Improvements of Smoothing Methods for Language Models

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

Lin Yih-Jeng
Department of Information Mangement
Chien-Kuo Technology University
Changhua, 500 Taiwan
yclin@ctu.edu.tw

Abstract

We study the improvement for the well-known Good-Turing smoothing and a novel idea of probability redistribution for unseen events is proposed. The smoothing method is used to resolve the zero count problem in traditional language models. The cut-off value *co* for number of count is used to improve the Good-Turing Smoothing. The best *k* on various training data N are analyzed.

Basically, there are two processes for smoothing techniques: 1)discounting and 2)redistributing. Instead of uniform assignment of probability used by several well-known methods for each unseen event we propose new concept of improvement for redistribution of smoothing method. Based on the probabilistic behavior of seen events, the redistribution process is non-uniform. The empirical results are demonstrated and analyzed for two improvements. The improvements discussed in the paper are apparent and effective for smoothing methods, especially on higher unseen event rate.

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

1 Introduction

Language models (LMs) [4], [16] have been successively used in many fields of application; speech recognition, word segmentation, information retrieval. In natural language processing (NLP), LMs can be used, for instance, to decide the correct target word sequence *W*. The conditional probability P(*W*), where $W=w_1w_2w\ldots w_m$ is a possible translation of *Str*.

A language model can be regarded the probability distribution over events sequences that models how often each sequence occurs as a sentence. Chain rule of probability is used to decompose the probability calculation:

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

1.1 n-gram Model

Due to the finite training corpora, the approximate probability of a given word by using the $(n\text{-}1)^{th}$ preceding words or tokens is employed to estimate.

The probability model with various *n* can be written as:

$$P(w_1^m) = P(w_1)\prod_{i=1}^{m} P(w_i \mid w_{i-n+1}). \quad (2)$$

In many applications, the models for *n*=1, 2 and 3 are called unigram, bigram and trigram models [1], [6] and [11], respectively.

In Eq. (2), 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:

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

where $C(w_i)$ is the count of word w_i occurred in training corpus. The probability *P* of Eq. (3) is the relative frequency and such a method of parameter estimation is called *maximum likelihood estimation* (MLE).

As shown in Eq. (3), $C(\cdot)$ of event or word (in the paper) may be zero because of the limited training data and infinite resource language. It is hard for us to collect large enough data. 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 zero count will lead to the zero probability of $P(\cdot)$ in Eqs. (2) and (3). Furthermore, it will degrade the performance of LMs.

1.2 Smoothing Methods

The estimation of zero probability of certain event or object is unreliable and unfeasible for most applications, especially for language models. The smoothing techniques [2], [3], [12] and [13], are essential and employed by language mode to overcome the problems of traditional language models, as described above.

There are many smoothing methods, such as Add-1, deleted interpolation [7], Good-Turing [5], Katz [8], etc. Among the smoothing methods, Good-Turing has been used extensively, used to decipher the German Enigma code during War II by I. J. Good and A. M. Turing [5]. There are several literatures discussing about smoothing methods, refrerring to papers of [2], [3], [9], [10], [11] [14] and [15]. We aim at the feature analysis on Good-Turing smoothing methodfor several language models in Mandarin text.

1.3 Processes of Smoothing Methods

The main idea of smoothing 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 unseen events. Smoothing algorithms can be considered as discounting some counts of seen events in order to obtain the escape probability P_{esc} which will be assigned into the zero counts of unseen events. The adjustment of smoothed probability for all possibly occurred events involves discounting and redistributing processes:

A). Discounting:

The probability of all seen and unseen events is summed to be 1 (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. In the process, there are two issues:

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

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.

B). Redistributing:

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 in uniform distribution.

The main issues in this process should be considered:

1) How to redistribute the escape probability P_{esc} to each unseen event?
2) Whether the incoming and all other unseen events be assigned same probability or not?

The redistribution of most well known smoothing methods, such as *Add-one*, *Absolute discounting*, *Good-Turing*, *Delete interpolation*, *Back-off* and *Witten-Bell*. The escape probability P_{esc} (or called probability mass for unseen events) is shared uniformly 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.

2 Improvement for Good-Turing Smoothing

2.1 Basic Idea of Good-Turing Smoothing

Good-Turing method was first described by I. J. Good and A. M. Turing in 1953 [6], which was used to decipher the German Enigma code during World War II. Some previous works are in [3], [7] and [12]. 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. Therefore, n_c will be described as:

$$n_c = \sum_{w:C(w)=c} 1 , \quad (4)$$

where *w* denotes a bigram in training corpus. Based on *Good-Turing* smoothing, the redistributed count c^* will be presented in term of n_c, n_{c+1} and *c* as:

$$c^* = (c+1)\frac{n_{c+1}}{n_c} \quad (5)$$

Note that the numerator in Eq. (3) will be substituted by c^*

for the bigram models as:

$$P(w_{i-1}w_i) = \frac{c^*}{\sum_w C(w_{i-1}w)} \quad (6)$$

The probability of Eq. (6) is called Good-Turing estimator. Similarly, the revised count for bigrams can be derived from Eq. (5). As shown in Eq. (6), *Good-Turing* smoothing method just employs the bigram models to smooth the probability, rather than interpolating higher and lower order models (such as unigrams). Hence, *Good-Turing* is usually a tool used by other smoothing methods.

2.2 Issue of Good-Turing Smoothing

The idea of Good-Turing is described by Eq. (5). In training cprpus, the larger *c*, the smaller n_c. Basically, count *c* increases while n_c decreases. In some counts, however, it is not always true, such as *c*=765~769. It is apparent that zero n_c will leads to the zero Good-Turing estimator. Redistributed count c^* is zero while n_c is zero. Consequently, probability *P* will be zero.

Supposed that the incoming bigram b_{N+1} are the unseen bigram and seen bigram with count 2 on left and right column, respectively. Shown in Table 1, n_{50}=68, n_{50}=59 while n_{50}=64. The number n_0 of unseen bigrams is decreased gradually while training data size *N* is increased. The number in grey cells are changed while the .

Table 1 :when an event (bigram) occurs, the changes for some n_c, n_{c+1} and *c* on *N*=1M and 1M+1 Mandarin characters text.

count *c*	n_c	count *c*	n_c
0	168158426	765	2
1	357056	**766**	**0**
2	134394	767	9
3	68092	768	3
4	43983	**769**	**0**

3 Improvements for Smoothing Method

3.1 Improvement: Cut-off Count of Events

We can prove that Good-Turing estimator satisfy the total trainig data *N* derived by count c^* as :

$$\sum_i c_i n_i = c_0 n_0 + c_1 n_1 + c_2 n_2 + c_3 n_3 + ... = N, \text{ for all } i >= 0.$$

The events with higher count are reliable and we need therefore to make no adjustment. In real applications, only the lower redistributed count c^* are used to aviod the zero n_c, which will lead zero probability also. Thus, the cut-off count c_k is needed. As shown in [8], Katz suggest k for English will get better performance.The number of unseen events varies with the training data and affects the behavior and performance of smoothing techniques. The discounting c^* is as follow:

$$c^* = \begin{cases} (1+0)\frac{n_1}{n_0} = \frac{n_1}{U} & \text{for } c = 0, \\ \dfrac{(c+1)\frac{n_{c+1}}{n_c} - c\frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}, & \text{for } 1 \le c \le k, \\ c & \text{for } c > k, \end{cases} \quad (7)$$

where *k* denotes the cut-off value.

3.2 Best Cut-off Value

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 triigram models in English text or character triigrams 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 *co* for event count is used to improve the Good-Turing Smoothing, as shown in previous section. The best *co* on various Training data N should be analyzed to obtain better improvement.

3.3 The Redistributing Process

As described in Section 1.3, there are two main processes for smoothing methods; discounting and redistributing. Within the redistributed process, the escape probability P_{esc} (or so-called probability mass for unseen events) is shared uniformly by all unseen events for most

smoothing methods, such as *Add-one*, *Delete interpolation* and *Witten-Bells* method *A* and *C*. In other words, 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 distrubition for each event is quite different. Therefore, It is unreasonable to assign same probability to each incoming unseen events.

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 interval between two successive events varied with the training data *N*. Basically, the larger the training data *N*, the smaller the interval. It means that next new event will occur rapidly at smaller *N* while slowly at larger *N*.

As shown in Fig. 1, the figures draw the 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 characters per word in average.

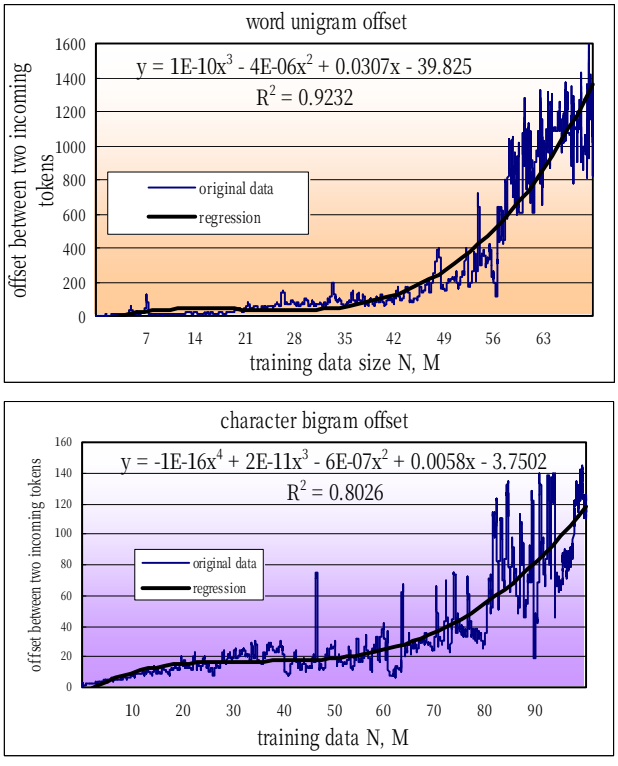


Fig. 1: the interval (offset) between two successive events varied with training data; (top) word unigrams, (bottom) character bigrams.

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:

$Y_1 = 1E\text{-}10x^3 - 4E\text{-}06x^2 + 0.0307x - 39.825$

$Y_2 = -1E\text{-}16x^4 + 2E\text{-}11x^3 - 6E\text{-}07x^2 + 0.0058x - 3.7502$

where x and y denotes the data size the offset.

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.

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 (1); 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:

$$d_i = \frac{\frac{1}{y_i}}{\sum_{j=1}^{U} \frac{1}{y_j}}, \qquad (8)$$

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:

$$p_i = P_{esc} * d_i \qquad (9)$$

Referring to Eqs (8) and (9), the total smoothed probability for all unseen events is P_{esc}. Therefore, the total probability for seen and unseen events can be calculated as unity.

4 Evaluation

4.1 Cross Entropy

In the subsection, we introduce *cross entropy*, which has always been used to evaluate and compare different probabilistic model. For a testing data set *T* which

contains a set of events, $e_1,e_2,\ldots,e_m$, the probability for the testing set $P(T)$ can be described as:

$$P(T)=\prod_{i=1}^{m}P(e_i), \qquad (10)$$

where m denotes the number of events in testing set T and $P(e_i)$ denotes the probability of event e_i, obtained from n-gram language model, assigning to event e_i.

When we don't know the actual probability distribution p that generated some data the cross entropy CH can be employed. For example, we use some M, which is a model of p. Therefore, the cross entropy CH of M on p can be regarded as:

$$CH(p,M)=\lim_{n\to\infty}\frac{1}{n}\sum_{W\in L}p(w_1w_2w_3...w_n)\log M(w_1w_2w_3...w_n)\,. \qquad (11)$$

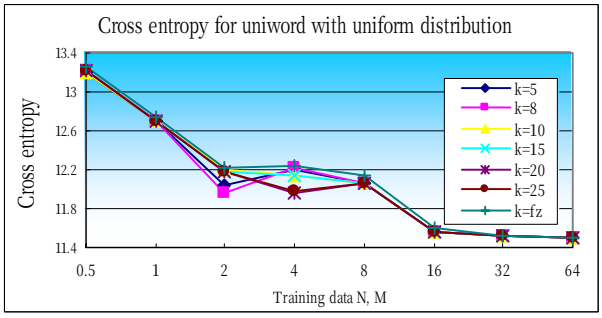
4.2 Data Sets and Empirical Models

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 three models to evaluate the Cross entropy CE of smoothing methods discussed in the paper; Chinese character unigrams, bigrams and word unigrams model (word length is 1.45 characters in average). The entropy is calculated on various data size N in our experiments. The number n_c of first zero for event with count 0 on various N are shown in Table 2.

Table 2: The first zero fz of n_c in term of training size N for two models.

N, M	10	20	30	40	50	60	70	80	90	100
char. Bigram	606	698	1014	520	310	446	249	277	283	389
N, M	0.5	1	2	4	8	16	32	64		
Word unigram	177	230	287	418	455	620	778	817		

Based on the non-uniform distribution probability for unseen events, Fig. 2 and Fig. 3 display the cross entropy (CE), unseen event rates and improvements of different cut-off co on various N for word unigram and character bigram models respectively. The best cut-off co can be found on various N for both models. 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. 2. The best CE improvement reaches near 14.3% at N=1M, and the effectiveness decreases while the N is larger, as shown in Fig. 3. Both improvements can reach lower CE while the cut-off and non-distribution methods are used. Two methods can improve better, especially on higher unseen event rate.



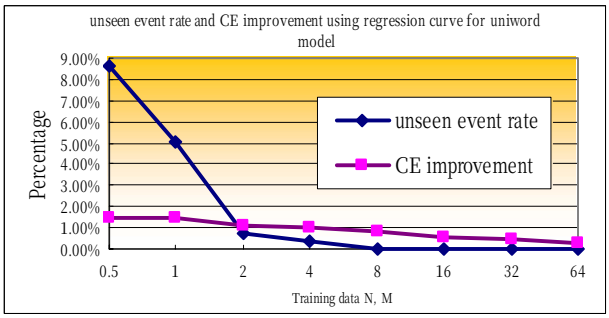
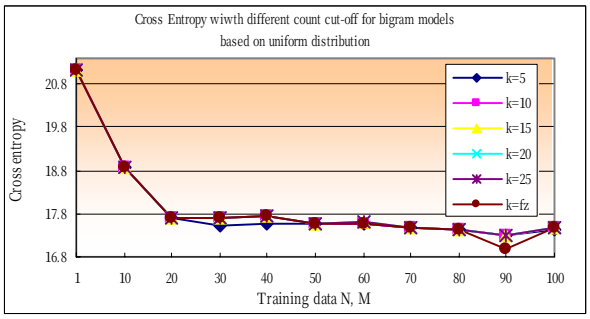


Figure 2: the cross entropy, unseen event rates and improvements on different cut-off co on various N for word unigram model.



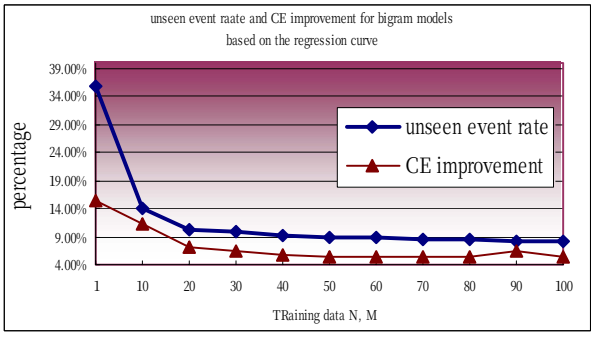


Figure 3: the cross entropy, unseen event rates and improvements on different cut-off co on various N for character bigram model.

5 Conclusions

In the paper, we study an improvement for the well-known Good-Turing smoothing and propose a novel idea of probability redistribution for unseen events. The smoothing method is used to resolve the zero count problem in traditional language models. The cut-off *co* for event count is used to improve the zero n_c issue of Good-Turing Smoothing. The best *co* on various training data *N* is analyzed.

Based on the probabilistic behavior of seen events, the redistribution process is non-uniform. The empirical results are demonstrated and analyzed for two improvements. The improvements discussed in the paper are apparent and effective for smoothing methods.

We construct two models to evaluate the improvement methods discussed in the paper; Chinese word unigrams, character bigram model. The cross entropy can be reduced in these two models. Both improvements can reach lower CE while various cut-off co on different *N* and non-distribution methods 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 unigram 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, Vol. 18, pp. 467-479.

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

[3] 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.

[4] Essen U. and Steinbiss, 1992, Cooccurrence Smoothing for Stochastic Language Modelling, IEEE International conference on Acoustic, Speech and Signal Processing, Vol. 1, pp. 161-164.

[5] Good I. J., 1953, The Population Frequencies of Species and the Estimation of Population Parameters, Biometrika, Vol. 40, pp. 237-264.

[6] Jelinek F., 1997, Automatic Speech Recognition-Statistical Methods, M.I.T.

[7] 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.

[8] 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.

[9] 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.

[10] Nadas A., 1985, On Turing's Formula for Word Probabilities, IEEE Trans. On Acoustic, Speech and Signal Processing, Vol. ASSP-33, pp. 1414-1416.

[11] 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.

[12] 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.

[13] Jurafsky D. and Martin J. H., 2000, Speech and Language Processing, Prentice Hall.

[14] Juang B. H and Lo S. H., 1994, On the Bias if the Turing-Good Estimate of Probabilities, IEEE Trans. on Signal Processing, Vol. 42, No. 2, pp. 496-498.

[15] 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.

[16] Jianfeng Gao,Jian-Yue Nie, Guangyuan Wu, and Guihong Cao, 2004, Dependence Language Model for Information Retriveal, SIGIR '04, Sheffield Yorkshine, UK.