http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (16): 3685-3691, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.3685.3691

Evaluations on Several Smoothing Methods for Chinese Language Models

[1]Huang Feng-Long, [2]Yu Ming-Shing and [2]Hwang Chien-Yo
[1]Department of Computer Science and Information Engineering, National United University, MaioLi, 360, Republic of China
[2]Department of Computer Science and Engineering, National Chung-Hsing UniversityTaichung 402, Republic of China

Abstract: In this study, several smoothing methods for language models on Chinese corpus with various sizes are evaluated and analyzed. Basically, there are two phases for smoothing procedures (1) Discounting and (2) Redistributing. Ten models are generated on various size of corpus from 30-300 M Chinese words. We evaluated several smoothing methods for statistical language models. In our experiments, four smoothing methods, Winter-Bell C (WB-C) and our proposed YH-A and YH-B smoothing method, are evaluated for inside testing and outside testing. Based on empirical observations, our YH-B smoothing is superior to WB-C for the TrM models with size between 30 and 90 M.

Key words: Evaluation, language model, smoothing method, cross entropy, perplexity

INTRODUCTION

In many domains of Natural Language Processing (NLP); such as machine translation (Brants *et al.*, 2007) and speech recognition (Jelinek, 1997); the statistical Language Models (LMs) (Naptali *et al.*, 2010) always plays an important role.Data sparseness has been always an inherited issue of statistical language models and the smoothing method is usually used to resolve the zero count problems for unknown events.As shown in Fig. 1 of a speech recognition system, the P(W) is the conditional probability of a word sequence W given a speech data S, where $W = w_1w_2w\ldots w_m$ is a possible translation of texts, m is word number of M. The sequence w can be predicted as a final target.

Language models: The statistical language models have been widely used in NLP. Supposed that $W = w_1, w_2, w_3, \ldots, w_n$, where w_i and n denote the the ith

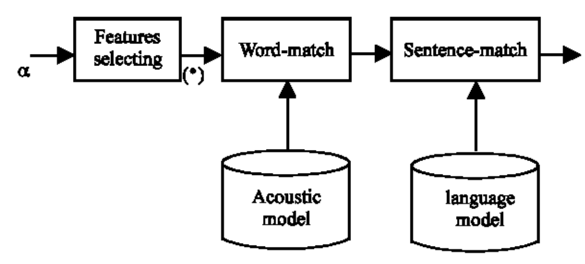

Fig. 1: LMs in a speech recognition system

Chinese character and its number in a sentence $(0 \le i \le n)$。 $P(W) = P(w_1, w_2, \ldots, w_n)$, the probability can be calculated by using chain rules:

$$P(w_1 n) = P(w_1)P(w_2|w_1)P(w3|w_1^2)\ldots P(wn|w_1^{n-1}) = \Pi^n_{k=1} P(wn|w_1^{k-1}) \quad (1)$$

where, $w_1^{k-1}, w_2, w_3, \ldots w_{k-1}$.

In general, unigram, bigram and trigram (3< = N) are generated. N-gram model calculates P(.) of Nth events by the preceding N-1 events, rather than the string $w_1, w_2, w_{3\ldots} w_{N-1}$.

N-gram models: Basically, N-gram is so-called N-1) th-order Markov model which calculate conditional probability of successive events: calculate the probability of Nth event while preceding (N-1) event occurs.

Basically, N-gram Language Model is simplified expressed as follows:

$$P(w^n_1) \approx \Pi^n_{k=1} P(w_k|w_{k-N+1}^{k-1}) \quad (2)$$

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}w_i)}{\sum_w C(w_{i-1}w)} \quad (3)$$

where, C(w) denotes the counts of event *w* occurring in dataset.

In Eq. 3 above, the obtained probability P(.) is so called Maximum Likelihood Estimation (MLE). The

Correspondence Author: Yu, Ming-Shing, Department of Computer Science and Engineering, National Chung-Hsing University, Taichung, 402, Republic of China

category with maximum probability $P_{max}(\bullet)$ will be the target and then the correct pronunciation with respect to the polyphonic character can be decided further.

Unknown events-zero countissue: As shown in Eq. 3, C(.) of a novel (a unknown event) which don't occur in the training corpus, may be zero because of the limited training data, infinite language and its expansion of language. It is always a hard work for us to collect sufficient datum. The potential issue for MLE is that the probability for unseen events is exactly zero. This is so-called the zero-count problem (Witten and Bell, 1991; Katz, 1987). It is obvious that zero count always leads to the issue of zero probability of P(.) in Eq. 2-3. Therefore, the smoothing methods are needed and exploited to alleviate the zero-count issue for statistical language models.

PROCESSES OF SMOOTHING METHODS

As described above, the zero count issue of unknown events will lead to the degradation of language models;therefore we need the smoothing methods to alleviate the situation. The idea of smoothing processes is to adjust the total probability of seen events to that of unseen events, leaving some probability mass (so-called escape probability, P_{esc}) for all the unseen events.

Smoothing algorithms (Jurafsky and Martin, 2000; Gale and Geoffrey, 1995) can be considered as discounting some counts of seen events in order to obtain the escape probability P_{esc}. And then P_{esc} will be assigned into unseen events based on the smoothing algorithm. The adjustment of smoothed probability for all possibly occurred events involves discounting and redistributing processes.

Discounting process: Based on the statistical feature, the probability of all seen and unseen (unknown) events is summed to be unity (one). First operation of smoothing method is the discounting process which discount the probability of all seen events. It means that the probability of seen events will be decreased a bit. In the process, there are two issues:

- How to discount the probability of seen events with various count c, c> = 1. Whether the discounted probability from the seen events with count c is uniform or not will affect the performance of statisticallanguage models
- The effectiveness between the size of escape probability and performance of language models

The adjustment processes can be usually divided into two categories: static and dynamic. Static smoothing methods, forthe most smoothing methods, discount the probability of events based on the events occurrences in models. However, dynamic smoothing method, i.e., cached-based language, discounts the probability based on the occurrencesfor allseen events inboth cacheand models.

Redistributing process: In this operation of smoothing algorithm,the escape probability discounted from all seen events will be redistributed to unseen events. The escape probability is usually shared by all the unseen events. That is, the escape probability is redistributed uniformly to each unseen event, P_{ESC}/U, where U is the number of unseen events. On the other hand, each unseen event obtains same probability.

The redistribution of most well known smoothing methods, such as Add-one, Absolute discounting, Good-Turing (Gale and Geoffrey, 1995; Good, 1953), Delete interpolation, Back-off (Kneser and Ney, 1995) and Witten-Bell smoothing (Ostrogonac *et al*., 2013) is uniform for all unseen events. It is a possible factor that affects the performance of smoothing algorithm. There are few previous works to discuss how to redistribute the escape probability.

SMOOTHING METHODS

In the Section, several well-known smoothing methods will be presented. We also proposed two novel methods. All these methods will be evaluated in next section.

Witten-bell method: In the study, we discussed two of five smoothing schemes: methods A and C (called WB-Aand WB-C), introduced by Wetten-Bell[1] (Ney and Essen, 1991). Previous study was in (Ostrogonac *et al*., 2013).

Method A: In this method, just one count is allocated to the probability that an unseen bigram will occur next. The probability mass P_{mass} assigned to all unseen bigrams can be summed up to 1/(N+1). The smoothed probability P* can be expressed as:

$$P^*_{i,N}(w_{i-1}w_i) = \begin{cases} \dfrac{1}{U(N+1)} & \text{for } c(w^i_{i\text{-}1}) = 0, \\ \dfrac{c(w^i_{i-1})}{N+1} & \text{for } c(w^i_{i\text{-}1}) \geq 1, \end{cases} \tag{8}$$

Method C: It is more complex than the additive discount technique. The basic concept is recurring.

The W-BC is described as:

$$c_i^* = \begin{cases} \frac{S}{U}\frac{N}{N+S}, & \text{if } c_i = 0 \\ c_i\frac{N}{N+S}, & \text{if } c_i > 0 \end{cases} \tag{9}$$

where, U, S and N denote the types of all possible unseen bigrams, seen bigrams in training corpus and the number of all the seen bigrams in training corpus, respectively.

The discounted probability will be expressed for seen bigrams as:

$$P_{i,N}^* = \frac{c_i}{N+S}, \text{ if } c_i > 0 \tag{10}$$

where, the probability mass P_{mass} for all unseen bigrams assigned by W-B C is obtained as following:

$$\sum_{i:c_i=0} P_{i,N}^* = \frac{S}{N+S}, \text{ if ci} = 0 \tag{11}$$

where, the probability for each unseen bigram will be derived from Eq. 11 divided uniformly by U:

$$P_{0,N}^* = \frac{1}{U}\frac{S}{N+S} \tag{12}$$

where, as shown in Eq. 12, it is obvious that the redistributed count c* for each bigram which doesn't appear in corpus is equal to S/U. On other word, the size of c* is subject to the ratio of S and U. The ratio may be greater or less than 1, depending on the value of S and U.

Yu-huang A(YH-A):

Basic concept: In case for a bigram,our method YH-A calculates the smoothed probabilities as:

$$Q(w_{i-1}w_i) = \begin{cases} \frac{d_A}{U(N+1)} & \text{for } c(w_{i-1}^i) = 0, \\ \frac{c(w_{i-1}^i)}{N}\frac{N+1-d_A}{N+1} & \text{for } c(w_{i-1}^i) \geq 1, \end{cases} \tag{13}$$

where, d_A denotes a constant ($0<d_A<1$) and independent of U.

When computing the smoothed probability, our proposed method don't employ interpolating scheme to combine the high order models and lower order models. As shown of Eq. 13, $(N+1\text{-}d_A)/(N+1)$ is the normalization factor for Q* of seen bigrams. The probabilities for all the seen bigrams will be discounted by the normalization factor and then the accumulated probability then is re-distributed to the unseen bigrams. All the unseen bigrams will share uniformly the distribution mass $d_A/(N+1)$:

$$\sum_{i:c_i=0} P_i^* = \frac{d_A}{N+1}, \text{ for } c_i = 0 \tag{14}$$

where, Eq. 14 of Y-H Ais similar to Eq. 11 of WB-Ain (Ney and Essen, 1991). Instead of the constant 1 of numerator in Eq. 15, it is replaced with a constant d_A ($0<d_A<1$.) It is necessary that we will evaluate d_A with respect to perplexity for language models in the next section. Hence, the better d_A for lower perplexity can be found.

Yu-Huang methodsmoothing (YH-B): Our proposed smoothing methodYu-Huang B (YH-B) describes other smoothing scheme; in which the probability mass for unseen bigrams is assigned $Ud_B/(N+1)$. Consequently, it varied with N and U; the number of training data and types of unseen bigrams.

The basic concept of our smoothing YH-B can be described in detail as follow. The smoothed probabilities will be calculated as follows:

$$P(w_{i-1}w_i) = \begin{cases} \frac{d_B}{(N+1)}, & \text{for } c(w_{i-1}^i) = 0, \\ \frac{c(w_{i-1}^i)}{N}\frac{N+1-Ud_B}{N+1}, & \text{for } c(w_{i-1}^i) \geq 1, \end{cases} \tag{15}$$

$$d_B < \min\{\frac{N}{N+2U}, \frac{N+2}{2U}\} \tag{16}$$

where, calculating the smoothed probability P*, our proposed method don't employ interpolating scheme to combine the high order models with lower order models. As shown of Eq. 13, $(N+1\text{-}Ud_B)/(N+1)$ is the normalization factor of Q* for seen bigrams. The probabilities Q will be discounted by the normalization factor and then remained Q* are redistributed to unseen bigrams; which share uniformly the distributed probability mass $Ud_B/(N+1)$:

$$\sum_{i:c_i=0} P_i^* = \frac{Ud_B}{(N+1)}, \text{ for } c_i = 0 \tag{17}$$

Models evaluation-cross entropy and perplexity: Two commonly used schemes for evaluaitng the quality of language model LM arereferred to the entropy and perplexity (Ostrogonac *et al*., 2013; Brown *et al*., 1992).

Supposed that a sample T is consisted of several events e_1, e_2, …,e_m of mstrings. The probability P for a given testing sample T is calculated as following:

$$P(T) = \prod_{i=1}^{m} P(e_i) \tag{18}$$

where, $P(e_i)$ is the probability for the event e_i and E(T) can be regarded as the coded length in testing datasets:

$$\begin{aligned} E(T) &= -\sum_{x} P(x)\log_2 P(x) \\ &= -\sum_{i=1}^{m} P(e_i)\log_2 P(e_i) \end{aligned} \tag{19}$$

$$PP(T) = 2^{E(T)} \tag{20}$$

where, E(T) and PP(T) denote the entropy (log model probability) and perplexityfor testing dataset T, respectively. E_{min} stands for the minimum entropy for a model.

The perplexity PP is usually regarded as the average number for selected number which will be the possible candidates referred to a known sequence.When a language model is employed to predict the next appearing word in the current given context, the perplexity is adopted to compare and evaluate n-gram statistical language models.

In general, lower entropy E leads to lower PP for the language models. It means that the lower PP, the better performance of language models.Therefore, perplexity is a quality measurement for LM.While two language models, LM_1 and LM_2, are compared, the onewith lower perplexity is the better languagerepresentation and commonly provides higher performance.

In fact, the probability distribution for testing language models is usually unknown. The model which can predict better the next occurring event always achieves lower cross entropy. In general CE> = E, E denotes the entropy using same language model M for training and testing models. Based on the Shannon-McMillan-Breiman theorem (Algoet and Cover, 1988; Antti, 2013), PP Evaluation can be expressed as following:

$$CE(p,M) = \lim_{n\to\infty}\frac{1}{n}\log M(w_1 w_2 w_3 \ldots w_n) \tag{19}$$

EXPERIMENTS AND EVALUATION

Training chinese corpus-textual gigaword: Chinese GigaWord (CGW) is the Chinese corpus collected from several world news databases and issued by Linguistic Data Consortium (LDC). In the study, we adopted the CGW 3.0 published on September 2009. The CGW sources are Agence France-Presse, Central News Agency of Taiwan, Xinhua News Agency of Beijing and Zaobao Newspaper of Singapore.

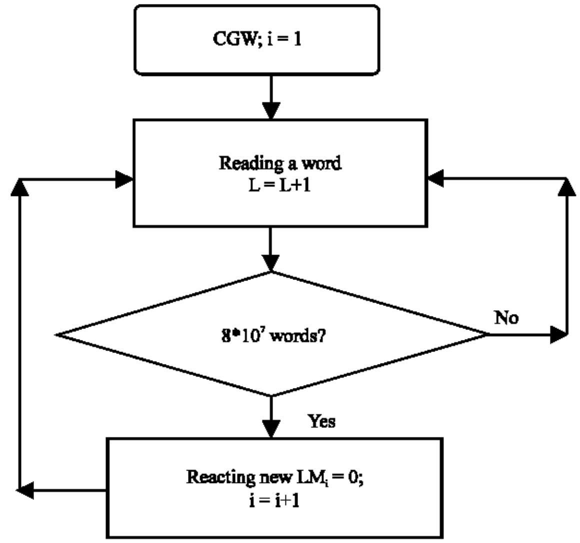

Fig. 2: Procedure for creating 10 models inour experiments

Models generation for evaluations: In the study, we will create 10 Unigramlanguage models with Chinese words for experiments. At first, we read in randomly the study of Chinese words from CGW corpus, a language model LM_1 will be created for the first 3×10^7 (30M) Chinese words.

In the following, the other new model LM_2 can be created consequently for the next 3×10^7 Chinese words. In other words, LM_2 is consisted of first 6×10^7 (60M) Chinese words of CGW, first half of which is also used to create LM_1.

Empiricalevaluation of inside testing: In the study, the 10 language models created by different size of corpus are evaluated sequentially for inside testing on these 10 models. As presented in Table 1, the x-axis and y-axis present the training model (TrM) and testing models (TeM), respectively. For each row in Table 2, testing models are used forevaluating 10 training models TrM. On the other side, 10 testing models TeM will be used, respectively to evaluate one of 10 training models for WB- C smoothing. Figure 3 present the results on 3 dimensions respect to Table 1.

WB-C smoothing, testing models, shown in Table 1, are used for evaluating 10 training models TrM. Figure 3 presents the results of perplexity PP of WB-C.

Table 1: Perplexityfor WB-C smoothing method

TrM/TeM	30M	60M	90M	120M	150M	180M	210M	240M	270M	300M
test 30M	4046	4127	4199	4256	4308	4357	4401	4442	4488	4529
test 60M	4227	4088	4131	4176	4220	4264	4303	4342	4384	4422
test 90M	4494	4307	4236	4263	4299	4335	4369	4404	4442	4478
test 120M	4707	4499	4395	4357	4378	4407	4435	4466	4501	4534
test 150M	4912	4693	4568	4506	4478	4495	4519	4546	4577	4608
test 180M	5107	4878	4738	4662	4615	4594	4609	4631	4659	4687
test 210M	5270	5034	4884	4798	4742	4705	4688	4702	4725	4749
test 240M	5428	5187	5027	4933	4869	4822	4788	4772	4787	4807
test 270M	5610	5363	5192	5089	5017	4960	4914	4883	4863	4876
test 300M	5785	5532	5352	5242	5163	5099	5042	5000	4966	4951
Avg.	4959	4771	4672	4628	4609	4604	4607	4619	4639	4664

Table 2: Perplexityfor YH-B smoothing method

TrM/TeM	30M	60M	90M	120M	150M	180M	210M	240M	270M	300M
test 30M	4102	4142	4199	4250	4297	4344	4385	4425	4470	4509
test 60M	4264	4102	4132	4169	4210	4251	4288	4325	4366	4403
test 90M	4520	4317	4236	4257	4288	4322	4354	4387	4424	4459
test 120M	4725	4505	4395	4350	4368	4393	4420	4449	4483	4515
test 150M	4923	4696	4568	4502	4467	4482	4503	4528	4558	4588
test 180M	5112	4878	4738	4659	4608	4580	4593	4613	4640	4667
test 210M	5270	5033	4884	4796	4737	4695	4672	4684	4705	4729
test 240M	5421	5183	5027	4933	4868	4817	4777	4754	4768	4786
test 270M	5596	5355	5192	5091	5020	4961	4910	4872	4843	4855
test 300M	5762	5521	5352	5247	5170	5104	5045	4998	4954	4930
Avg.	4970	4773	4672	4625	4603	4595	4595	4604	4621	4644

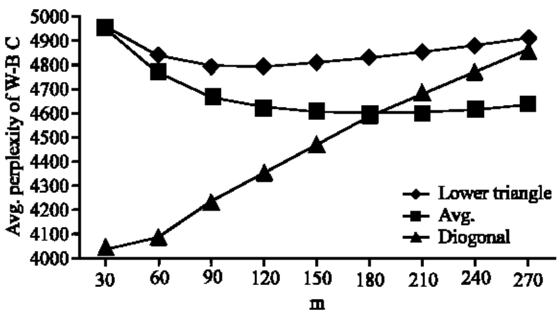

Fig. 3: Results of perplexity PP of WB-C smoothing

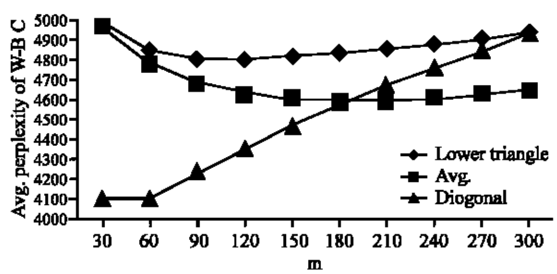

Fig. 4: Results of perplexity PP of YH-B smoothing

The smaller size of testing models, the lower of perplexity and the larger size of training models, the higher of perplexity. The lowest PP for each row is on the diagonal linein the table above.

In Fig. 3, the results of perplexity PP of WB-C smoothing are presented. Two observations are as following:

- The lowest PP was achieved on TrM_{120M}, for average PP of lower triangle of each training models
- The lowest PP was also achieved on the model TrM_{180M}, for average PP of each training models. The perplexity for TrM with size larger than 180M words will be also gradually increased

In the study, the third one is our proposed smoothing method YH-B smoothing. As displayed in Table 2and Fig.4, the lowest PPfor YH-B experiments on model TrM_{30M} and TrM_{60M} are same, 4103. It is obvious the trend is also same as that of two other smoothing methods above.

Based on the evaluation results of PP for two smoothing methods, we could conclude that, in general case for average perplexity, the lowest PP can be achieved on model TrM_{120M}. The experiment results couldprove that the model which was created on larger than 180M corpus can't achievea better performance. On the other hand, our experiments supported that the model with middle size of corpus of 180M Chinese words can always achieve the best performance of language model.

We furthermore consider the PP differencesfor two smoothing methods. As shown in Fig. 5, the trend of PP for these methods is almost same. Totally,the YH-B smoothing performwell a bit than others WB-C methods for all training models. Note that the perplexities of YH-A and WB-A are all higher than WB-C and YH-B and results therefore do not be displayed in the study.

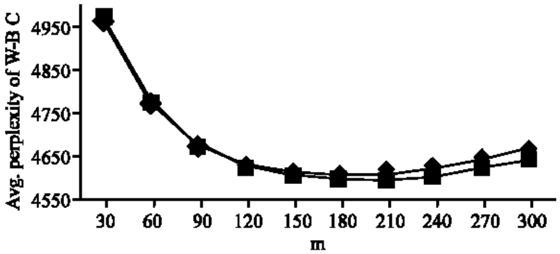

Fig. 5: Trend of PP for four smoothing methods

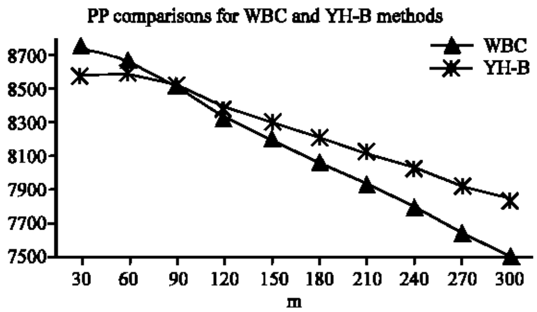

Fig. 6: PP Comparisons for WB-C and YH-B methods

Table 3: Distribution of chinese characters, word and papers of testing corpus in the paper

Topics	No. of words	No. of characters	No. of papers
Liteature	777050	1169801	1385
Living	858750	1398791	2301
Society	1610997	2711720	3246
Science	629838	1054738	994
Philosophy	439955	673080	695
Arts	474340	781415	518
Others	101394	160306	89
Total	4892324	7949851	9228

Empiricalevaluation of outside testing: In the following experiments, the text sources from ASBC corpus isexploited as outside datasets. The Academic Sinica Balanced Corpus version 3.0 (ASBC) includes 9228 text files distributed in different fields, occupying 118MB and near5 millions of Chinese words labeled with POS tag. The contents and study distributions of ASBC are listed in Table 3.

Ten Chinese language models LM_1, LM_2, to LM_{10} which contains different size of Chinese words from CGW 3.0 described in Section 3.0, will be evaluated for outside testing. In our experiments the perplexity of each method is calculated on 10 models and then we compared for these methods.

Finally, the perplexity distributions of four smoothing methods are presented in Fig. 6. Several observationsare listed below:

- The largersize of models, the lower perplexity for all the TrM models. It is apparent that the models with larger size of corpus will alleviate the issue of data sparseness. In general, the conclusion matches the statistical features
- YH-B smoothing is superior to WB-C smoothing methodsfor the TrM models with size between 30M and 90M only and degrades on larger models. We conclude that YH-B will perform well for smaller size of models in which the unknown events will occur frequently

CONCLUSION

In the study, we evaluated several smoothing methods for statistical language models. These models are created on various size of corpus,between 30M and 300M Chinese words of CGW. Several smoothing methods, Winter-Bell A andC and two our proposed YH-A and YH-B smoothing, are all evaluated. Our YH-B smoothing is superior to other smoothing methods for the TrM models with size between 30M and 90M.Based on Several observations, weanalyzed furthermore the empiricalresults which is helpful for employingthe effective smoothing methods to alleviate theissue of data sparsenesson various size of training corpus.

REFERENCES

Algoet, P.H. and T.M. Cover, 1988. A sandwich proof of the shannon-mcmillan-breiman theorem. Ann. Probab., 16: 899-909.

Antti, K., 2013. An application of ergodic theory: The shannon-mcmillan-breiman theorem. Master Thesis, University of Helsinki, Department of Mathematics and Statistics Applied mathematics.

Brants, T., C.A. Popat, P. Xu, F.J. Och and J. Dean, 2007. Large language models in machine translation. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech, pp: 858-867.

Brown, P.F., V.J. DellaPietra, R.L. Mercer, S.A. Della Pietra and J.C. Lai, 1992. An estimate of an upper bound for the entropy of English. Comput. Ling., 18: 31-40.

Gale, W.A. and S. Geoffrey, 1995. Good-turing frequency estimation without tears. J. Quant. Ling., 2: 217-237.

Good, I.J., 1953. The population frequencies of species and the estimation of population parameters. Biometrika, 40: 237-264.

Jelinek, F., 1997. Automatic Speech Recognition-Statistical Methods. MIT Press, Cambridge, MA.

Jurafsky, D. and J. Martin, 2000. Speech and Language Processing. Prentice Hall, Upper Saddle River, NJ., USA.

Katz, S., 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans. Acoustics Speech Signal Process., 35: 400-401.

Kneser, R. and H. Ney, 1995. Improved backing-off for m-gram language modeling. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Vol. 1, May 9-12, 1995, Detroit, MI, pp: 181-184.

Naptali, W., M. Tsuchiya and S. Nakagawa, 2010. Topic-dependent language model with voting on noun history. ACM Trans. Asian Lang. Inform. Process., Vol. 9. 10.1145/1781134.1781137.

Ney, H. and U. Essen, 1991. On smoothing techniques for bigram-based natural language modelling. IEEE International Conference on Acoustic, Speech and Signal Processing, April 14-17, 1991, Toronto, Ont, pp: 825-828.

Ostrogonac, S., B. Popovic, M. Secujski, R. Mak and D. Pekar, 2013. Language model reduction for practical implementation in LVCSR systems. Infoteh-Jahorina, 12: 391-394.

Witten, L.H. and T. Bell, 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. IEEE Trans. Inform. Theory, 37: 1085-1094.