Survey of the Smoothing Issues on Mandarin Language Models

[1]HuangFeng-Long, [2]Yu Ming-Shing and [1]Chiang Yang-Kua
(黃豐隆，余明興，江緣貴)
[1] Department of Computer Information Engineering
National United University, MiaoLi, 360, Taiwan, R. O. C.
[2]Department of Computer Science
National Chung-Hsing University, Taichung 40227, Taiwan, R. O. C.
[1]{flhuang,ykchiang}@nuu.edu.tw, [2]msyu@nchu.edu.tw

ABSTRACT

We survey several frequent smoothing methods used by language models for Mandarin. Due to the problem of data sparseness, smoothing techniques are employed to re-estimate the probability for all events while calculating the probability of occurrence. Among well-known smoothing methods, G*ood-Turing* is employed widely. We have proposed a set of properties to analyze the behaviors of *Good-Turing* in this paper. Two novel smoothing methods are proposed. Finally, we implement three *n*-gram for Mandarin and then analyze the entropy and related problems of the G*ood-Turing*; such as cut-off value and types of events.

Keywords: Language models, smoothing methods, statistical behavior, entropy.

1、Introduction

In natural language processing (NLP), language models have been employed in many domains: information retrieval [Pronte, Croft, 1998; Djoerd, 2002], Part-of-speech (POS) taggers [Chen et al., 1994], speech recognizers [Katz, March 1987, Jelinek, 1997], and so on. For instance, Language models (LMs) are used 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*, can be represented as:

$$P(w_1w_2w_3,\ldots w_m). \quad (1)$$

The chain rule of probability is used to decompose this probability as:

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

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

In general, the word sequence $W_{\max}$ with maximum conditional probability $P(W)$ in *n*-gram model will be expressed as:

$$P(w_1^m)=\prod_{i=1}^{m}P(w_i). \quad (3)$$

$$W_{\max}=\arg\max_w P(w_1^m)=\arg\max_w\prod_{i=1}^{m}P(w_i\mid w_{i-n+1}^{i-1}) \quad (4)$$

As shown in Eq. (4), the probability for each event can be obtained by training the bigram model. A generalization of LMs is repressed in Eq. (5), called the *n*-gram LMs. In physical applications of NLP, *n* can be 3, 4, even up to 5.

$$P(w_i\mid w_{i-n+1})=\frac{C(w_{i-n+1}^{i-1}w_i)}{\sum_w C(w_{i-n+1}^{i-1}w)}. \quad (5)$$

where $C(\cdot)$ denotes the count of word w_i in training corpus. The probability *P* of Eq. (4) is the relative frequency and such a method of parameter estimation is called *maximum likelihood estimation* (*MLE*).

1.1 Smoothing Issue of LMs

Basically, traditional language models rely on the estimation of the *2*-gram or *3*-gram models; which calculate the probability of each event by using its frequency in training corpora. Even though we do best to collect training corpora, it is on finite size. The new event (like new words or unknown words) will occur in future.

This situation leads to the zero frequency of event and furthermore zero probability while calculate the probability of word sequence *W*. For instance, if bigram $w_{i-1}w_i$ (for bigram models) never occur in the training corpus, then $C(w_{i-1}w_i)$ is equal to 0. It leads to zero of Eq. (2).

1.2 Why Smoothing Methods must be Considered

It is unreasonable to assign probability 0 to the unseen events which don't occur in training data. If we should assign certain probability to such novel events, how is the probability assigned effectively? The schemes used to resolve the problems are called *smoothing* methods. The probability obtained from MLE will be adjusted and redistributed. Such a process will maintain the total probability to be unity. Usually, the smoothing methods can improve the performance language models.

It is almost impossible that for us to collect all possible permutations of events in natural language. The issue of zero probability should be accounted for. Therefore, smoothing methods must be considered in detail to generate the

robust and effect language model.

In fact, there are many works to be alternative methods to alleviate or improve the zero probability of LMs; such as extend language models to X-gram [Knerser, 1996; Chebra et al., 1997; Fong and Wu, 1995], the situation of smoothing should be considered in detail on *n*-gram modeling approach.

1.3 Questions of Smoothing Methods

A basic question is how the smoothing methods affect the performance of LMs. Individual smoothing uses various features.

One key point is that smoothing can avoiding assign zero to all the novel events. From the statistical point, the summation of total probability assigned to all possible events is equal to unity. Other question is that how much probability mass should be re-allocated to all novel events.

1.4 Characteristics of Chinese

The smoothing approaches are used principally by LMs on other languages. Chinese has some special attributes and challenges. First, there is no standard definition of a word, and there are no spaces between characters (字). A Chinese word (詞) is composed of one to several characters(字). The combination of one to several such characters gives an almost unlimited number of words, in which some of them are frequently used and can be found in Chinese dictionaries. Second, linguistic data resources are not yet plentiful, so the best source of training data may be the web.

Chinese word (詞) is the elementary unit which has specific meanings. Because of the absence of word delimiters (like white space) in a sentence, it is necessary to segment a sentence into one more than words. During the processing of Chinese language, each sentence is segmented into one to several words for further processing.

2. Overview of Smoothing

The principal purpose of smoothing is to alleviate the zero count problems, as described above. In this section we will describe the formalisms of smoothing methods in detail. There are several well-known smoothing methods in various applications: *Additive discounting*, *Good-Turing*, *Witten-Bell* and *Katz*. In this section, we propose two novel methods.

In all the following sections, we assume that the domain of smoothed probabilities and related properties analysis of language models is limited only on bigram models for clarity. It is evident that these smoothing methods can be easily expanded into higher order *n*-grams models, for Mandarin or other languages.

2.1 Additive Discount Method

Additive smoothing method is intuitively simple. A small amount δ is added into all *n*-grams (including all seen and unseen *n*-grams). Clearly, the count of each type of bigrams is increased by 1. According to the previous experiments [Chen and Goodman, 1999], the performance was usually degraded by using *add-one* smoothing.

2.2 Good-Turing Method

Good-Turing is first described by Good in 1953 [Good, 1953]. Some previous works are [Chen and Goodman, 1999; Jelinek, 1997; Nadas, 1985]. Notation n_c denotes the number of *n*-grams with exactly *c* count in training 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.

Similarly, the recounted count c^* for *n*-grams can be derived. *Good-Turing* smoothing just employs the *n*-gram models to smooth the probability, rather than interpolating higher and lower order models (such as *n*-1 grams). Hence, *Good-Turing* is usually used as a tool by other smoothing methods.

2.3 Witten-Bell Method

Here we discuss only one of five smoothing schemes: methods ***A*** (called *W-B A*), introduced by *Wetten-Bell*[1] [Ney and Essen, 1991]. Other four work can be referred in [Chen and Goodman, 1999].

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 $1/(N+1)$.

2.4 Katz Method

Katz first proposed the smoothing method in 1987 [Katz, 1987]. Previous works are of [Chen and Goodman, 1999, Juraskey and Martin, 2000]. The basic concept is that *n*-grams can be computed by using the count of *n*-gram and lower order count, such as up to unigram models. If the count of bigrams is 0, then use count of unigram.

Note that the total smoothed probability should be summed up to 1. The smoothing criterion is the well-known *BackOff*; *Katz* is the most typical method.

2.5 Our Smoothing Methods

Several well-known smoothing schemes for estimating probability of bigrams have been explained in this section, such as *Good-Turing*, *Katz*, etc. We have proposed five properties (in Appendix A), which are employed to analyze the statistical behaviors of these smoothing methods. None of four methods comply with all these

[1] There are 5 methods in [Ney and Essen, 1991]; method *A*, *B*, *C*, *P* and *X*. We only discuss one of them in this paper.

properties. In other words, from statistical point, they have some weaknesses or drawbacks while employed in language models.

We propose two novel smoothing methods (***Method A*** and ***B,*** hereafter) and then analyze the statistical behaviors of these five methods.

Method A:

In case for a bigram, *Method A* calculates the smoothed probabilities as:

$$Q(w_{i-1}w_i)=\begin{cases}\dfrac{d_A}{U(N+1)} & \text{for } c(w_{i-1}^i)=0,\\ \dfrac{c(w_{i-1}^i)}{N}\dfrac{N+1-d_A}{N+1} & \text{for } c(w_{i-1}^i)\geq 1,\end{cases} \quad (6)$$

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. (6), $(N+1-d_A)/(N+1)$ 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 $d_A/(N+1)$,

$$\sum_{i:c_i=0} P_i^* = \frac{d_A}{N+1} \qquad \text{for } c_i=0 \quad (7)$$

Obviously, Eq. (7) of Method *A* is similar to *W-B A*. Instead of the constant 1 of numerator in *W-B A*, 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 value of d_A for lower perplexity can be found.

Method B

Method 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 smoothed probabilities will be calculated as follows:

$$P(w_{i-1}w_i)=\begin{cases}\dfrac{d_B}{(N+1)} & \text{for } c(w_{i-1}^i)=0,\\ \dfrac{c(w_{i-1}^i)}{N}\dfrac{N+1-Ud_B}{N+1} & \text{for } c(w_{i-1}^i)\geq 1,\end{cases} \quad (8)$$

and

$$d_B < \min\{\frac{N}{N+2U}, \frac{N+2}{2U}\}. \quad (9)$$

When computing 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. (9), $(N+1-Ud_B)/(N+1)$ is the normalization factor of Q^* for seen bigrams. The probabilities Q will be discounted by the normalization factor and then the remained Q^* are redistributed to unseen bigrams; which share uniformly the distributed probability mass $Ud_B/(N+1)$:

$$\sum_{i:c_i=0} P_i^* = \frac{Ud_B}{(N+1)} \qquad \text{for } c_i=0 \quad (10)$$

3. Properties Analysis of Good-Turing Smoothing

3.1 Why the properties are needed?

As shown in Appendix A, we have proposed five properties which can be used to analyze the statistical behaviors of smoothing. Basically, the estimation of probability of event, supposed for bigram models, is calculated based on several variables; type U of unseen bigrams, type S of seen bigrams in a corpus,the size N of training data,: size V of vocabulary, type B of all the n-grams ($B=V^2$ and $B=U+S$), probability mass P_{mass} re-distributed to all unseen bigrams by a smoothing method.

It is obvious that these variables are varied and will affect each other. The statistical behaviors of smoothing should be analyzed using these properties and to observe specially it on various training data N. The Statistical behavior will affect smoothing methods and furthermore lead to the performance of LMs.

3.2 Analysis of Good-Truing

In previous section, several well-known smoothing methods are introduced. Due to the limit of size, we just analyze the statistical properties of the widely-used Good-Truing smoothing. Smoothed probability for bigrams computed from various smoothing methods should still comply with these properties.

Total number of smoothed count c^* can be

$$\sum_i c_i n_i = c_0 n_0 + c_1 n_1 + c_2 n_2 + c_3 n_3 + \ldots = N, \text{ for all } i>=0. \quad (11)$$

It is apparent that property 1, 2 and 3 does not hold. For instance, look at following case: when n_m is equal 0, $c_m^* = (m+1)\frac{n_{m+1}}{n_m} = \infty$ (violating property 1 and 2) and $Q_{m-2,N}^* > P_{m-1,N}^*$ and $Q_{m,N}^* > Q_{m+1,N}^*$ (violating property 1). The results also violate property 3.

It is possible that one of n_m for certain amount of training set is zero. The smoothed probability for unseen and seen bigrams with c counts, property 4 does not hold.

When a new bigram b_{next} is read in, then training size is increased by one ($N=N+1$). The smoothed count $Q_{c,N}^* > Q_{c,N+1}^*$. Supposed that the bigram b_{next} is ever seen with c counts on training size N, upon the b_{next} appears, $N=N+1$, $n_c=n_c-1$ and $n_{c+1}=n_{c+1}+1$, the smoothed probability for bigrams with c on training size N and $N+1$ can be computed as:

$$Q^*_{c,N} = (c+1)\frac{n_{c+1}}{n_c}/N$$

and

$$Q^*_{c,N+1} = (c+1)\frac{n_{c+1}+1}{n_c-1}/(N+1)$$

$$\frac{Q^*_{c,N}}{Q^*_{c,N+1}} = \frac{(N+1)\frac{n_{c+1}}{n_c}}{N\frac{n_{c+1}+1}{n_c-1}} = \frac{(N+1)(n_c-1)n_{c+1}}{Nn_c(n_{c+1}+1)} \qquad (12)$$

According to inequality Eq. (A7), $Q^*_{c,N} > Q^*_{c,N+1}$. Eq. (12) should be greater than 1. In fact, $N >> n_c$ and $N >> n_{c+1}$. Eq. (12) may be < 1 on certain situation, while it is also possibly greater than 1. Hence, property 5 does not hold.

For the bigram b_{next}, what is the relationship between the smoothed probabilities P^* on training size N and N+1?

$P^*_{c,N} = (c+1)\frac{n_{c+1}}{n_c}/N$, $P^*_{c+1,N+1} = (c+2)\frac{n_{c+2}}{n_{c+1}+1}/(N+1)$, then:

$$\frac{P^*_{c,N}}{P^*_{c+1,N+1}} = \frac{(c+1)(N+1)n_{c+1}(n_{c+1}+1)}{(c+2)Nn_c n_{c+2}}. \qquad (13)$$

According to Eq. (A8), Eq. (13) should be less than 1. It is obvious that Eq. (13) may be greater than 1 in certain situations, while it is possibly less than 1. Therefore, property 5 does not hold.

4. The Evaluations

In this section, first the empirical data sets and three Mandarin models are evaluated. The probability mass P_{mass} assigned to unseen events by various smoothing methods are analyzed. The entropy of all smoothing method discussed for three models are shown. We further discuss the relationship between the P_{mass} and entropy which is a metric for evaluating a LM.

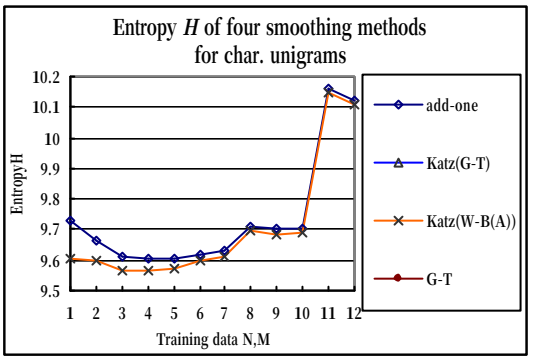
Whether the volume of P_{mass} affecting the entropy H of LM or not will be shown. Finally, we further discuss some special problems in *Good-Turing* for Mandarin models. We will suggest the best cut-off k_b in term of training size N for Mandarin corpus.

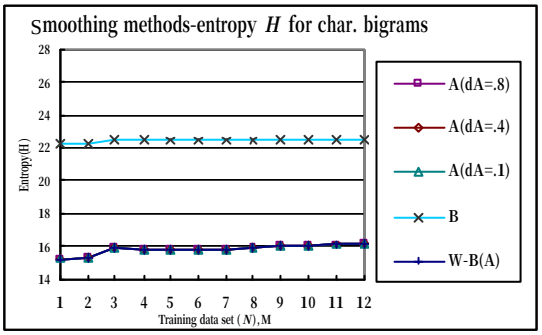
4.1 Data Sets and Models

In the following experiments, two text sources are used as data sets; the news texts collected from Internet and ASBC corpus [Huang, 1995]. The HTML tags and all unnecessary symbols are extracted and there are about 7M Mandarin characters in news texts. The Academic Sinica Balanced Corpus version 3.0 (ASBC) includes 316 text files distributed in different fields, occupying 118MB memory and about 5.22 millions of words labeled with a POS tag. Our corpus contains totally up to 12M Mandarin characters.

In our experiments, we construct three models to evaluate the entropy of smoothing methods discussed in the paper; Mandarin character unigrams, character bigrams and word unigrams model. The entropy of each method is calculated on various data size N in our experiments, from 1 M to 12M Mandarin characters. The first two models employ up to 12M Mandarin characters (unigrams and bigrams) and the 3rd model use about up to 5M Mandarin words in ASBC corpus. Table 1 displays the entropy of smoothing methods on various size N for unigram character Mandarin model; the entropy of Good-Truing and Katz is similar. On the bottom of Table 1, entropies of Method A, B and Witten-Bell are shown; Method B is higher than others on all various N and Method A is similar to Witten-Bell.

Table 1. (top)the entropy of four methods on various size N for unigram character Mandarin model.
(bottom) entropy of four methods on various size N for bigram character Mandarin model.





4.2 Probability Mass Assigned to Unseen Events

Table 2 shows the probability mass P_{mass} to be redistributed to all unseen bigrams and normalized factor for each seen bigram. P_{mass} is varied primarily with the smoothing methods and then with respect to the parameters N, U, S and some constants. It is obvious that the normalization factor (NF) will affect the probability discounted from the probability P assigned to the events with c counts (c >= 1) prior to smoothing process. Note that *Katz* isn't shown in Table 1 because the method adopts the *Backoff* scheme to calculate the probability.

Table 2: The probability mass P_{mass} and normalization factor (NF).

perperty / method	probability mass for all novels	*N. F.* for all seen events
Additive	$U/(N+U)$	$N/(N+B)$
Good-Turing	$\frac{n_1}{N}$	$\frac{(m+1)n_{m+1}}{mn_m}$
Witten-Bell (A)	$1/(N+1)$	$N/(N+1)$
Witten-Bell (C)	$S/(N+S)$	$N/(N+S)$
Katz[2]	$\frac{n_1}{N}$	$\frac{(m+1)n_{m+1}}{mn_m}$
Absolute	DT/N	[3]
Method A	$d_A/(N+1)$	$(N+1-d_A)/(N+1)$
Method B	$Ud_B/(N+1)$	$(N+1-Ud_B)/(N+1)$

4.3 Cut-off *k* for Recount of Events

Good-Turing has been employed in many natural language applications. Previous works [Chen and Goodman, 1999] and [Nadas, 1985] discussed the related parameters, such as cut-off k in *Good-Turing* method. However, these works employ English corpora only. In this section, we will focus on the *Good-Turing* method in Mandarin corpus and further analyze the problems of *Good-Turing* for Mandarin texts, such as cut-off k and discounted counts for seen and unseen bigrams.

Good-Turing re-estimate the count c^* of all events in term of original count c and event number n_c and n_{c+1}. In practice, the discounted count c^* is not used for all count c. It is assumed that larger counts are much reliable. The recount c^* are set by *Katz* [Katz, 1987] as:

$$c^* = \begin{cases} (1+0)\frac{n_1}{n_0} = \frac{n_1}{U} & \text{for } c = 0, \\ \dfrac{(c+1)\frac{n_{c+1}}{n_c} - c\frac{(k+1)n_{k+1}}{n_1}}{1-\frac{(k+1)n_{k+1}}{n_1}}, & \text{for } 1 \le c \le k, \\ c & \text{for } c > k, \end{cases} \quad (14)$$

where:

c denotes the count of an event,

c^* denotes the recount of an event, suggested by Katz ,1987 for English data.

n_i denotes the number of bigrms with i counts.

k denotes the cut-off value.

Good-Turing was first applied as a smoothing method for n-gram models by *Katz* [Katz, 1987]. Until now, few papers discuss the related problems between cut-off k and entropy for Mandarin corpus, even for English. *Katz* in [Katz, 1987] suggested a cut-off k at 5 as threshold for English corpus. Another important parameter of *Good-Turing* is the best k_b (not ever discussed in previous works) in term of training size N, which will lead to minimum entropy at different N. We will analyze these related problems for Mandarin in this section.

4.4 Type Number n_c of Events with c

For Mandarin character unigram model, we first calculate the recount c^* (c>=0). Referring to the empirical results, some recounts c^* are negative (<0). In such case, furthermore it leads to negative probability P) and violate the statistical principle. For instance, c=8, n_8=106, n_9=67, k=10, recount c^* can be calculated and is negative (-20.56). This situation also happens to some other recounts in character unigram model. Therefore, we must exclude the problem when *Good-Turing* is employed as smoothing method for Mandarin character unigram model.

One way to solve the problem of negative recount is that we must define a cut-off k carefully. When we decide the cut-off k, only the count c less than k will be re-calculated while the counts $c >= k$ is not calculated. In other words, we should choose a suitable k based on the empirical observation to avoid the negative situation. For the example above, we may choose k=8. Therefore, all the counts c>=8 need not be re-calculated. For the Mandarin character bigrams and words unigram models, based on the results, the negative recount situation doesn't happen through all the training size N of two corpora.

The type number n_c of counts c and recount c^* of English words and Mandarin characters (12M) bigram model are listed in Table 3. Church and Gale [Church and Gale, 1991] used the 22M English corpus from Associated Press (AP) to calculate the recount of character bigrams. 4.49E+5 bigrams have a count c of 2 and recount c^* of 1.26.

Table 3: The events number n_c and recount c^* by *Good-Turing* discounting for English word bigrams and Mandarin character bigram.

models / count c	English[4] bigrams (22M)		Mandarin char. bigrams (12M)	
	n_c	c^*	n_c	c^*
0	7.46E+10	2.70E-5	1.69E+8	2.11E-3
1	2.01E+6	4.46E-1	3.57E+5	7.50E-1
2	4.49E+5	1.26	1.34E+5	1.51
3	1.88E+5	2.24	6.81E+4	2.58
4	1.05E+5	3.24	4.39E+4	3.54
5	6.83E+4	4.22	3.12E+4	4.47
6	4.81E+4	5.19	2.33E+4	5.51

[2] In the paper, we employ the *Good-Turing* method to discount the count of each event.

[3] Interpolating high order bigram with lower order unigrams.

[4] 22 million ($2.2*10^{+7}$) words bigrams from Associated Press (AP).

5. Conclusion

We survey several smoothing methods used by Mandarin language models. Evaluations based on entropy are implemented. Good-Turing is analyzed on the parameters, such as mass redistributed to unseen event, cutoff k and type number of event, observing the statistical behavior of smoothing methods.

Reference

Chebra C., Eagle D., Jelinek F. Jimenez V. Khudanpr S., Mangu L., Printz H. Ristad E., Rosenfeld R., wu D., 1997, Structure and Performance of a Dependency Language Model, In Eurospeech '97, Rhodes, Greece.

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.

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.

Djoerd Hiemstra, 2002 August, Term-specific Smoothing for the Language Modeling Approach to Informational Retrieval, In SIGIR'02.

Fong E. and Wu D., 1995, Learning Restricted Probabilistic Link Grammars, IJCAI workshop on New Approaches to Learning for Natural Language Processing, Montreal.

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

Huang C.-R., 1995, Introduction to the Academic Sinica Balance Corpus, Proceeding of ROCLLING VII, pp. 81-99.

Jelinnek, F., 1997, Statistical Methods for speech Recognizers, MIT press.

Juraskey D. and Martin James H., 2000, Speech and Language Processing, Prentice Hall.

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.

Knerser Reinhard, 1996, Statistical Language Modeling Using a Variable Context Length, International Conference of spoken Language and Processing (ICSLP-96), pp. 494-497.

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

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.

Pronte, J. M., Croft W. B., 1998, A Language Modeling Approach to Informational Retrieval, in 21st ACM conference on research and department in Informational Retrieval, pp. 275-281.

Appendix A. The Proposed Properties

We have proposed five properties which can be used to analyze statistical behaviors of language models. The properties are analyzed briefly on bigram model.

Property 1

The smoothed probability for any one bigram b_i with i counts should falls between 0 and 1 (0,1), which is described as follows:

$$0 < P^*_{i,N} < 1 \quad \text{For all bigram on various } N \qquad \text{(A1)}$$

Property 2

The summation of smoothed probability P^* for all the bigrams is necessarily equal to 1 on any training size N. Total probability P is summed as:

$$P^*_{1,N} + P^*_{2,N} + \ldots + P^*_{B,N} = \sum_{b_i \in seen\ bigrams} P^*_{i,N} + \sum_{b_j \in unseen\ bigrams} P^*_{j,N} = 1, \qquad \text{(A2)}$$

where B denotes the total number of bigrams.

Property 3

The smoothed probability assigned to the bigrams b with different count should satisfy all the following inequality equations[5]:

$$Q^*_{c,N} < Q^*_{c+1,N}, \quad \text{for } c=0,1,2,\ldots, \qquad \text{(A3)}$$

Smoothed probability for any bigram b_i and b_j with same count ($b_i \neq b_j$, $i=j$) should be same on any training size N. Instead, the probability for bigram b_{i+1} with c+1 counts should be larger than that of bigrams with c counts.

Property 4

Comparing to the probability P prior to smoothing process, the smoothed probability P^* for all bigrams will be changed. Property 4 can be expressed as follows:

$$Q^*_{0,N} > Q_{0,N}, \quad \text{for } c=0 \quad \text{//for all unseen bigrams} \qquad \text{(A4)}$$

$$Q^*_{c,N} < Q_{c,N}, \quad \text{for } c \geq 1 \quad \text{//for all seen bigrams} \qquad \text{(A5)}$$

Property 4 shows $Q^*_{0,N}$ for unseen bigrams will be larger than original $Q^*_{c,N}$ while will be decreased for all bigrams with more than one count (c>=1).

Property 5

Three notations B, S and U can be expressed as $B=S+U$ for bigram models. When the number of training size is increased, all the smoothed probability Q^* for bigrams with same counts on training size N+1 should be decreased a bit while comparing to the Q^* on training size N. The smoothed probability Q^* on N+1 training set should be less than the probability Q^* on N for $c \geq 0$, except the P^* for the incoming bigram b_{next}:

$$P^*_{c+1,N+1} = \frac{c(\bullet)+1}{N+1}. \qquad \text{(A6)}$$

In other words, in addition to the P^* of b_{next} at training size N+1, all other smoothed probability Q^* at training size N+1 will be decreased than those at training size N. Although both the numerator and denominator of Eq. (31) are increased by 1, due to $N>>c$, so the inequality equation $P^*_{c,N} < P^*_{c+1,N+1}$ will hold. In summary, property 5 can be expressed as:

$$Q^*_{c,N} > Q^*_{c,N+1} \quad \text{for all bigrams with } c >=0, \qquad \text{(A7)}$$

$$P^*_{c,N} < P^*_{c+1,N+1} \quad \text{for new bigram } b_{next}. \qquad \text{(A8)}$$

where $Q^*_{c,N}, Q^*_{c+1,N+1}$ denote the smoothed probability for bigram with c on size N and N+1.

---

[5] The property was first proposed in [Ney and Essen. 1991] and we make a little modification.