E-063

Context Dependent Class Language Model Based on LSA

Welly Naptali* Masatoshi Tsuchiya† Nakagawa Seiichi*

Abstract

We propose an alternative method in language model, called context dependent class language model (CDC), to solve data sparseness problem which is suffered by n-gram language model. The proposing method makes usage of the successful ideas of latent semantic analysis (LSA) in projecting discrete words into continuous vector space. We perform classification on the resulting space and then formulate the CDC. Experimental results on the Wall Street Journal (WSJ) corpus show that the interpolation of the proposed method and a backoff trigram model, achieves better performance than state-of-the-art trigram language model as a baseline.

1 Introduction

Speech recognition task is to find the corresponding word sequence given an acoustic input. For an acoustic input A, the corresponding word sequence W is the word sequence that has the maximum posterior probability $P(W|A)$ given by the following equation:

$$\hat{W} = \arg\max_{W}(\log P_A(A|W) + \log P_L(W)) \quad (1)$$

where P_A is based on an acoustic model and P_L is based on a language model. The language model purpose is to assign probabilities to word sequences. The most common language model used in today's automatic speech recognition system is n-gram. An n-gram language model is a simple and powerful method based on assumption that the current word depends on only $n-1$ preceding words. In case of trigram ($n = 3$), the language model gives the following probability to a word sequence $W = w_1, w_2, ..., w_N$

$$P_{NGRAM}(W) = \prod_{i=1}^{N} P(w_i|w_{i-2}, w_{i-1}) \quad (2)$$

The parameters of the language model are usually trained from a very large corpus. If the corpus is not large enough, unreliable probability will be assigned to words which occur only few times. This problem is also known as data sparseness problem.

Recently LSA, which is originally from information retrieval, has been used in language modeling to map discrete word into continuous vector space (LSA space), then use an estimator in the resulting space. Bellegarda [6] combines the global constraint given by LSA with the local constraint of n-gram language model. The same approach is used in [5] but using Neural Network (NN) as an estimator. Gaussian mixture model (GMM) also could be trained on this LSA space [4]. Instead of word-document matrix, word-phrase co-occurrence matrix is used in [1] as a representation of a corpus. Their model shows better performance than the clustering method based on the maximization of the amount of mutual information. However, their model is limited to only the class of the previous word. Our work is similar to their model with some extensions.

This paper describes a context dependent class language model using LSA. The word in the vocabulary is projected to LSA space according to their word's role or position on the sentences. Vector quantization (VQ) is applied to classify a vector space. Then a simple formulation is introduced to calculate its occurring probability.

This paper is divided into four sections. The next section describes the context dependent class language model including a review about LSA and how to build the matrix representation to get the projection matrices. Section 3 reports the experiments of the proposed model. The last section is the conclusions.

2 Context Dependent Class Language Model

Let V be a size of a vocabulary, each word in the vocabulary can be mapped into an l-dimensional vector space according to the following equation:

$$u_i = \mathbf{X}c_i (1 \leq i \leq V), \quad (3)$$

where $\mathbf{X}$ is a projection matrix with $V \times l$ dimension, and c_i is a discrete vector of word w_i, where the i-th element of the vector is set to 1 and all other $V-1$ elements are set to 0. Since u_i is a vector which representing word w_i, any familiar clustering method could be applied, and the word probability could be approximated according to a class based language model

$$P_{CLASS}(w_i|w_{i-n+1}, ..., w_{i-1}) = P(C_i|C_{i-n+1}, ..., C_{i-1})P(w_i|C_i) \quad (4)$$

In LSA, we have a different projection matrix for a different word position. For instance, LSA (see Section 2.1) with bigram matrix gives $\mathbf{U}$ matrix that project the current word w_i into l-dimension space, and matrix $\mathbf{V}$ is a projection matrix for the 1^{st} preceding word w_{i-1}. Thus, we need another formulation that could handle such situation. Hence, we define the context dependent class language model as

$$P_{CDC}(w_i|w_{i-n+1}, ..., w_{i-1}) = P(C(w_i, \mathbf{X}_i)|C(w_{i-n+1}, \mathbf{X}_{i-n+1}), ..., C(w_{i-1}, \mathbf{X}_{i-1})) \times P(w_i|C(w_i, \mathbf{X}_i)) \quad (5)$$

where $C(w_i, \mathbf{X}_i)$ is a class of word w_i based on projection matrix $\mathbf{X}_i$. For an unseen n-gram class, we applied *class backoff* to a lower context class.

The CDC tried to utilize the semantic structure of the language. In the literature, language models that model different aspects have been successfully combined with an

*Toyohashi University of Technology, Department of Information and Computer Sciences

†Toyohashi University of Technology, Information and Media Center

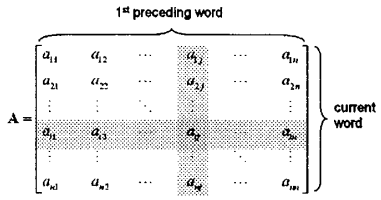

Figure 1: Bigram Matrix

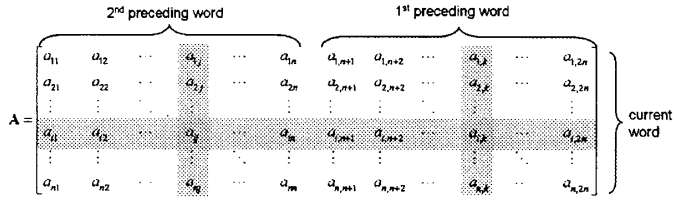

Figure 2: Trigram Matrix

n-gram language model. Here, the statistical n-gram language model used to capture the local constraint using linear interpolation

$$P_L \approx \alpha P_{CDC} + (1-\alpha) P_{NGRAM} \quad (6)$$

where α is a weight constant.

2.1 Latent Semantic Analysis

LSA extracts semantic relations from a corpus, and shows them on the l-dimension vector space. The discrete words are projected into LSA space by applying singular value decomposition (SVD) to a matrix that representing a corpus. Let $\mathbf{A}$ be a representational matrix with $M \times N$ dimension, SVD decomposed matrix $\mathbf{A}$ into three other matrices $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$

$$\mathbf{A}_{M\times N} = \mathbf{U}_{M\times k}\mathbf{S}_{k\times k}\mathbf{V}^T_{k\times N} \quad (7)$$

Because the solution's dimensionality is too large for computing resources and the original matrix $\mathbf{A}$ is presumed to be noisy, the LSA matrices ($\mathbf{U}$ and $\mathbf{V}$ matrix) dimension is smaller than the original

$$\hat{\mathbf{A}}_{M\times N} = \mathbf{U}_{M\times l}\mathbf{S}_{l\times l}\mathbf{V}^T_{l\times N} \quad (8)$$

where $l \ll k$ and $\hat{\mathbf{A}}$ is the best least square fit approximation to $\mathbf{A}$.

2.2 Matrix Representational

Bigram matrix is a matrix representation of a corpus where each row represents a current word w_i, and each column represents a 1^{st} preceding word w_{i-1} as illustrated by Figure 1.

Each cell a_{ij} is a co-occurrence frequency of word sequence $w_j w_i$ in the corpus. The resulting SVD matrix $\mathbf{U}$ is used to project current word into LSA continuous space. While matrix $\mathbf{V}$ is used to project 1^{st} preceding word. In this case, the CDC in Equation (5) becomes

$$P_{CDC}(w_i|w_{i-1}) = P(C(w_i,\mathbf{U})|C(w_{i-1},\mathbf{V}))P(w_i|C(w_i,\mathbf{U})) \quad (9)$$

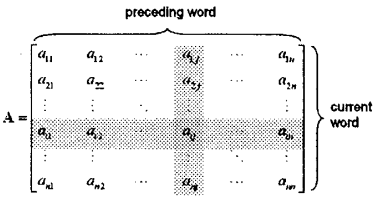

Figure 3: 1-r distance Bigram Matrix

Table 1: Experimental data statistics.

	#Word	OOV rate
Training set	17,283,668	0.0227
Closed test set	22,746	0.0183
Open test set	23,567	0.0283

2.2.1 Trigram Matrix

Figure 2 illustrates the trigram matrix. Unlike the trigram matrix defined in [1], in this paper the two previous words will not be seen as a phrase, but will be put as independent words in the column. By doing this, we made the matrix dimension even smaller. Thus, the trigram matrix is defined as a matrix where each row represents a current word w_i, each column in the first n columns represents 2^{nd} preceding word w_{i-2}, and each column in the second n columns represents 1^{st} preceding word w_{i-1}.

Each cell a_{ij}, for the first n columns ($1 \le j \le n$), is a co-occurrence frequency when the word w_j occurs as the 2^{nd} preceding word of word w_i. For the second n columns, each cell a_{ij} ($n+1 \le j \le 2n$) is a co-occurrence frequency of word sequence $w_j w_i$. The resulting SVD matrix $\mathbf{V}$ consists of two different parts. The first n rows is used to project the 2^{nd} preceding word, and the next n rows is used to project the 1^{st} preceding word. Matrix $\mathbf{U}$ is used to project a current word. In this case, the CDC is calculated as follows:

$$P_{CDC}(w_i|w_{i-2},w_{i-1}) = P(C(w_i,\mathbf{U})|C(w_{i-2},\mathbf{V}_1),C(w_{i-1},\mathbf{V}_2))P(w_i|C(w_i,\mathbf{U})) \quad (10)$$

where $\mathbf{V}_1$ is the first n rows of matrix $\mathbf{V}$ and the rest is $\mathbf{V}_2$.

2.2.2 1-r distance Bigram Matrix

Different with bigram or trigram matrix, in this matrix we tried to collect the information about the previous word in general by accumulating the co-occurrence of r-distance bigram words. So the column in 1-r distance bigram matrix represents the preceding words $w_{i-r},...,w_{i-1}$ in general as illustrated by Figure 3.

Each cell a_{ij} is the accumulation of co-occurrence word w_i as a current word with w_j appearing from the 1^{st} preceding word to the w_j as r^{th} preceding word. The resulting SVD matrix $\mathbf{U}$ is used to projecting the current word into LSA space. While matrix $\mathbf{V}$ is used to projecting the preceding words. In this case, Equation (5) becomes

$$P_{CDC}(w_i|w_{i-n+1},...,w_{i-1}) = P(C(w_i,\mathbf{U})|C(w_{i-n+1},\mathbf{V}),...,C(w_{i-1},\mathbf{V})) \times P(w_i|C(w_i,\mathbf{U})) \quad (11)$$

Because the matrix $\mathbf{V}$ contains information about all the preceding words, not only 1^{st} or 2^{nd} preceding word in bigram or trigram matrix case, the CDC context could

Table 2: Accuracy and average rank (200 dimensions and 2000 classes).

No	Model	Closed Test						Open Test					
		Accuracy					Avrg Rank	Accuracy					Avrg Rank
		1-best	5-best	10-best	50-best	100-best		1-best	5-best	10-best	50-best	100-best	
1	Baseline	0.2924	0.5585	0.6602	0.8390	0.8890	96	0.1839	0.3806	0.4681	0.6488	0.7125	672
2	CDC-B	0.1978	0.3702	0.4547	0.6260	0.6903	497	0.1790	0.3381	0.4261	0.5946	0.6569	848
3	CDC-T	0.2349	0.4433	0.5435	0.7338	0.7989	222	0.1742	0.3383	0.4257	0.6022	0.6661	926
4	CDC-DB2	0.2372	0.4423	0.5425	0.7321	0.7963	219	0.1722	0.3319	0.4224	0.5949	0.6589	1195
5	CDC-DB3	0.2301	0.4565	0.5552	0.7443	0.8061	212	0.1684	0.3434	0.4307	0.6010	0.6620	1177
6	CDC-DB4	0.2416	0.4620	0.5626	0.7467	0.8083	217	0.1776	0.3521	0.4377	0.6041	0.6636	1212
7	CDC-B+	0.3047	0.5543	0.6523	0.8228	0.8773	108	0.2243	0.4139	0.4958	0.6630	0.7211	645
8	CDC-T+	0.2917	0.5461	0.6423	0.8191	0.8742	110	0.1961	0.3848	0.4670	0.6437	0.7035	673
9	CDC-DB2+	0.2928	0.5470	0.6450	0.8188	0.8732	110	0.1940	0.3812	0.4655	0.6399	0.7006	697
10	CDC-DB3+	0.2922	0.5489	0.6508	0.8217	0.8751	109	0.1942	0.3823	0.4706	0.6408	0.7029	697
11	CDC-DB4+	0.2941	0.5536	0.6506	0.8229	0.8756	110	0.1957	0.3867	0.4710	0.6426	0.7038	704

be extended into n-gram context without increasing cost to calculate matrix. Unless stated, the class context will always have $n = 3$. In the experiment we will also conduct the 4-gram of CDC using this matrix.

3 Experiments

The training data set was taken from WSJ corpus year 1987 consists of 17 million words. The evaluation was conducted using two data sets, the closed test set and the open one. The closed test set data was taken from training data, consists of 22 thousand words. For open test, the data set was taken from WSJ year 1988 consists of 23 thousand words. Words which occur less than 24 times are mapped into an out of vocabulary (OOV) symbol. This makes the vocabulary size 20,291 in total. The detail about experimental data is given by Table 1.

The baseline, Katz backoff trigram language model, was build using HTK Language Model toolkit[7]. The matrix representation was decomposed and reduced using SVDLIBC [1] with Lanczos method. The LSA dimension was varied from 20, 50, 100, and 200 dimensions. The clustering was conducted by VQm[2] using K-means algorithm with Euclidean distance with various numbers of classes from 100, 200, 1000, and 2000. The models are evaluated by calculating its accuracy, average rank, and perplexity according to the following equation:

$$Accuracy = \frac{\#correctly\,predicted\,word}{\#of\,sample} \quad (12)$$

$$Averagerank = \frac{\sum_W correct\,word\,position\,in\,n-best\,list}{\#of\,sample} \quad (13)$$

$$Perplexity = 2^{-\frac{1}{N}\log_2 P_L(W)} \quad (14)$$

The perplexity is calculated only on the open test set.

For the first experiment, we consider 10 CDC model, they are CDC with bigram matrix (CDC-B), CDC using trigram matrix (CDC-T), CDC with 1-2 distance bigram matrix (CDC-DB2), CDC with 1-3 distance bigram matrix (CDC-DB3), CDC with 1-4 distance bigram matrix (CDC-DB4), and model with "+" sign means that the model is interpolated with the baseline using Equation 6 with $\alpha = 0.5$. In the observed LSA dimension and class number, generally the model's performance shows better results when both values are increasing. So in this paper, we will show only some results which are representative to the model performance.

The accuracy and average rank result is given by Table 2. The baseline has better accuracy compared to other models except CDC-B+ and CDC-DB4+ on 1-best with accuracy 30.47% and 29.41% respectively. The accuracy on the open test shows that CDC-B+ accuracy is better than the baseline. Other combination (CDC+) model shows comparable results to the baseline. Evaluation on average rank also shows similar behaviour. The baseline is better in closed test, but CDC-B+ leading in the open test.

[1]http://tedlab.mit.edu/~dr/SVDLIBC

[2]http://www.dice.ucl.ac.be/ lee/software/vq/main.html

Table 3: Perplexity against number of classes (200 dimensions).

No	Model	Number of Classes			
		100	200	1000	2000
1	Baseline	172	172	172	172
2	CDC-B	431	410	389	374
3	CDC-T	446	381	356	350
4	CDC-DB2	652	668	1110	815
5	CDC-DB3	679	694	1062	809
6	CDC-DB4	675	782	1144	778
7	CDC-B+	148	147	143	142
8	CDC-T+	164	157	153	150
9	CDC-DB2+	196	185	160	155
10	CDC-DB3+	197	186	161	154
11	CDC-DB4+	197	189	160	153

Table 4: Perplexity against dimension (2000 classes).

No	Model	Dimension			
		20	50	100	200
1	Baseline	172	172	172	172
2	CDC-B	431	410	389	374
3	CDC-T	446	381	356	350
4	CDC-DB2	1066	878	969	815
5	CDC-DB3	1086	855	831	809
6	CDC-DB4	1031	913	847	778
7	CDC-B+	148	147	143	142
8	CDC-T+	164	157	153	150
9	CDC-DB2+	170	161	158	155
10	CDC-DB3+	169	160	157	154
11	CDC-DB4+	173	162	158	153

The perplexity of each model with the increasing class number is shown in Table 3. CDC-B+ and CDC-T+ give lower perplexity than the baseline. While other CDC+ model should at least have 1000 classes to achieve better perplexity. In Table 4 we can see that the interpolation model shows better perplexity than the baseline except for CDC-DB4+ with 20 dimension and 2000 classes. The best perplexity is achieved by CDC-B+ with perplexity 142. It means 17.44% relative improvement against the baseline trigram.

As shown from these results, the CDC performance is below the baseline. But when combined with the baseline the result shows that the CDC+ performance is improved and comparable with the baseline. CDC-B+ shows better performance compared to the other CDC models. It also shows smaller difference between the closed test and open test performance than the baseline.

Next, we conducted an experiment similar to CDC-B/CDC-B+. But here only one LSA matrix is used, which is only U, to project all words (either current word or preceding word) to LSA space. And after clustering, the probability is calculated using a class based language model as shown in Equation 4 (context-independent class). This model denoted as MU-B and MU-B+, where MU-B+ is

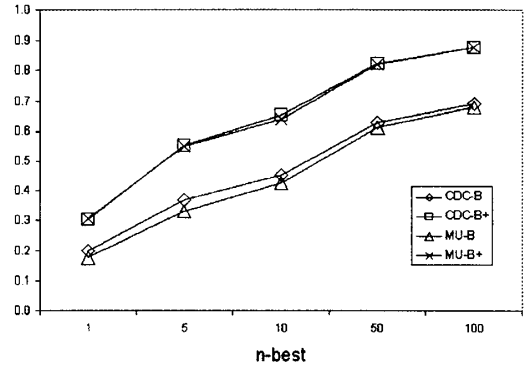

Figure 4: Accuracy of closed test

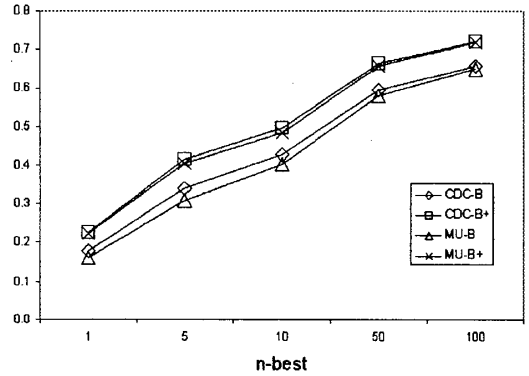

Figure 5: Accuracy of open test

an interpolation model of MU-B. The purpose of this experiment is to show that the information about the 1^{st} preceding word is contained in **V** matrix. Here we only shows the result of 200 dimensions and 2000 number of class. The comparison on accuracy of this experiment is given by Figures 4 and 5. The average rank of MU-B and MU-B+ on the closed test set is 505 and 108, and on the open test set is 881 and 658 respectively. The perplexity is 408 for MU-B and 146 for MU-B+. From these results, it is clear that CDC-B and CDC-B+ gives better performance.

In the last experiment, we conducted the 4-gram of CDC using 1-r distance bigram matrix (CDC-4DBr). The accuracy of this model is given by Table 5 for the closed test, and Table 6 for the open test. Followed by Table 7 that shows the average rank and the perplexity. By increasing the context class into 4-gram, the performance of CDC is greatly improved, especially on the closed test. The performance of CDC-4DBr is far better than the baseline on the closed test accuracy and gets even better when interpolated with the baseline. But the performance on the open test is not as good as the closed test. The perplexity of this model shows the lowest perplexity of CDC, that is 132. It means 23.26% relative improvement against the baseline.

4 Conclusions

We proposed an alternative way to calculate the language model. It has been shown that the performance differences between the closed test and open test of CDC with bigram matrix is closer than the baseline's. This result validates our goal on solving the sparseness problem. Furthermore our proposed model achieved 23.26% relative improvement on perplexity, compared to state-of-the-art statistical trigram language model.

For future works, there are still many things that can be improved, such as using another distance in VQ or changing clustering method. We also looking forward to use another extraction method such as Probabilistic LSA (PLSA).

Table 5: Accuracy of closed test (200 dimensions and 2000 classes).

No	Model	Accuracy				
		1-best	5-best	10-best	50-best	100-best
1	Baseline	0.2924	0.5585	0.6602	0.8390	0.8890
2	CDC-B+	0.3047	0.5543	0.6523	0.8228	0.8773
3	CDC-4DB2	0.3809	0.6458	0.7353	0.8867	0.9235
4	CDC-4DB3	0.3794	0.6567	0.7469	0.8911	0.9276
5	CDC-4DB4	0.3904	0.6613	0.7553	0.8921	0.9287
6	CDC-4DB2+	0.4092	0.6851	0.7768	0.9086	0.9407
7	CDC-4DB3+	0.4113	0.6933	0.7837	0.9091	0.9421
8	CDC-4DB4+	0.4140	0.6963	0.7852	0.9113	0.9435

Table 6: Accuracy of open test (200 dimensions and 2000 classes).

No	Model	Accuracy				
		1-best	5-best	10-best	50-best	100-best
1	Baseline	0.1839	0.3806	0.4681	0.6488	0.7125
2	CDC-B+	0.2243	0.4139	0.4958	0.6630	0.7211
3	CDC-4DB2	0.1772	0.3465	0.4290	0.5999	0.6594
4	CDC-4DB3	0.1742	0.3506	0.4371	0.6056	0.6622
5	CDC-4DB4	0.1836	0.3610	0.4441	0.6055	0.6639
6	CDC-4DB2+	0.1989	0.3901	0.4715	0.6407	0.7004
7	CDC-4DB3+	0.1986	0.3911	0.4756	0.6428	0.7020
8	CDC-4DB4+	0.2014	0.3956	0.4775	0.6444	0.7041

Table 7: Average Rank and perplexity (200 dimensions and 2000 classes).

No	Model	Test set		Perplexity
		Closed	Open	
1	Baseline	96	672	172
2	CDC-B+Baseline	108	645	142
3	CDC-DB2	70	1283	804
4	CDC-DB3	64	1264	802
5	CDC-DB4	68	1321	801
6	CDC-DB2+Baseline	44	707	134
7	CDC-DB3+Baseline	42	707	134
8	CDC-DB4+Baseline	42	717	132

References

[1] Terashima, S., Takeda, K., Itakura, F., "A linear space representation of language probability through SVD of N-gram matrix," Electronics and Communications in Japan, Part 3, Vol. 86, No. 8, 2003.

[2] Brown, P.F., Della Pietra, V.J., Desouza, P.V., Lai, J.C. and Mercer, R.L., "Class-based n-gram models of natural language," Comp. Linguistics, 184:467-479, 1992.

[3] Yamamoto, H., Isogai, S., Sagisaka, Y., "Multi-Class Composite N-gram Language Model for Spoken Language Processing Using Multiple Word Clusters," 39th Meeting on Association for Computational Linguistics, pp. 531-538, 2001.

[4] Afify, M., Siohan, O., Sarikaya, R.,"Gaussian Mixture Language Models for Speech Recognition," ICASSP, Hawaii, USA, Vol. IV, pp.29-32, April 2007.

[5] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., "A neural probabilistic language model," Journal of Machine Learning Research, vol. 3, pp. 1137-1155, 2003.

[6] J.R. Bellegarda, "A multi-span language modelling framework for large vocabulary speech recognition," IEEE Trans. Speech Audio Processing, vol. 6, no.5, pp. 456-457, Sept. 1998.

[7] Yung, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P., "The HTK book (for HTK Version 3.3)," Cambridge University Engineering Department, Cambridge, 2005.