previous hidden layer. Theoretically, recurrent neural networks can store relevant information from previous time steps for an arbitrarily long period of time, making it possible to learn long-term dependencies. In [4,5], RNNLMs are used to model the long-term context information by directly treating the contextual information as input to the networks. In this paper, we use k-component strategy together with RNNLMs to take advantage of context domains in language modeling.

The potential of integrating context information into conventional n-gram language models has been long well-established by the speech recognition community [6,7,8,9]. Our work extends previous work on adapting language models to topical context, by looking not only at topic, but rather at general context domains. Such domains are more subtle since they may involve factors such as style, with less marked lexical distribution patterns.

Conventionally, language models that aim to capture context are not applied during the first decoding pass, but rather are used for rescoring [9]. In this paper, we apply standard rescoring that optimizes WER [10]. We use a iterative alternating N-best rescoring approach, motivated by a desire to avoid local optima during the rescoring process.

In our experiments, we test two patterns that we use for combining general models and context models. The first is sentence level mixtures, which is promising since it has previously demonstrated its usefulness within conventional n-gram language modeling framework [6,7]. The second is word-conditioned combinations, which can be seen as related to the approach of [11], that proposes a model in which each word is treated as a topic mixture model for predicting further word occurrences.

3 K-Component Adaptation Recurrent Neural Networks Language Models

3.1 Recurrent Neural Network Language Models

The recurrent neural network adopted in our work originated with [2]. It has three layers: an input layer x, a hidden layer h and an output layer y. It is characterized by the loop between the input layer and hidden layer. At each time t, the input vector $x(t)$ is constituted by the current word vector $w(t)$ as well as a copy $h(t-1)$ from the previous hidden neurons. The sigmoid function and softmax function are used as the activation functions in the hidden layer and output layer, respectively. The weight matrix is estimated by backpropagation-through-time (BPTT)[3].

3.2 K-Component Model and Domain Adaptation

In order to create the k-component domains, we cluster the training data on the sentence level using Latent Dirichlet Allocation (LDA) [12]. We chose LDA since it is a state-of-the-art method for clustering text data; other clustering algorithms can be expected to work as well. The LDA is a probabilistic generative model that represents each sentence as a combination of latent topics. Each sentence is assigned to a cluster based on its largest latent component.

We realize language model adaptation, by interpolating the general RNNLM models with specific contextual component domain RNNLM models:

$$p(w_t|h_t) = \mu_g p_g(w_t|h_t) + \mu_k p_k(w_t|h_t), \tag{1}$$

where p_g is the general model trained on the complete training set and p_k the k-component model. The interpolation weights μ_g and μ_k are tuned using the development data.

Language Model Adaptation. During rescoring, no information concerning the identification of the 'correct' component model is available. We propose two approaches for selection of the appropriate component model, thereby adapting the overall language model. In the first model (KARNNLM), we assign a sentence probability by applying a hard maximum likelihood decision at the word level.

$$p_{krnnlm}(s) = \prod_i \max_k p(w_i|h_i, M_k), \tag{2}$$

where s is a sentence under evaluation. $p(w_i|h_i, M_k)$ represents the conditional probability of the current word w_i given the component language model M_k and history h_i. In the second model (MIX), we do not use a hard decision, but rather create a sentence-level linear combination of the sentence of all k-component models to assign the sentence probability, expressed by:

$$p_{mix}(s) = \sum_k \lambda_k p(s|M_k) = \sum_k \lambda_k \prod_i p(w_i|h_i, M_k), \tag{3}$$

where λ_k is the interpolation weight of component model k. Notice that within each sentence, the w_i is dependent on the same model M_k. The interpolation weight is determined by the N-best list held out data.

3.3 Iterative Alternating N-Best Rescoring

We experiment with a further combination of the general model and models specific to the k-component using an iterative alternating N-best rescoring approach. In the N-best rescoring paradigm [10], N-best hypotheses are generated from one model, which are rescored by other models. In general, the combination weight is learned from the held out data, which can correspond to a local optimum. Here, we propose a simple iterative rescoring strategy to extend the standard strategy. It works as follows:

For two different language models m_1 and m_2, N-best hypotheses N and ratio $\alpha \in (0, 1)$.

1. The language model m_1 is used to rescore the N-best hypotheses N, of which a α portion of hypotheses are selected. $N := \alpha * N$.
2. The hypotheses selected in previous step are rescored by m_2. The N-best list is further reduced to a new list $N := \alpha * N$.
3. If $N == 1$, stop. Otherwise repeat from step 1.

This strategy approaches the optimization problem by exploiting a filter method. In each iteration, the size of N-best list is first reduced by the general RNNLM and then by the KARNNLM or the MIX. The result of the refined N-best list is used into next iteration until the best hypothesis is obtained. This method can contribute to preventing the different combinations of scores from falling into a local optimum.

4 Experimental Evaluation

We conduct two types of comparisons in our experiments. The first compares the general RNNLM with the KARNNLM specific to a particular contextual domain under the oracle situation in which the specific domain is known. The second compares the RNNLM, the KARNNLM and the mixture RNNLM in a speech recognition experiment involving N-best list rescoring.

Table 1. Word prediction accuracy (WPA) results by context domain (CGN data), comparing conventional (RNNLM) trained on the whole data set with domain-specific recurrent neural networks language models (context domain known). Substantial improvement in word prediction accuracy is indicated in bold. '-' indicates the component is too small, random selection did not select data from it.

comp	socio-situational settings	words	RNNLM	KRNNLM
a	Spontaneous conversations ('face-to-face')	2,626,172	24.0	24.2
b	Interviews with teachers of Dutch	565,433	20.2	19.4
c&d	Spontaneous telephone dialogues	2,062,004	25.5	25.4
e	Simulated business negotiations	136,461	24.5	25.0
f	Interviews/ discussions/debates	790,269	18.6	18.3
g	(political) Discussions/debates/ meetings	360,328	**15.9**	**16.5**
h	Lessons recorded in the classroom	405,409	20.9	20.5
i	Live (eg sports) commentaries (broadcast)	208,399	**16.5**	**19.9**
j	Newsreports/reportages (broadcast)	186,072	17.3	16.6
k	News (broadcast)	368,153	**14.5**	**20.0**
l	Commentaries/columns/reviews (broadcast)	145,553	15.2	13.7
m	Ceremonious speeches/sermons	18,075	-	-
n	Lectures/seminars	140,901	14.8	13.0
o	Reading speech	903,043	**14.2**	**15.6**
overall			20.6	21.3

4.1 Data

Spoken Dutch Corpus. This corpus, referred to in Dutch as *Corpus Gesproken Nederlands* and abbreviated CGN, [13,14] is an 8 million word corpus of contemporary Dutch as it is spoken in Flanders and the Netherlands. It consists of 14 components, each associated with a type of socio-situational setting, as shown in Table 1. The socio-situational setting is related to speech style, which is in turn related to the situation in which the speech is produced. Settings range from informal spontaneous conversation to formal read speech. Of the CGN data, 80% is randomly selected for training, 10% for development testing and 10% for evaluation. We selected a vocabulary with 45K words by choosing word types that occur more than once in the training data. In the test data, words which are not in the vocabulary are replaced by an out-of-vocabulary token (OOV rate is 3.8%).

Wall Street Journal. This corpus, abbreviated here WSJ is drawn from the DARPA WSJ '92 and WSJ'93 data sets. We chose to use the same issue of the data set, and the same N-best lists, as used by [2,15]. The training corpus contains 37M words of running text from the NYT section of English Gigaword. The held-out 230K words is used for testing. A part of the N-best list rescoring data is used as development data for tuning the weight for interpolation, language model score, acoustic model score and word insertion penalty. The rest of them are used for evaluation.

4.2 Word Prediction Accuracy Results

Word prediction has many applications in natural language processing, such as augmentative and alternative communication, spelling correction, word and sentence auto completion, etc. Typically word prediction provides one word or a list of words which fit the context best. It actually provides a measurement of the performance of language models [16].

Table 1 provides the comparison, in terms of WPA, between the general RNNLM and the specific component RNNLM over each component. The hidden layer has a size of 300 neurons, the class size is 100 and we train using 5 iterations of backpropagation through time (BPTT), with a block size of 10. We used the same parameter settings for the general and the component models [3].

The results in Table 1 indicate that if the context domain is known, in general, the domain-specific model outperforms the general model. Especially, in the component "News (broadcast)", the domain-specific model improves the WPA of the RNNLM by almost 38%. However, the component RNNLM is a balance of speciality and reliability. For some components which have much similar characteristics with other components, the specialized training can degrade the performance. For example, the component "Commentaries/columns/reviews (broadcast)", the component model gets almost 10% relative WPA reduction. From [17], we can find that this component also got the lowest classification accuracy. In other words, this component is not strictly discriminative with other components.

4.3 Word Error Rate Results

In WPA results, using a hand-labeled Dutch-language corpus, we confirm the ability of KARNNLM to automatically choose the appropriate component. However, in practice, the component information is not available before hand. The experiment on WSJ is intends to deal with such situation.

Table 2 shows the comparison of RNNLM, KARNNLM and MIX in N-best rescoring experiments performed on the the WSJ data set in terms of word error rate 'WER' under the setting –hidden layer size of 100 neurons, class size of 100, 5 iterations of BPTT, BPTT size of 10. The MIX represents mixture k-component RNNLM. T stands for latent topics. The final column is the WER for iterative alternating N-best rescoring, where a pass is carried out with RNNLM before KARNNLM or MIX are applied.

The best performance is achieved by KARNNLM with 10 latent topics. Beyond that, more topics lead to reduced performance. At the same time, we notice that the performance of KARNNLM faithfully tracks the number of topics indicating that determining the optimal number of topics is critical for applying the approach. The sentence-level mixture MIX is not as effective as KARNNLM, and the RNNLM framework appears to do well exploiting the hard decision between components imposed by KARNNLM. The final column shows that both the KARNNLM and MIX achieve additional improvement with the iterative alternating N-best rescoring strategy. The best KARNNLM reduces the WER of RNNLM by absolute 0.70%.

Table 2. The WER results of Kneser-Ney 5-gram, conventional RNNLM, KARNNLM and k component mixture RNNLM

model	WER(%) rescore	WER(%) iterative
KN5	17.30	-
RNNLM	16.83	-
KRNNLM+5T	16.60	16.25
KRNNLM+10T	16.34	16.13
KRNNLM+15T	17.13	16.57
KRNNLM+20T	17.15	16.56
MIX+5T	16.59	16.31
MIX+10T	16.55	16.21
MIX+15T	16.94	16.53
MIX+20T	17.09	16.52

5 Conclusion

We proposed a k-component recurrent neural network language model for speech recognition N-best list rescoring. Our approach addresses the shortcomings of conventional language models with its ability to capture long-distance dependencies and robustness to variations in domain. Each k component language model is a combination of a general RNNLM with a dedicated RNNLM trained on its associated contextual domain. The experiment on the CGN data set demonstrated the ability of k-component models to outperform general RNNLM under the oracle situation in terms of WPA. In N-best list rescoring of the WSJ data set, the KARNNLM with 10 latent topics reduced WER by 0.49% absolute. In order to reduce the risk of overfitting, we used a novel iterative alternating N-best rescoring strategy, which resulted in an absolute WER reduction of 0.70% over the RNNLM.

By demonstrating the potential of component language models in the recurrent neural network language modeling framework, we have set the stage for future work. In [2], it is observed that the performance improves when RNNLM with different architectures are combined. Our results suggest that diverse component models can be selected or combined to strengthen RNNLM. Future work will involve understanding how they can be further improved by combining different architectures. Further, although KARNNLM and RNNLM demonstrate large improvements over n-gram language models, they remain computationally expensive. An immediate next step will focus on methods for reducing computational cost.

Acknowledgement. Thank you to Tomas Mikolov for making the RNNLM Toolkit publicly available and for helpful discussion.

References

1. Rosenfeld, R.: Two decades of statistical language modeling: where do we go from here? Proceedings of the IEEE 88, 1270–1278 (2000)

2. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. In: INTERSPEECH, pp. 1045–1048 (2010)
3. Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531 (2011)
4. Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. In: SLT, pp. 234–239 (2012)
5. Shi, Y., Wiggers, P., Jonker, C.M.: Towards recurrent neural networks language models with linguistic and contextual features. In: 13th Annual Conference of the International Speech Communication Association (2012)
6. Iyer, R., Ostendorf, M., Rohlicek, J.R.: Language modeling with sentence-level mixtures. In: Proceedings of the Workshop on Human Language Technology, pp. 82–87. Association for Computational Linguistics, Morristown (1994)
7. Iyer, R., Ostendorf, M.: Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In: Proc. ICSLP 1996, Philadelphia, PA, vol. 1, pp. 236–239 (1996)
8. Kneser, R., Peters, J.: Semantic clustering for adaptive language modeling. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1997, vol. 2, pp. 779–782 (1997)
9. Clarkson, P., Robinson, A.: Language model adaptation using mixtures and an exponentially decaying cache. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1997, vol. 2, pp. 799–802 (1997)
10. Ostendorf, M., Kannan, A., Austin, S., Kimball, O., Schwartz, R., Rohlicek, J.R.: Integration of diverse recognition methodologies through reevaluation of n-best sentence hypotheses. In: Proceedings of the Workshop on Speech and Natural Language, HLT 1991, pp. 83–87. Association for Computational Linguistics, Stroudsburg (1991)
11. Chin, H.S., Chen, B.: Word topical mixture models for dynamic language model adaptation. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007, vol. 4, pp. IV–169 –IV–172 (2007)
12. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
13. Hoekstra, H., Moortgat, M., Schuurman, I., van der Wouden, T.: Syntactic annotation for the Spoken Dutch Corpus Project (CGN). In: Computational Linguistics in the Netherlands 2000, pp. 73–87 (2001)
14. Nelleke, O.N., Wim, G.W., Van Eynde, F., Boves, L., Martens, J.P., Moortgat, M., Baayen, H.: Experiences from the Spoken Dutch Corpus project. In: Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, pp. 340–347 (2002)
15. Wang, W., Harper, M.P.: The SuperARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, vol. 2, pp. 238–247 (2002)
16. den Bosch, V.: Scalable classification-based word prediction and confusible correction. Traitement Automatique des Langues 46, 39–63 (2006)
17. Shi, Y., Wiggers, P., Jonker, C.M.: Socio-situational setting classification based on language use. In: IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 455–460 (2011)