Research journal of Applied Sciences, Engineering and Technology 4(8): 902-905, 2012
ISSN: 2040-7467

Submitted: October 12, 2011 Accepted: February 09, 2012 Published: April 15, 2012

Percentage Discounting: A New Good Turing Smoothing Technique

[1]Rakhshanda Yousaf, [1]Waqas Anwar, [1]Usama Ijaz Bajwa and [2]Ali Nawaz Khan
[1]COMSATS Institute of Information Technology, Abbottabad, Pakistan
[2]COMSATS Institute of Information Technology, Lahore, Pakistan

Abstract: In this study, we have applied percentage discounting technique to overcome a drawback in Good-Turing smoothing technique. Data sparseness is an inherent and a severe problem in language modeling. Smoothing is one of the important processes to handle this problem. There are several well-known smoothing techniques which are used to solve data sparseness problem. In general, smoothing techniques ignore linguistic knowledge and are particularly based on statistical hypotheses. Good Turing is very effective for data sparseness problem but it has a drawback that it calculates zero probability if frequency of next frequency is zero. Consequently a new technique is presented in this study, which is percentage discounting technique and this technique can overcome the drawback of Good Turing smoothing.

Key words: Percentage discounting, smoothing methods, statistical language model

INTRODUCTION

Statistical Language Model (SLM) plays a vital role in many applications of natural language processing, such as speech recognition, optical character recognition, machine translation; spelling correction etc. In SLM, n-gram models play a dominant role but it needs huge amount of training corpora for reliable probability estimation. Due to the availability of limited training corpus it has inherited data sparseness problem. In such circumstances, where huge amount of training corpus is not available, it is impossible to estimate their probabilities from the observed sequences, smoothing techniques have to be used to overcome the data sparseness problem. Language model depends on some training data that is used to estimate the n-gram probabilities. It is not possible to include every word or character of a language into the training data. Resultantly there are many words that never appear in the training data and they are called unseen events. Different types of events include, unseen events (these events never occur in training data), singleton events (these events occur exactly once in training data), doubleton events (these events occur twice in training data) (Feng-Long and Ming-Shing, 2004; Chen and Goodman, 1999).

Smoothing means changing the probability distribution (assigned by a language model) so that all the sequences (even unseen) can have some probability. This often involves broadening the distribution by redistributing weight from high probability regions to zero probability regions. A lot of work has been done on devising the smoothing techniques for assigning some non zero probability to such events which never occur in the training set or occur with a very low probability i.e. approximately equal to 0. Suppose we have a vocabulary of 100,000 words, and a word is occurring only 2 times then the probability of that word is very low and approximately equal to zero. Practically, a large number of unigrams, bigrams, and trigrams may not occur in a corpus and as a result probability assigned to them is zero.

To date many smoothing methods have been used to solve the unseen words problem, such as Laplace, Lidstone, Witten Bell and Good Turing etc. In this study we are introducing a modified Good Turing algorithm by applying percentage discounting method on the original Good Turing algorithm. This modified algorithm statistically appears to be a promising approach. The mathematical explanation and the obtained results are discussed in the sections to follow. The results prove its practical applicability.

N-gram models: Ann-gramis a sequence of n grams (for e.g. words) from a givenstring or sequence. These grams may be words, letters or characters depending on the application. If size of n-gram is one, it is called unigram, if size is two it is called bigrams and so on.

When we talk about n-gram model it means a model that is used to predict the next occurring character or word in a given sequence or string, but it is difficult to compute the probability of the n-1 form $P(w_i | w_1 .... w_{i-1})$. We typically assume that the probability of a word or a sequence is dependent on the previous one word or on two previous words:

$$P(w_i | w_1 .... w_{i-1}) \approx P(w_i | w_{i-1}) \quad (1)$$

Corresponding Author: Rakhshanda Yousaf, COMSATS Institute of Information Technology, Abbottabad, Pakistan

$$P(w_i \mid w_1 .... w_{i-1}) \approx P(w_i \mid w_{i-2} w_{i-1}) \quad (2)$$

Maximum likelihood estimation: The parameters of any n-gram model are estimated by exploring a sample space and this sample space is usually called training data. Suppose C(w) is the count of number of times that a character w occurs in the string T, then for a unigram language model the probability is:

$$P(w) = C(w) / T \quad (3)$$

In the case of bigram and trigram model the computation of probability distribution is bit different then unigram model. In the bigram case the estimation of the probability is based on the counts of previous word. The probability $P(w_i \mid w_{i-1})$ is calculated as in Eq. (4), where $C(w_{i-1} w_i)$ represents the number of occurrences of w_{i-1}, w_i in the training corpus, and also compute $C(w_{i-1})$ is the count of previous word. Then we can compute (Joshua, 2001; Peter *et al*., 1992):

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1} w_i)}{C(w_{i-1})} \quad (4)$$

Similarly, in case of trigram:

$$P(w_i \mid w_{i-2} w_{i-1}) = \frac{C(w_{i-2} w_{i-1} w_i)}{C(w_{i-2} w_{i-1})} \quad (5)$$

Unfortunately, such case like in bigram and trigram the probability computation will be noisy, because there are many word sequences that never occur in the training corpus. In order to estimate the parameters on n-gram model the following equation is used:

$$\Pr(w_n \mid w_1^{n-1}) \approx \frac{C(w_1^{n-1} w_n)}{\sum_w C(w_1^{n-1} w_n)} \quad (6)$$

Discounting techniques: Smoothing process is divided into two phases. First one is discounting and second is redistributing.

Discounting: Discounting process is the first step in smoothing. In this process the probability of events are discounted. Resultantly the probabilities of seen events decrease slightly. There are two types of discounting static and dynamic.

Redistribution: Re calculated distribution is the next step to discounting for smoothing. The discounted escape probabilities calculated in discounting step are uniformly distributed to unseen events. The redistribution algorithm for all popular smoothing techniques such as Laplace, Lidstone, Witten Bell and Good Turing etc uniform distribute the escape probabilities.

Different methods have been proposed in literature for used for computing the discounted probability. Some of them are discussed.

Absolute discounting: The method interpolates the higher order and lower order models. The smoothed probability for a bigram can be understood by looking at Eq. (7), where D denoted a constant discount, c is discount count, λ is a constant with value ranging from 0 to 1 (Feng-Long and Ming-Shing, 2007):

$$P(w_n \mid w_1^{n-1}) \approx \frac{\max\{c(w_{i-1}^i) - D, 0\}}{\sum_{w_i \in i} c(w_i^i - 1)} + (1 - \lambda(w_{i=1})) P(w_i) \quad (7)$$

Good-Turing discounting: The Good-Turing (GT) smoothing technique was illustrated by Good and Turing. The basic idea of GT method is to compute the probability mass regarding zeros and low level counts based on the higher count events. According to this method, the redistribution count c* is calculated by Eq. (8), represented in term of n_c, n_{c+1} and c as (Jurafsky and Martin, 2009):

$$c^* = (c+1) \frac{N_{c+1}}{N_c} \quad (8)$$

where n_c denotes the number of n-grams with exactly c count in corpus (e.g.) n_0 represent the number of n-gram with count zero and similarly n_1 represents the number of n-grams which occur only once in the training corpus. Finally Good Turing estimator is represented as in Eq. (9)

$$\Pr(w_n \mid w_1^{n-1}) \approx \frac{c^*}{\sum_w C(w_1^{n-1} w_n)} \quad (9)$$

where this equation is a modified probability of word bigram equation which is shown in Eq. (9). The numerator in Eq. (6) is replaced by the redistributed count c* hence becomes the Good Turing estimator as Eq. (9).

Witten-Bell discounting: This method was illustrated by Witten and Bell. WB is one of the simplest estimation yet it has clever assumptions regarding zero frequency events. The basic idea of this method is Use the count of things you've seen once to help estimate the count of things you've never seen. In particular, we say that it is the probability estimation of the event occurrence first time. After assigning the probability of unseen events we have to adjust the extra probability mass. The extra probability mass of events can be adjusted by discounting the probability of all the seen events as shown in the Eq. (10) (Jurafsky and Martin, 2009).

$$\sum_{i:c_i=0} p_i^* = \frac{T}{N+T} \quad (10)$$

where T represents the number of distinct number of types and W represents the number of tokens:

$$Z = V - T \quad (11)$$

$$Z = \sum_{i:c_i=0} 1 \quad (12)$$

where Z represents the number of distinct unseen events. After computing the value of Z, each of the Z unseen events will be assigned an equal portion of the redistributed probability mass T / T+N. As far as the seen events are concerned, the extra probability mass must come by discounting the probability of all the seen events (Jurafsky and Martin, 2009):

$$p_i^* = \frac{T}{Z(N+T)} if\ (c_i = 0) \quad (13)$$

$$p_i^* = \frac{c_i}{N+T} if\ (c_i > 0) \quad (14)$$

Usefulness of Good Turing smoothing in linguistics: Good Turing smoothing is very useful in Linguistics. It is very important in natural language processing that probability of all the words or grams should be greater than zero. Applications like machine translation improve their performance if the probability of any word of interest can be calculated easily and it never turns out to be zero events if it is unseen. Generally it is important for all the applications that the probability of an event is greater than zero. Good Turing smoothing is very useful in this case, as it provides a simple way to estimate the probability of unseen events.

Problems in Good Turing smoothing: Table 1 shows the frequencies of different words along with the frequencies of these frequencies. Where c is the count of the occurrence of an event (i.e.), the frequency and n_c is the frequency of these frequencies. Normally when n decreases c increases. But it is not the case in following table. Following table shows that if value of n_c is zero in case of c = 766, the Good Turing value for some c* will be zero and as a result probability will be zero which is not desirable.

If events have higher c value then they must have higher probability but empirically it is not the case in some situations. As the table shows, the probability of events with count 767 is larger than that of events with count 765. As in the case, the next bigram has a value 0 and it can create problem in Good Turing smoothing technique (Feng-Long and Ming-Shing, 2004; Church and Gale, 1991)

Table 1: Some counts and n_c of character bigrams on mandarin text, training data N = 12M (Feng-Long and Ming-Shing, 2004)

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

Proposed percentage discounting approach: In this paper a new technique is presented for discounting probabilities from seen events and redistributing them to unseen events. This technique does not take into account the frequency of frequency concept used in Good Turing, it rather uses the probabilities of events and based on these probabilities determines the discounted or escaped probability. This discounted probability is then equally divided among the unseen events. The details of this new technique are given below:

N = Total nimber of events
S = Total seen events
U = Total unseen events
N = N + S
P_{DIS} = Discounted probability
G = U / N
k = Percentage to be discounted
Original probability of a seen events lests say A,
P(A) = C_A / N
P(U) = Probability of unseen events
P(S) = Probability of seen events
P(N) = Total probability - 1

We can discount a fixed amount of probability from total seen probabilities and then distributed it among the unseen events, using the formula (15):

$$P_{DIS} = P(S) - k\% \text{ of } P(S) \quad (15)$$

$$k \propto G$$

That means k% of total probability of seen events and is take out for unseen events. The value of k is directly proportional to G that is the ratio of unseen events to total events. If we have more unseen events, we can discount more probability and if events are few we should discount minimum possible probability. Actual value of k cannot be exactly determined but it can be estimated through experiments. Now the smoothed probabilities of unseen events become:

$$P(U) = P_{DIS}$$

Suppose D is an unseen event then its probability will be equal to:

$$P(U) = P_{DIS} / U$$

Now the probability of a seen event becomes:

Table 2: Unsmoothed data

Event	Count	Probability
A	5	0.167
B	2	0.067
C	3	0.1
D	0	0

$$P(A) = \frac{C_A}{N} - \frac{\{k\% of\ P(S)\}}{S}$$

Let N = 30, S = 18, and U be 12. Then G becomes 0.4. Let A, B, C, D be four events. Table 2 shows the probabilities (Unsmoothed data) of events before applying any smoothing technique.
Let k = 5, the total discounted probability will be:

P_{DIS} = P(S) - k% of P(S)
P_{DIS} = (18/30) - (5/100 * 18/30)
P_{DIS} = 0.6-0.03
P_{DIS} = 0.57

So the total probability of unseen events is P(U) = 0.57 and probability of D becomes:

P(D) = P_{DIS} / D
P(D) = 0.57/12 = 0.0457

Similarly probability of a seen event is discounted. So probability of A becomes:

P(A) = CA/N - {k% of P(S)}/S
P(A) = 0.167 - 0.03/18
P(A) = 0.167 - 0.001667
P(A) = 0.1653

Smoothed probabilities of rest of the events are shown in the Table 3.

Analysis of proposed technique: The proposed technique conforms to following properties:

- The smoothed probabilities of all bigrams lie between 0 and 1.
- The sum of all the probabilities is equal to 1.
- The smoothed probabilities with different counts have different values.
- The probability of all the rents is changed during smoothing process.
- When the training size is increased smoothed probabilities are decreased.

Theoretically and statistically, it seems that this technique is very useful and effective but only the experiments on real data and corpus can yield the real effectiveness and validity of this technique.

Table 3: Smoothed data (using percentage discounting method)

Event	Count	Probability
A	5	0.1653
B	2	0.0653
C	3	0.0983
D	0	0.0457

CONCLUSION

A new smoothing technique is presented in this paper. This technique discounts a constant percentage from the total probability of seen events and then distributes this probability equally among unseen events. It affects all the original probabilities and total probability is always one. This technique eliminates the drawback of Good Turing technique and it can be very useful. Experiments have not been performed yet but theoretically and statistically this seems to be a good alternate to existing techniques.

REFERENCES

Chen, S.F. and J. Goodman, 1999. An empirical study of smoothing techniques for language modeling. Comput. Speech Language, 13: 359-394.

Church, K.W. and W.A. Gale, 1991. A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilies of english bigram. Comput. Speech Language, 5: 19-54.

Feng-Long, H. and Y. Ming-Shing, 2007. Analyzing the statistical behavior of smoothing method. Innov. Adv. Tech. in Comput. Info. Sci. Eng. pp: 187–192.

Feng-Long, H. and Y. Ming-Shing, 2004. Study on Good-Turing and a Novel Smoothing Method Based on Real Corpora for Language Models. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics.

Joshua, T.G., 2001. A bit of progress in language modeling. Comput. Speech Language, 15(4): 403-434.

Jurafsky, D. and J.H. Martin, 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd Edn., Prentice-Hall.

Peter, F.B., P.V. deSouza, R.L. Mercer, V.J. Della Pietra and J.C. Lai, 1992. Class-based *n*-gram models of natural language. J. Comput. Linguist., 18(4): 467-479.