An Approach to Improve the Smoothing Process Based on Non-uniform Redistribution

Feng-Long Huang
Department of Computer Science and Information Engineering
National United University
MaioLi, 360, Taiwan
flhuang@nuu.edu.tw

Ming-Shing Yu *
Department of Computer Science,
National Chung-Hsing University,
Taichung 402, Taiwan
msyu@nchu.edu.tw

Abstract

In the paper, an effective technique, based on the non-uniform redistribution probability for novel events (the unknown events), to improve the smoothing method in language models is proposed. Basically, there are two processes in the smoothing methods: 1) discounting and 2) redistributing. Instead of uniform probability assignment to each unseen events used by most smoothing methods, we propose new technique to improve the redistribution process. Referring to the probabilistic behavior of all seen events, the redistribution process for novel events in our method is non-uniform. The proposed technique is exploited on well-known and frequently-used Good-Turing smoothing method. The empirical results are demonstrated and analyzed for two n-gram models. The improvement is apparent and effective for smoothing methods, especially on higher unseen event rate.

Keywords: Language model, Smoothing method, Good-Turing, Cross entropy, Non-uniform Redistribution.

1. Introduction

1.1 Statistical language Models

In many domains of natural language processing (NLP); such as speech recognition [1], grammar parser [4], document retrieval [17] and machine translation [Brown]; the statistical language models (LMs) [6], [10] plays an important role in natural language processing. The LMs can be exploited, for instance, to decide the correct target word sequence $\overline{w}$. 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 $W=w_1w_2w...w_m$ is a possible translation of texts, *m* is word number of *M*. The predicted sequence $\overline{w}$ can be expressed:

$$\overline{W} = \arg\max_{w} P(W \mid S) = \arg\max_{w} P_{\partial}(S \mid W) P(W) \qquad (1)$$

where $P_{\partial}(S \mid W)$ is the probability of input speech given a word sequence *W*.

A language model is regarded as the probability distribution over events or token sequences (texts) that models how often each sequence occurs as a sentence. Chain rule is used to decompose probability prediction:

* Correspondence author.

$$P(w_1^m) = P(w_1 w_2 .. w_m) = P(w_1)P(w_2 \mid w_1^1)P(w_3 \mid w_1^2)...P(w_m \mid w_1^{m-1})$$
$$= P(w_1)\prod_{i=2}^{m} P(w_i \mid w_1^{i-1}). \qquad (2)$$

where w_1^m denotes the word sequence with m words.

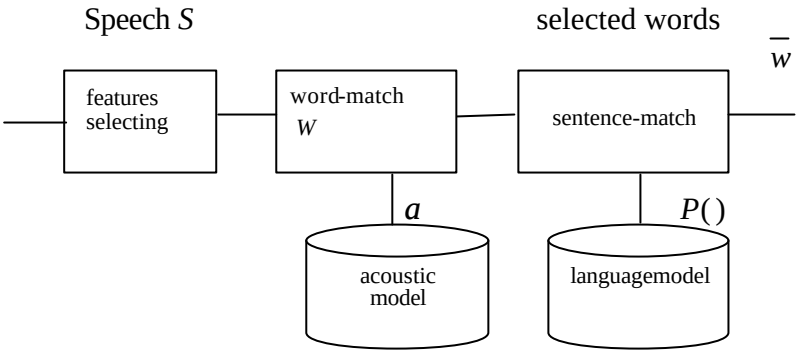

Fig. 1: LMs in speech recognition system.

1.2 *n*-gram Model

Because of the finite training corpora in real world and to reduce the parameter space of word feature in languages, the approximate probability of a given word by using the $(n\text{-}1)^{th}$ preceding words is used to estimate sequence *W*.

The probability model with various n can be written:

$$P(w_1^m) \cong P(w_1)\prod_{i=1}^{m} P(w_i \mid w_{i-n+1}). \qquad (3)$$

where $w_{i\text{-}n+1}$ denotes the history of n-1 word for word w_i.

In many applications, the models for n=1, 2 and 3 are called unigram, bigram and trigram models [1], [8] and [16], respectively.

In Eq. (3), the probability for each event or token can be obtained by training the bigram model (for clarity, bigram model is illustrated). Therefore the probability of a word bigram will be written as:

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}w_i)}{\sum_w C(w_{i-1}w)}, \qquad (4)$$

where $C(w_i)$ is the count of word w_i appeared in training corpus. The probability P of Eq. (4) refers to the relative frequency and such method is called maximum likelihood estimation (MLE).

1.3 Smoothing Issue in Language Models

As shown in Eq. (4), $C(\cdot)$ of a novel word, which don't occur in the corpus, may be zero because of the limited training data and infinite language. It is always hard for us to collect sufficient datum. The potential issue of MLE is that the probability for unseen events is exactly

zero. This is so-called the zero-count problem. It is obvious that zero count will lead to the zero probability of $P(\cdot)$ in Eqs. (3) and (4).

The prediction of zero probability of certain event is unreliable and unfeasible for most applications, especially for language models. The smoothing techniques [3], [4], [11] and [19], are essential and employed by language mode to overcome the issue zero count of traditional language models, as described above.

There are many smoothing methods, such as Add-1, Good-Turing [6], deleted interpolation [7], Katz [13], etc. There are several literatures discussing about smoothing methods [3], [4], [12], [14], [15], [16] and [18].

2. Smoothing Processes in LMs

The adjustment of smoothed probability for all possibly occurred events involves discounting and redistributing processes:

2.1 Discounting Process

The probability of all seen and unseen events is summed to be one (unity). 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.

The adjustment can be divided into two types: static and dynamic. Static smoothing methods, as most smoothing methods, discount the probability based on the frequency of events in trained corpus. However, dynamic smoothing method, i.e., cached-based language, discounts the probability based on the frequency of seen events in cache and trained corpus.

2.2 Redistributing Process

In this operation of smoothing algorithm, the escape probability P_{esc} obtained from all seen events will be redistributed to all unseen events. P_{esc} is usually shared by all the unseen events. That is, P_{esc} is redistributed uniformly to each unseen event, P_{esc}/U, where U is the number of unseen events of a language model. In other hand, each unseen event obtains same probability based on the uniform distribution.

The redistribution process of most well known smoothing methods, such as Add-1, Absolute discounting, Good-Turing, Delete interpolation, Back-off and Witten-Bell, and so on. The escape probability P_{esc} (or called probability mass assigned to all unseen events) is uniformly shared by all unseen events. It is a possible factor that affects the performance of smoothing algorithm. There are little previous papers discussing how to redistribute the escape probability P_{esc}, and how the different redistribution can improve the smoothing methods for language models.

3. Improving the Smoothing Process

3.1 Interval Behavior of Seen Events Count

As described in the section 2, there are two main processes for smoothing methods; discounting and redistributing. In the redistributed process, the escape probability P_{esc} is shared uniformly by all unseen events for most smoothing methods, each event obtain same smoothed probability P_{esc}/U. Based on the observation of behaviors for seen events, each event has its probability relying on the event frequency in the training corpus. It is obvious that the probability distribution

for each event is quite different. Therefore, It is unreasonable to assign same probability to each incoming unseen events.

As shown in Fig. 2, the figures draw the frequency interval (offset) between two new successive events for two models; Chinese character word unigrams and bigrams. There are 100M (10^8) Chinese characters for source training data. The sentences in source are segmented into words and 65M ($65*10^6$) words are obtained. The length of word is 1.45 Chinese characters per word in average.

The recourse files are randomly selected and we obtain the offset diagrams. More than 100 training processes are implemented and then the final curve can be obtained in average. The regression curves Y_1 and Y_2 for Chinese word unigram and character bigram models can be described as follows:

$$Y_1 = 1E^{-10}x^3 - 4E^{-06}x^2 + 0.0307x - 39.825$$
$$Y_2 = -1E^{-16}x^4 + 2E^{-11}x^3 - 6E^{-07}x^2 + 0.0058x - 3.7502$$

where x and y denotes the data size the offset.

An idea for redistributing escape probability P_{esc} is that how many tokens read-in while the next new event will occur? It means the count interval between two successive events, which vary usually with the training data N. Basically, the larger the training data N, the larger, and the interval. In the beginning of training phase, next new events will occur in short interval of count. It means that next new event will occur rapidly at smaller N while slowly at larger N.
The larger the training data N is, the larger the offset (interval) is. It is apparent that the. The regression curves present the general interval of original intervals and its trend increased gradually. Note that the regression curves varied with N and flatter at the beginning and steeply at end of curves.

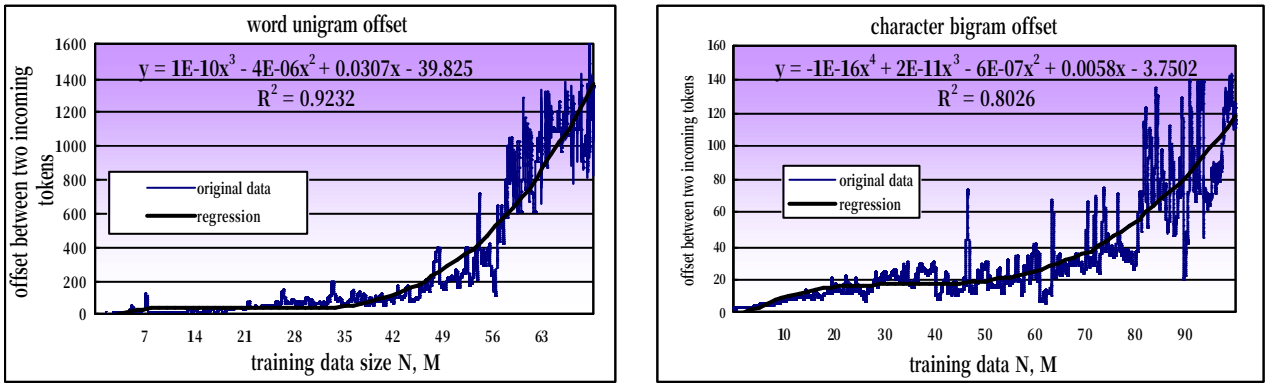

Fig. 2: the interval (offset) between two successive events varied with training data; (left) word unigrams, (right) character bigrams.

3.2 Redistributing Process for Unseen Events

As described above, the regression curves from seen events can be used to demonstrate the interval of unseen events. Based on the curves derived from the seen events occurrence, we can furthermore derive the behaviors for estimating the probability assigning to the next incoming unseen event. Note that all the probability for seen and unseen events should be unity; which must satisfy the basic statistical condition.

Supposed that the interval y_i on training data N_i, the distribution for all unseen events can be as follows:

$$d_i = \frac{\frac{1}{y_i}}{\sum_{j=1}^{U} \frac{1}{y_j}}, \qquad (6)$$

where y_i denotes the interval on location i in Fig. 2 and U denotes the types of unseen events. $1/y_i$ can be regared as the derivatives at y_i and as the probability for unseen events.

The smoothed probability assigning to an unseen event U_i is:

$$p_i = P_{esc} * d_i \ . \qquad (7)$$

Referring to Eqs (6) and (7), the total smoothed probability for all unseen events is P_{esc}. The probability for all seen and unseen events are summed as unity.

4. Evaluation

Our proposed method will be evaluated on the well-known and popular smoothing Good-Turing technique. The cut-off value for word count is usually used to improve the technique. Based on the empirical results, we can obtain best cut-off value on various training data N.

4.1 Basic Idea of Good-Turing Method

Good-Turing method is a well-known and effective smoothing technique, which was first described by I. J. Good and A. M. Turing in 1953 [7] and used to decipher the German Enigma code during World War II. Some previous works are in [4] and [9]. Notation n_c denotes the number of n-grams with exactly c count in the corpus. For example, n_0 represent that the number of n-grams with zero count and n_1 means the number of n-grams which exactly occur once in training data.

The redistributed count c^* for *Good-Turing* smoothing will be presented in term of n_c, n_{c+1} and c as follows:

$$c^* = (c+1)\frac{n_{c+1}}{n_c} \ . \qquad (8)$$

4.2 Best Cut-off Value in Good-Turing Smoothing Method

In the most previous works of smoothing methods, they discussed the situation the possible event types B were much larger than the training data N ($N/B << 1$), such as words trigram models in English text or character trigrams in Mandarin. However, the situation $N/B >> 1$ or $N/B \cong 1$ should be considered in certain applications. For instance, the event types B for Chinese character

bigram is close to $1.69*10^8$ while the training data size N, in general, is usually less than $1*10^8$. In such case, the ratio of N/B is close to 1.

The cut-off value c_o for event count is used to improve the Good-Turing Smoothing, as shown in previous section. The best c_o on various training data N should be analyzed to obtain better improvement, shown in next section

4.3 Data Sets and Empirical Models

In the following experiments, two text sources are collected from the news texts and Academic balanced corpus (ASBC); the former and the later contain 100M and 10M Mandarin characters, respectively. We construct two models to evaluate the cross entropy CE of the technique to improve smoothing process; word unigrams model (word length is 1.45 characters in average) and Chinese character bigrams. The cross entropy is calculated on various data size N in our experiments.

Comparing uniform with non-uniform redistribution probability for unseen events, Fig. 3 and Fig. 4 display the cross entropy (CE), unseen event rates and improvements of different cut-off c_o on various N for word unigram and character bigram models, respectively. The best cut-off c_o can be found on various N for both models. For the word unigrams model, it is apparent that the best CE improvement reaches near 1.8% at N=0.5M, and the effectiveness decreases while the N is larger, as shown in Fig. 3. For bigram model, the best CE improvement reaches near 14.3% at N=1M, and the effectiveness decreases while the N is larger, as shown in Fig. 4.

Both models reach lower CE while the cut-off and non-uniform redistribution technique are exploited. It can improve better, especially on higher unseen event rate.

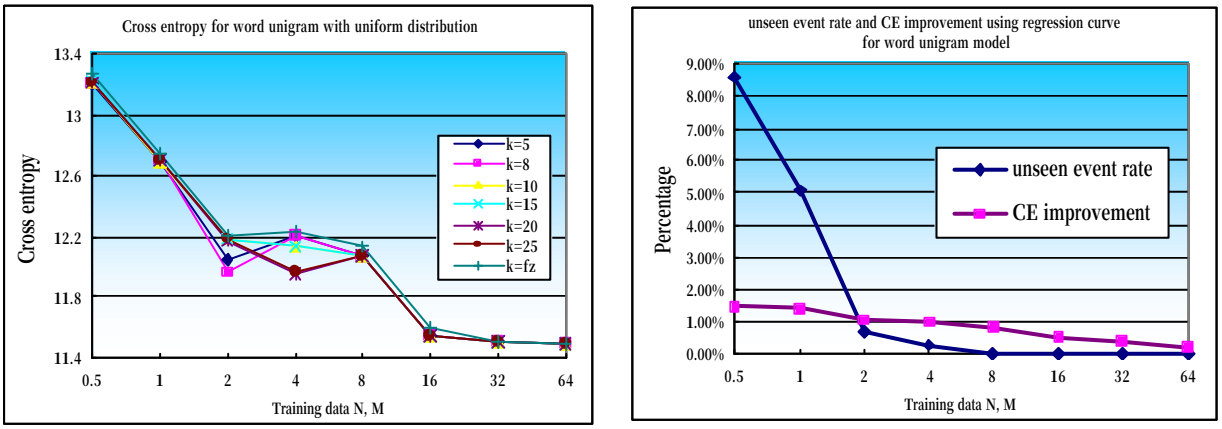

Figure 3: the cross entropy, unseen event rates and improvements on different cut-off c_o on various N for word unigram model.

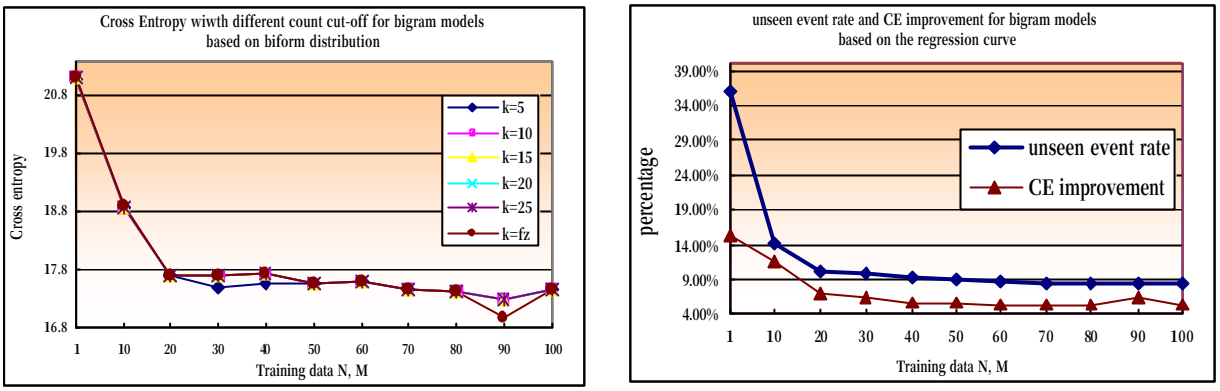

Figure 4: the cross entropy, unseen event rates and improvements on different cut-off c_o on various N for character bigram model.

5. Conclusions

In the paper, we have proposed an effective technique, based on the non-uniform redistribution probability for novel events, to improve the redistribution process in smoothing method of language models. The smoothing method is used to resolve the zero count problems in traditional language models. The cut-off c_o for event count is used to improve the zero n_c issue of Good-Turing Smoothing.

Based on the probabilistic behavior of seen events, the redistribution process exploited by our technique is non-uniform. The improvements discussed in the paper are apparent and effective on Good-Turing smoothing methods.

Empirical results are demonstrated and analyzed for two language models to evaluate the proposed technique methods discussed in the paper; Chinese word unigrams, character bigram model. The cross entropy can be reduced in these two models.

Both models reach lower CE for various cut-off c_o on different training data N and non-uniform redistribution probability are used. Two methods can improve better, especially on higher unseen event rate. In other word, we can improve especially the CE for application with small training data N. The best CE improvement reaches 1.8% and 14.3% for word unigrams and character bigram models.

Reference

[1] Brown P. F., Pietra V. J., deSouza P. V., Lai J. C., and Mercer R. L., 1992, Class-Based n-gram Models of Natural Language, Computational Linguistics, 18, pp. 467-479.

[2] Brown P. F., Pietra V. J., Jphn Cocke, Stphen A. DellaPietra, Vencent J. DellaPietra, Frederick Jelinek, John D. Laffery, Robert L.Mercer, and Paul S. Roossin, 1990, A Statistical Approach to Machine Translation, Computational Linguistics, 16(2), pp. 79-85.

[3] Chen Stanley F. and Goodman Joshua, 1999, An Empirical study of smoothing Techniques for Language Modeling, Computer Speech and Language, Vol. 13, pp. 359-394.

[4] Church K. W., 1988, A Stochastic Parts Program and Noun Phrase parser for Unrestricted Text, Proceedings of the 2nd Conference on Applied natural Language processing, pp. 136-143.
[5] Church K. W. and Gale W. A., 1991, A Comparison of the Enhanced Good-Turing and Deleted Estimation Methods for Estimating Probabilies of English Bigrams, Computer Speech and Language, Vol. 5, pp 19-54.
[6] Essen U. and Steinbiss, 1992, Cooccurrence Smoothing for Stochastic Language Modeling, IEEE International conference on Acoustic, Speech and Signal Processing, Vol. 1, pp. 161-164.
[7] Good I. J., 1953, The Population Frequencies of Species and the Estimation of Population Parameters, Biometrika, Vol. 40, pp. 237-264.
[8] Jelinek F., 1997, Automatic Speech Recognition-Statistical Methods, M.I.T.
[9] Jelinek F. and Mercer R. L., 1980, Interpolated Estimation of Markov Source Parameters from Spars Data, Proceedings of the Workshop on Pattern Recognition in Practice, North-Holland, Amsterdam, The Northlands, pp. 381-397.
[10] Jianfeng Gao,Jian-Yue Nie, Guangyuan Wu, and Guihong Cao, 2004, Dependence Language Model for Information Retriveal, SIGIR '04, Sheffield Yorkshine, UK.
[11] Jurafsky D. and Martin J. H., 2000, Speech and Language Processing, Prentice Hall.
[12] Juang B. H and Lo S. H., 1994, On the Bias if the Good -Turing Estimate of Probabilities, IEEE Trans. on Signal Processing, Vol. 42, No. 2, pp. 496-498.
[13] Katz S. M., March 1987, Estimation of Probabilities from Sparse Data for the Language Models Component of a Speech Recognizer, IEEE Trans. On Acoustic, Speech and Signal Processing, Vol. ASSP-35, pp. 400-401.
[14] Kneser R. and Ney H., 1995, Improved Backing-Off for M-gram Language Modeling, IEEE International conference on Acoustic, Speech and Signal Processing, pp. 181-184.
[15] Nadas A., 1985, On Turing's Formula for Word Probabilities, IEEE Trans. On Acoustic, Speech and Signal Processing, Vol. ASSP-33, pp. 1414-1416.
[16] Ney H. and Essen U., 1991, On Smoothing Techniques for Bigram-Based Natural Language Modeling, IEEE International conference on Acoustic, Speech and Signal Processing, pp. 825-828.
[17] Oren Kurland and Lillian Lee, 2004, Corpus Structure, Language and Ad Hoc Information Retrieval, SIGIR '04, Sheffield Yorkshine, UK.
[18] Standley F. Chen and Ronald Rosenfeld, January 2000, A Survey of Smoothing Techniques, for ME Models, IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 1, pp. 37-50.
[19] Witten L. H. and Bell T. C., 1991, The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression, IEEE Transaction on Information theory, Vol. 37, No. 4, pp. 1085-1094.