On Modeling Remote and Local Dependencies in Language

Yu-Sheng Lai and Chung-Hsien Wu
Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
E-mail: {laiys, chwu}@csie.ncku.edu.tw

Abstract

In this paper, a statistical language model that can model both remote and local dependencies is proposed. This model takes into account the relationship between the predicted word and its preceding words without considering the order of the preceding words. Two primary parameters, the reliability coefficient and the combination factor, are proposed to achieve a better performance of the language model. The reliability coefficients identify the reliabilities of the remote dependencies to the predicted word. The combination factor gives a weight to the combination of the local dependency and the remote dependency.

The language model was tested on the task of word clustering and compared to the traditional N-gram language model. A large corpus provided by Academia Sinica, Taiwan, containing 5 million words was used for training and testing. The experimental results show that the proposed model takes littler computation and achieves a better performance for large N compared to the traditional N-gram language model.

1. Introduction

Statistical language models have proved useful when enough data is abailable to estimate the word probabilities. The most commonly used statistical language modeling technique is to consider the word sequence $w_1 \cdots w_Q$ as a Markov process and is termed as the N-gram language model. The traditional N-gram language model estimates the word sequence probability by the following equation

$$P(w_1 \cdots w_Q) = \prod_{n=1}^{Q} P(w_n \mid w_{n-N+1}^{n-1}) \qquad (1)$$

where w_{n-N+1}^{n-1} represents the word sequence $w_{n-N+1} \cdots w_{n-1}$ for short and the conditional probability $P(w_n \mid w_{n-N+1}^{n-1})$ indicates that the probability of the word w_n

can be predicated by its preceding N-1 words $w_{n-N+1} \cdots w_{n-1}$.

The N-gram language model has been shown that it can work very well on dealing with local dependency in language. But it takes heavy computation and large memory requirement for large N. For practical reasons, most systems use bigram or trigram only. That is, they estimate the conditional probabilities only for N=2 or 3. Thus computational complexity and memory requirement can be reduced efficiently. In this model, however, the remote dependencies will not be taken into account. That is, some grammatical structures like "if...then" clause will not be modeled.

Without caring about heavy computation and memory requirement, the conditional probability $P(w_n \mid w_{n-N+1} \cdots w_{n-1})$ strictly constrains that the predicted word w_n is related to the preceding word sequence $w_{n-N+1} \cdots w_{n-1}$ and their order. In practice, however, the word w_n is partially related to the word sequence $w_{n-N+1} \cdots w_{n-1}$ only. In other words, the word w_n is only related to some words in the word sequence $w_{n-N+1} \cdots w_{n-1}$ rather than the whole word sequence. For instance, considering the sentence "I went for a long long walk this morning," using the conditional probability $P("walk" | "go", "for", "a")$ to predict the word "walk" will be more appropriate than using $P("walk" | "go", "for", "a", "long", "long")$. The phrase "go for a walk" is a very common usage in texts but the phrase "go for a long long walk" is often used in spoken language or is an unseen event.

One of the primary difficulties encountered using the N-gram language model is the problem of sparse data. No matter how large a training corpus you have, there will always be many unseen events that will come up in testing. For this sake, many people invested in modeling unseen events [1, 2]. Smoothing methods solved the problem of sparse data only for some cases. For instance, the unseen events never appearing in real world and the unseen events resulting from incomplete collection are different, but they are viewed as the same by the smoothing methods. In our opinion, the kinds of problems should be essentially dealt with in modeling phase rather than in smoothing phase.

A· different approach in language modeling was proposed by using the technologies of class mapping [3]. For an unseen word m-gram, it is still possible to map it to a corresponding class m-gram. Because the number of model parameters such as the m-gram probabilities is reduced due to the class mapping, each parameter can be estimated more reliably. On the contrary, reducing the number of model parameters will result in a rough model with less precise prediction of the next word. It is a tradeoff between these two extremes.

In terms of linguistics, however, word equivalence class is an important concept in syntax and semantics. It is defined by linguistic experts and is called part of speech (POS). In the past years, many techniques for word clustering have been proposed [4-6]. Generally, the algorithms are based on minimum perplexity or maximum likelihood. In this paper, the most commonly used quantity, perplexity, is used to evaluate the proposed language models on the task of word clustering.

The goal of this paper is to model both remote and local dependencies in language but just requires low computation and memory requirements. We will describe the remote dependency modeling in Section 2. The proposed language model will be described in Section 3. In Section 4, we will describe how to implement word clustering efficiently by the exchange algorithm. We designed several experiments to show the performance of the language model we proposed on word clustering. We will show the experimental results in Section 5. Finally, we will make some conclusions in Section 6.

2. Remote Dependencies Modeling

The N-gram language model encounters two difficulties while estimating remote dependencies. The first one is that it takes much time in computation and requires much memory for large N. The second one is the problem of sparse data. Here, we will describe a way for modeling remote dependencies but reducing the above requirements.

2-1 Estimation of Remote Dependencies

Estimating remote dependency between two disconnected words, intuitively, can be viewed as estimating remote bigram. If there is a pair of disconnected words v and w, where v appears in front of w in the text, then computing remote bigram of v and w can be viewed as computing the conditional probability $p_R(w \mid v)$ defined as

$$D_R(v,w) \equiv p_R(w \mid v) = \frac{F_R(v,w)}{F(v)} \tag{2}$$

where $F(v)$ denotes the frequency of the word v and $F_R(v,w)$ denotes the frequency of the disconnected word pair (v, w) in the corpus.

However, the estimation of conventional bigram is not applicable to remote bigram. For each word, it counts remote dependencies in a proper range M based on the corpus. It will happen that $\sum_w p_R(w \mid v) \geq 1$ due to $\sum_w F_R(v, w) \geq F(v)$ when the range M is greater than 2. For instance, for the word sequence $v \cdots w_1 w_2$, the summation $F_R(v, w_1) + F_R(v, w_2)$ will be greater than the frequency $F(v)$ if we increase the frequencies $F(v)$, $F_R(v, w_1)$ and $F_R(v, w_2)$ by 1 respectively. To avoid this inequality, we just increase the frequency by c rather than 1 for each remote frequency $F_R(w_{n-i}, w_n), i = 2 \cdots M-1$ and c can be computed as

$$c \equiv \max\{\frac{1}{M-2}, \frac{1}{L-2}\} \tag{3}$$

where L is the number of the words from the left boundary of the sentence to the predicted word . Thus, it will keep the equal sign of the following equation

$$\sum_w F_R(v, w) = F(v) \tag{4}$$

Nevertheless the above estimation will lose some dependencies from more complex grammatical structures like "prefer to ... rather than." To avoid this problem, we can increase the degree of remote dependency by using remote m-gram rather than remote bigram. In our experiments, we model remote dependencies by using remote bigram only.

2-2 Reliability Coefficients

The remote dependency $D_R(v, w)$ is defined to represent the dependency between the predicted word w and a prior word v. Since there are several dependencies in the proper range M, it is reasonable to assign a weight for each dependency. We call them reliability coefficients. They identify the reliability of the corresponding dependency to the predicted word. The more the appearance frequency is, the better the reliability is. For a remote dependency $D_R(w_{n-i}, w_n)$, therefore, the reliability coefficient $\lambda_{i,n}$ can be estimated as

$$\lambda_{i,n} = \frac{F_R(w_{n-i}, w_n)}{\sum_{j=2}^{M-1} F_R(w_{n-j}, w_n)} \tag{5}$$

3. The Proposed Model

In this section, we will describe how to combine remote dependencies into N-gram language model. In order to solve the problem of sparse data, we categorize words into word equivalence classes and estimate unseen events by using Turing-discounted probabilities [7].

3-1 Combination of Remote and Local Dependencies

The proposed model consists of two components: the N-gram language model (N-gram) and the language model with parallel remote dependencies (PRD). These two components could be defined as follows.

- *N-gram Language Model (N-gram)*

$$P_{N-gram}(w_n \mid w_{n-N+1}^{n-1}) = \frac{F(w_{n-N+1}^{n})}{F(w_{n-N+1}^{n-1})} \tag{6}$$

- *Language Model with Parallel Remote Dependencies (PRD)*

$$P_{PRD}(w_n \mid w_{n-M+1}^{n-1}) = \prod_{i=2}^{M-1} D_R(w_{n-i}, w_n)^{\lambda_{i,n}} \tag{7}$$

Since the N-gram model considers the local dependency only, it is enough for N=2 or 3 in the combination model. The combination model named Language Model with M-Remote and N-Local Dependencies (MRNLD) consists of N-gram with small N and the language model with M parallel remote dependencies. Fig.1 shows the relationship between the predicted word and the remote and local dependencies. The language model can be defined as

$$P_{MRNLD}(w_n \mid w_{n-M+1}^{n-1}) \equiv P_{N-gram}(w_n \mid w_{n-N+1}^{n-1})^{\alpha(w_n)} \cdot P_{PRD}(w_n \mid w_{n-M+1}^{n-N})^{1-\alpha(w_n)} \tag{8}$$

where $\alpha(w_n)$ is the combination factor. It weights the N-gram language model and the language model with M parallel remote dependencies for each word w_n. We model its behavior by using a sigmoid function that can be computed as

$$\alpha(w_n) = \frac{1}{1+e^{-(l(w_n)-r(w_n))}}, \tag{9}$$

where $l(w_n)$ and $r(w_n)$ represent the local and remote log likelihood functions for the word w_n respectively. They are defined as follows

$$
\begin{aligned}
l(w_n) &= \log \prod_{w \in W} p_L(w_n \mid w)^{F_L(w,w_n)} \\
&= \sum_{w \in W} (\log F_L(w,w_n) - \log F(w)) F_L(w,w_n)
\end{aligned} \tag{10}
$$

$$
\begin{aligned}
r(w_n) &= \log \prod_{w \in W} p_R(w_n \mid w)^{F_R(w,w_n)} \\
&= \sum_{w \in W} (\log F_R(w,w_n) - \log F(w)) F_R(w,w_n)
\end{aligned} \tag{11}
$$

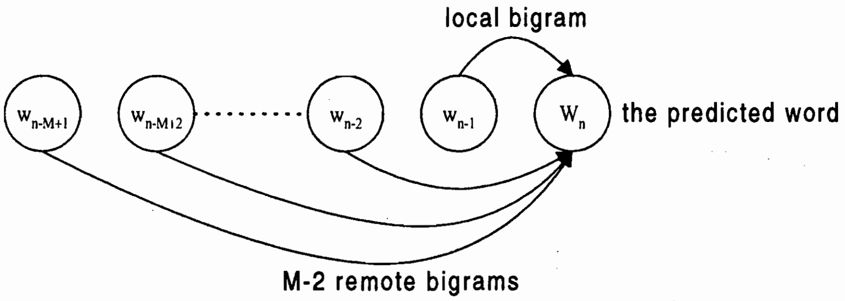

Fig.1 the relationship between the predicted word and the remote and local dependencies

3-2 Word Equivalence Class Mapping

In word clustering, we assumed that each word belongs to only one class. By this assumption, a mapping C from vocabulary W to classes G can be represented as

$$C : W \to G \tag{12}$$

and by this mapping, the bigram probability [1] can be defined as

$$P(w \mid v) \equiv P(w \mid C(w)) \cdot P(C(w) \mid C(v)) \tag{13}$$

where $P(w \mid C(w))$ denotes the membership probability of the word w and $P(C(w) \mid C(v))$ denotes the transition probability from class $C(v)$ to class $C(w)$. Then Eq.8 can be recomputed as

$$
\begin{aligned}
& P_{MRNLD}(w_n \mid w_{n-M+1}^{n-1}) \\
& = P(w_n \mid C(w_n)) \\
& \times P_{N-gram}(C(w_n) \mid C(w_{n-N+1}) \cdots C(w_{n-1}))^{\alpha(C(w_n))} \\
& \times P_{PRD}(C(w_n) \mid C(w_{n-M+1}) \cdots C(w_{n-N}))^{1-\alpha(C(w_n))}
\end{aligned} \tag{14}
$$

where the combination factor $\alpha(C(w_n))$ and the related tokens are well defined as follows:

$$\alpha(g) = \frac{1}{1 + e^{-(l(g) - r(g))}} \tag{15}$$

$$l(g) = \sum_{h \in G} (\log F_L(h,g) - \log F(h)) F_L(h,g) \tag{16}$$

$$r(g) = \sum_{h \in G} (\log F_R(h,g) - \log F(h)) F_R(h,g) \tag{17}$$

After the word clustering process, the number of unseen evens can be greatly reduced. For the remaining unseen events, the Turing-discounted probabilities [7] are adopted for further smoothing.

4. Implementation of Word Clustering

4-1 Clustering Algorithm

We use the exchange algorithm [4] in this word clustering process. The main idea of the algorithm is to find a class mapping $C : W \to G$ such that the perplexity of the language model is minimized over the training corpus, where an observation word may be exchanged from a class to another class in order to improve the criterion. In the case of language modeling, the optimization criterion is the entropy described in next subsection. The initialization method is to assign the most frequent $|G|-1$ words into their own word equivalence classes, where $|G|$ is the number of classes, and the remaining words into an additional word equivalence class.

4-2 Performance Measure

Having constructed a language model, we need to show how well the proposed language model performs in a task. It is necessary to have a method for measuring the performance. We use the perplexity to measure the performance of the MRNLD on word clustering. The formal perplexity PP is defined as [8]

$$PP \equiv P(w_1 w_2 \cdots w_Q)^{-\frac{1}{Q}} \tag{18}$$

For the MRNLD, the estimation of well-defined entropy can be decomposed in terms of frequencies as follows

$$H_P = \log PP \tag{19}$$

$$= -\frac{1}{Q} \log P(w_1 w_2 \cdots w_Q) \tag{20}$$

$$= -\frac{1}{Q} \log \prod_{n=1}^{Q} P_{MRNLD}(w_n \mid w_{n-M+1}^{n-1}) \tag{21}$$

$$=-\frac{1}{Q}\sum_{n=1}^{Q}\log P_{MRNLD}(w_n \mid w_{n-M+1}^{n-1}) \tag{22}$$

$$\begin{aligned}
=-\frac{1}{Q}\{&\sum_{n=1}^{Q}\log p(w_n \mid C(w_n)) \\
&+\sum_{n=1}^{Q}\alpha(C(w_n))\log p_L(C(w_n)\mid C(w_{n-N+1})\cdots C(w_{n-1})) \\
&+\sum_{n=1}^{Q}[(1-\alpha(C(w_n)))\sum_{i=N}^{M-1}\lambda_{i,n}\log p_R(C(w_n)\mid C(w_{n-i}))]\}
\end{aligned} \tag{23}$$

$$\begin{aligned}
=-\frac{1}{Q}\{&\sum_{w\in W}F(w)\log\frac{F(w)}{F(C(w))} \\
&+\sum_{g\in G,H\in G^{N-1}}\alpha(g)F_L(H,g)\log\frac{F_L(H,g)}{F(H)} \\
&+\sum_{n=1}^{Q}((1-\alpha(C(w_n)))\sum_{i=N}^{M-1}\lambda_{i,n}\log\frac{F_R(C(w_{n-i}),C(w_n))}{F(C(w_{n-i}))})\}
\end{aligned} \tag{24}$$

$$\begin{aligned}
=-\frac{1}{Q}\{&\sum_{w\in W}F(w)\log F(w)-\sum_{g\in G}F(g)\log F(g) \\
&+\sum_{g\in G}\alpha(g)\sum_{H\in G^{N-1}}F_L(H,g)(\log F_L(H,g)-\log F(H)) \\
&+\sum_{n=1}^{Q}((1-\alpha(C(w_n)))\sum_{i=N}^{M-1}\lambda_{i,n}(\log F_R(C(w_{n-i}),C(w_n))-\log F(C(w_{n-i}))))\}
\end{aligned} \tag{25}$$

$$\begin{aligned}
=-\frac{1}{Q}\{&\sum_{w\in W}F(w)\log F(w)-\sum_{g\in G}F(g)\log F(g) \\
&+\sum_{g\in G}\alpha(g)\sum_{H\in G^{N-1}}F_L(H,g)(\log F_L(H,g)-\log F(H)) \\
&+\sum_{g\in G}(1-\alpha(g))\sum_{n\ni C(w_n)=g}\frac{\sum_{i=N,h_i=C(w_{n-i})}^{M-1}F_R(h_i,g)(\log F_R(h_i,g)-\log F(h_i))}{\sum_{i=N,h_i=C(w_{n-i})}^{M-1}F_R(h_i,g)}\}
\end{aligned} \tag{26}$$

By Eq.26, it takes much time on computing remote dependencies due to dynamic reliability coefficients. In order to reduce the computational complexity, $\lambda_{i,n}$ is chosen as a constant, $\frac{1}{M-N}$. It means that the reliabilities for all remote dependencies are equal. Then Eq.26 can be rewritten as

$$
\begin{aligned}
H_p = -\frac{1}{Q}\{ & \sum_{w\in W} F(w)\log F(w) - \sum_{g\in G} F(g)\log F(g) \\
& + \sum_{g\in G}\alpha(g) \sum_{H\in G^{N-1}} F_L(H,g)(\log F_L(H,g) - \log F(H)) \\
& + \sum_{g\in G}(1-\alpha(g))\sum_{h\in G} F_R(h,g)(\log F_R(h,g) - \log F(h))\}
\end{aligned}
\tag{27}
$$

5. Experimental Results

In this section, we will show the experimental results for the word clustering process. The test corpora, ASBC (Academia Sinica Balanced Corpora), were provided by Academia Sinica, Taiwan. We tested on four aspects: The first one is model testing. It tests on three models: the traditional N-gram language model, the language model with M parallel remote dependencies, and the proposed model MRNLD. The second one is the testing for CPU time. It compares the CPU time in word clustering by using different language models: the class trigram language model and the MRNLD. The third one is parameter testing. It tests the reliability coefficient $\lambda_{i,n}$ and the combination factor α. The fourth one is corpus test including inside test and outside test. All of these tests evaluate the performance by perplexities.

5-1 Corpora

ASBC consists of several corpora that were collected and tagged by Institute of Information Science, Academia Sinica. It contains 5 million words and a vocabulary of 130,000 words including common words, proper nouns and compound words. In our experiments, we chose about 27,000 most frequent words as the vocabulary.

In the word clustering process, we predefined 6 classes. The first two classes consist of one word respectively. The first two classes are "iou3" (有) and "shz4" (是) and their grammar behaviors are very complex, so we pre-clustered them into 2 classes respectively. The third class consists of 4 words: "de" (的), "jr" (之), "de" (得), and "de" (地) due to their special functions. The fourth class collects all borrowed words from foreign languages in the corpora. The fifth class collects those out-of-vocabulary words. The sentence boundary was viewed as a word and pre-clustered into the sixth class.

5-2 Word Clustering Experiments

In the experimental results, the traditional trigram language model is abbreviated to trigram, the language model with 3 parallel remote dependencies is abbreviated to 3-PRD, and the language model with 3 remote and 2 local dependencies is abbreviated to 3-R-2-LD. Additionally, 3-PRD is defined as 3-R-2-LD with the local degree (N) being 1.

5-2-1 Model Test

Table 1 shows perplexities of trigram, 3-PRD, and 3-R-2-LD. In this experiment, we tested on remote degree of 3, dynamic combination factors, and static reliability coefficients. We used the whole corpus of 5 million words in testing. However, since trigram needs large computation, it was just tested on cluster numbers of 50, 100, and 200. The results show the language model with 3 remote and 2 local dependencies is better than the traditional trigram language model in word clustering.

Table 1. Perplexities for different models with different numbers of word equivalence classes

No. of Classes L. M.	50	100	200	500	1000	2000
Trigram	247.63	212.58	182.38	-	-	-
3-PRD	215.23	195.46	160.78	136.92	108.44	95.45
3-R-2-LD	201.39	173.85	135.26	112.45	89.86	78.93

Table 2 shows perplexities of PRD and MRNLD with different remote degrees (M) from 3 to 8 and a fixed local degree (N) being 2. In this experiment, we clustered the whole corpus of 5 million words into 50 classes by using dynamic combination factors and static reliability coefficients. The results show that the perplexities of both two models decrease as the remote degrees increase and MRNLD performs better than PRD.

Table 2. Effect of remote degree (M) for different models

M L. M.	3	4	5	6	7	8
PRD	215.23	208.91	199.73	193.28	205.63	211.37
MRNLD (N=2)	201.39	196.54	188.49	185.10	190.57	196.25

5-2-2 CPU Time Test

Table 3 shows the CPU time per iteration by using the 3-R-2-LD model and the trigram model on word clustering and the result shows that the 3-R-2-LD model is more efficient than the trigram model. This experiment is tested on the corpus of 5 million words. Due to large computations of trigram, we tested only on cluster numbers of 50, 100, and 200.

Table 3. CPU time (minutes per iteration) for clustering algorithm on different models

No. of Classes L. M.	50	100	200	500	1000	2000
Trigram	172	340	1035	-	-	-
3-R-2-LD	115	230	621	2016	5138	13740

5-2-3 Parameter Test

To reduce the computational complexity, we simplified the dynamic reliability coefficients to be static ones. We want to know the simplification effect in this experiment. Additionally, due to the large computation in testing on the dynamic reliability coefficients, we used a small corpus that is only part of the ASBC and it is also clustered into 50, 100, and 200 classes. The downsized corpus consists of 1 million words. Table 4 shows the experimental results. The static reliability coefficients are better than the dynamic ones. This seemly contradicts to our expectation. A reasonable explanation is the problem of data sparseness.

Table 4. Perplexities for dynamic and static reliability coefficients (λ)

No. of Classes λ	50	100	200	500	1000	2000
Dynamic	245.86	197.41	158.03	-	-	-
Static	223.07	185.15	149.79	116.93	95.23	87.74

The combination factor α is dynamic and defined by a sigmoid function. The MRNLD is the combination of the N-gram and PRD, the combination factor determines whether the N-gram model is more important than PRD or not. From Table 5, we know that sometimes N-gram is more important than PRD but sometimes not. It depends on classes. The corpus used in this experiment consists of 5 million words.

Table 5. Effect of combination factor (α) on the number of classes

No. of Classes / α	50	100	200	500	1000	2000
0.25	283.97	273.34	254.37	245.18	223.64	209.84
0.5	269.51	250.49	212.49	204.31	179.57	168.35
0.75	254.66	225.04	197.43	164.25	144.59	123.88
Dynamic	201.39	173.85	135.26	112.45	89.86	78.93

5-2-4 Corpora Test

A successful language model should be applied to any other corpora. So we divided the corpora into two groups of 1 and 4 million words. Let the small one be the training corpus and the big one be the test corpus. Table 6 and 7 show the experimental results. The same as our expectation, the results of the outside test are somewhat worse than the inside test. Besides, both of these two tests show that the language model with the remote degree of 6 has the best performance.

Table 6. Perplexities on inside test

No. of Classes / M	50	100	200	500	1000	2000
3	223.07	185.15	149.79	116.93	95.23	87.74
4	218.21	179.48	144.83	113.46	93.57	82.06
5	209.32	176.25	137.68	105.30	90.37	80.64
6	207.58	172.79	136.51	102.22	88.24	79.62
7	212.03	188.16	145.22	107.97	92.75	85.34
8	220.57	190.62	146.13	112.68	93.06	88.29

Table 7. Perplexities on outside test

No. of Classes / M	50	100	200	500	1000	2000
3	252.12	228.06	197.00	160.94	133.76	125.63
4	244.56	216.83	186.29	157.20	130.94	120.39
5	243.67	215.98	175.64	147.26	126.70	116.52
6	237.06	208.34	174.17	139.56	117.53	110.02
7	252.78	222.03	185.63	153.64	125.64	118.08
8	269.74	224.76	190.49	157.89	134.80	120.24

6. Conclusions

In this paper, we proposed a word equivalence class based language model that can model both remote and local dependencies. This model takes into account the relationship between the predicted word and its preceding words without considering the order of the preceding words. Although this model considers the remote dependency and the local dependency simultaneously, it requires littler computation than the traditional class-based N-gram language model on word clustering task and achieves a better performance for large N.

Two primary parameters, the reliability coefficient and the combination factor, are proposed to achieve a better performance of the language model. According to the experimental results, the language model achieves the best performance on static reliability coefficients and dynamic combination factors.

References

[1] S. M. Katz, "Estimation of Probabilities from Sparse Data for The Language Model Component of A Speech Recognizer," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 35, no. 3, March 1987, pp. 400-401.

[2] F. Jelinek and R. L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," Pattern Recognition in Practice, North Holland, 1980, pp. 381-397.

[3] S. Martin, J. Liermann, and H. Ney, “Algorithms for bigram and trigram word clustering,” Speech Communication, 1998, pp. 19-37.

[4] R. Kneser and H. Ney, "Improved Clustering Techniques for Class Based

Statistical Language Modeling," Proc. 3rd European Conference on Speech Communication and Technology, 1993, Berlin, pp. 973-976.

[5] P. F. Brown, V. J. Della Pietra, P. V. de Souza, J. C. Lai, R. L. Mercer, "Class Based N-gram Models of Natural Language," Computational Linguistics 18 (4), 1992, pp. 467-479.

[6] M. Jardino, G. Adda, "Automatic Word equivalence classification Using Simulated Annealing," Proc. 3rd European Conference On Speech communication and Technology, 1993, Berlin, pp. 1191-1194.

[7] I. J. Good, "The Population Frequencies of Species and The Estimation of Population Parameters," Biometrika 40, December 1953, pp. 237-264.

[8] Lawrence Rabiner and B. H. Juang, "Fundamentals of Speech Recognition," Prentice Hall, pp. 449-450.