EXTENSIONS OF ABSOLUTE DISCOUNTING (KNESER-NEY METHOD)

Jesús Andrés-Ferrer[*]

Universidad Politécnica de Valencia
Valencia, Spain

H. Ney

Lehrstuhl fuer Informatik 6 RWTH Aachen University
Aachen, Germany

ABSTRACT

The problem of estimating the parameters of an n-gram language model is a typical problem of estimating small probabilities. So far, two methods have been proposed and used to handle this problem: 1. the empirical Bayes method resulting in the Turing-Good estimates. Theses estimates do not have any constraints and tend to be very noisy. 2. discounting models like absolute (or linear) discounting. The discounting models are heavily constrained and typically have only a single free parameter. Both methods can be formulated in a leaving-one-out framework.

In this paper, we study methods that lie between these two extremes. We design models with various types of constraints and derive efficient algorithms for estimating the parameters of these models. We propose two novel types of constraints or models: interval constraints and the exact extended Kneser-Ney model. The proposed methods are implemented and applied to language modelling in order to compare the methods in terms of perplexities. The results show that the new constrained methods outperform other unconstrained methods.

Index Terms— language modelling, language smoothing, leaving one out, Kneser-Ney smoothing

1. INTRODUCTION

One of the most widespread models used in language modelling is the n-gram model [1]. Due to the large number of free parameters and the always existing scarce data problem, we have to resort to smoothing techniques. The events that occur only once or not at all typically represent a huge percentage of all n-gram events. Therefore, the probabilities of these events are difficult to estimate with conventional methods since there are not enough observations of them in the training data.

A solution to the small probability problems is to apply Bayesian theory [2]. This theory introduces a prior distribution over the parameters that alleviate the estimation problems. There are several ways to select the prior distribution. From the smoothing point of view, empirical Bayes [3] is one of the most appealing ones. In particular, it results in the *leaving-one-out (LOO)* estimation [4]. Previous studies have shown that smoothing methods based on LOO are able to counteract the scarce data problems.

The first smoothing method that used the LOO esimates was the Turing-Good method [5, 6]. Unfortunately, the Turing-Good estimates are still very noisy, i.e. they are overfitted to the underlying noisy distribution produced by the data spareness. Furthemore, the Turing-Good estimates do not guarantee the monotonicity of the probaility estimates. In order to avoid these problems, absolute discounting introduced in Kneser-Ney (KN)[1] smoothing [7] assumes a smoothing model with just a single free parameter. Thus, the Turing-Good method and the absolute discounting method represent two extremes, namely either no constraints at all or a heavily constrained model with only a single parameter.

In this paper, we focus on finding a trade-off between the number of free parameters and suitable constraints in order to avoid over-trained estimates and achieve optimum performance. This idea was previously outlined in [7], where monotonic and interval constraints were suggested. The novel contributions of this paper are the following:

- We propose new discounting methods that lie between the two extremes mentioned, namely the Turing-Good method and absolute discounting (Kneser-Ney).
- We show how the associated estimates can be computed efficiently from the training data.
- We carry out systematic experiments for two language modelling tasks and compare the performance of these methods.

The paper is organised as follows. First, an introduction to language modelling is covered in section 2. In section 3, the LOO method is reviewed to pave the way for the new proposals. The novel estimating methods are introduced in section 4. In section 5, we report experimental results. Concluding remarks are discussed in section 6.

2. LANGUAGE MODELLING

Language modelling (LM) consists in modelling the probability of a word sequence, w_1^L. One of the most widespread techniques for LM is the n-gram models [8], where the probability, $p(w_1^L)$, is modelled as follows

$$p(w_1^L) = \prod_{l=1}^{L} p(w_l|w_1^{l-1}) = \prod_{l=1}^{L} p(w_l|h) \tag{1}$$

[*] Work partially supported by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), by the EC (FEDER), the Spanish MEC under grant TIN2006-15694-CO2-01 and the Valencian "Conselleria d'Empresa, Universitat i Ciència" under grant CTBPRA/2005/004.

[1] We adhere to the widely used terminology, with apologies by the second author for this breach of modesty.

where the previous word history w_1^{l-1} is approximated by the $n-1$ most recent words, h.

In order to alleviate problems derived from scarce data several smoothing techniques for n-gram models have been proposed. All these smoothing techniques, discount a probability mass B_h from all seen n-grams and for each history h; and, then, redistribute it according to a smoothing distribution, $\beta(\cdot)$. For instance, the linear interpolation [1] distributes the gained probability mass B_h among all words according to the smoothing distribution $\beta(w, \bar{h})$. On the other hand, the backing-off redistributes the probability only among the unseen events, according to $\beta(w, \bar{h})$. Note that $\bar{h}$ stands for the previous history h where we have dropped the furthest word.

Some of the smoothing techniques [1] are based on the LOO estimation. For instance, Turing-Good smoothing [1], the KN [8], or Katz's smoothing [4].

The first proposed smoothing based on LOO, Turing-Good degradates the probability estimates as the n-gram occurs more frequently. The KN smoothing solved this problem by approximating all the probabilities with just one parameter. In this work we present some novel estimation methods to avoid the sparsity problems for the Turing-Good counts.

Given a training dataset, we define $N(w, h)$ as the number of occurrences of the n-gram hw in this dataset. For any smoothing technique based on the LOO approach, we define the *modified counts* r^* as

$$r^* = p_r N \tag{2}$$

where p_r is joint probability assigned by the LOO smoothing to the n-grams which have occurred r times in the corpus. For a given n-gram such that $N(w, h) = r$, we will use $N^*(w, h)$ to refer to the modified count, r^*.

Using these modified counts, the smoothed probability is defined as follows

$$\tilde{p}(w|h) = \begin{cases} \frac{N^*(w,h)}{N(h)} & N(w,h) > 0 \\ B_h \beta(w|\bar{h}) & \text{otherwise} \end{cases} \tag{3}$$

with $B_h = [\sum_{w:N(w,h)>0}(N(w, h) - N^*(w, h))]/N(h)$ and $N(h) = \sum_w N(w, h)$; and where the distribution $\beta(w|\bar{h})$ is also obtained by LOO [8].

3. REVIEW OF UNCONSTRAINED LEAVING ONE OUT (LOO)

The leaving-one-out (LOO) estimation technique was introduced for language modelling in [8] as a smoothing technique and can be summarized as follows. Each possible event, (w, h), is assigned its count in the training data, $N(w, h)$. We form equivalence classes by gathering all events with the same count $r = N(w, h)$ into the same equivalence class. Then, we count the number of events in each class with count $r = 0, 1, ..., R$ and denote them by n_r. These quantities are often referred to as *counts of counts (COC)* because they count how often each count r occurs in the training data. In particular, the count n_0 refers to the number of events that have not been observed in the training data.

By leaving one out, an observation in equivalence class with count r is moved into the equivalence class with count $r-1$. Therefore, the associated probability $p_r = p(w, h)$ with $N(w, h) = r$, is replaced by p_{r-1}. If we repeat this process for all equivalence classes $r = 1, ..., R$, we obtain the LOO log-likelihood as a function of the unknown probabilities $p_0^{R-1} := p_0, p_1, ..., p_{R-1}$ as follows

$$F(p_0^{R-1}) = \sum_{r=1}^{R} r n_r \log p_{r-1} \tag{4}$$

Since there are n_r events each with probability p_r, the following constraint must be satisfied

$$\textstyle\sum_{r=0}^{R} n_r p_r = 1 \tag{5}$$

The probability p_R is not used in the LOO log-likelihood function. Instead, we estimate it by relative frequency

$$p_R = R/N \tag{6}$$

The optimal probability estimates are obtained when Eq. (4) is maximised constrained by Eqs. (5) and (6)

$$p_r = \frac{(1 - n_R p_R)}{N} \frac{(r+1)n_{r+1}}{n_r} \qquad r = 0, \ldots, R-1 \tag{7}$$

We typically want the LOO estimates to be 'close' to the relative frequencies, since the conventional maximum likelihood approach results in those relative frequencies

$$p_r = r/N \qquad r = 0, \ldots, R-1 \tag{8}$$

4. CONSTRAINED LEAVING ONE OUT

The original LOO estimates as introduced in the previous section suffer from over-training. In particular they do not need to be monotonic. In this section, we will constrain the LOO probabilities p_r with interval constraints to enforce the monotonicity of the probability estimates p_r. Additionally, we extend the modified Kneser-Ney (mKN) giving the optimal analytical solution.

4.1. Interval Constraints

The goal of this method is to modify the conventional probabilities as given in Eq. (8) only a little bit. Therefore, we introduce what we call the interval constraints

$$(r-1)/N \ \leq p_r \leq \ r/N \qquad r = 1, \ldots, R-1 \tag{9}$$

$$p_0 \ \leq 1/N \tag{10}$$

The idea of applying this constraints was previously outlined in [7], where a heuristic and not optimal solution was proposed and analysed. In order to obtain an optimal solution to the problem we use the Karush-Kuhn-Tucker (KKT) conditions [9].

The KKT conditions result in estimates that depend on a normalisation constant λ

$$p_r(\lambda) = \max\{\frac{r-1}{N}, \min\{\frac{1}{\lambda}\frac{(r+1)n_{r+1}}{n_r}, \frac{r}{N}\}\} \tag{11}$$

The interpretation of this equation is as follows. We compute the unconstrained LOO estimate $p_r = (1/\lambda)[(r+1)n_{r+1}]/n_r$, with the unknown normalisation constant λ. This estimate is then compared with the lower and upper bound; and finally, it is clipped if necessary. Now the problem is that this comparison requires the normalisation constant to be known. To this purpose we introduce the λ depending *normalisation function*

$$Q(\lambda) = \sum_{r=0}^{R} n_r p_r(\lambda) \tag{12}$$

Therefore, the normalisation constraint is reformulated as $Q(\lambda) = 1$. Since $Q(\lambda)$ is a monotonically decreasing function, the value for λ can be easily computed.

Note that in order to ensure monotonicity the constraint $p_0 \leq p_1$ must be added to the algorithm. However, its addition does not significantly modify the algorithm, though it becomes more awkward. Anyway, this constraint is always verified in practice, and hence, it becomes useless.

4.2. Exact extended Kneser-Ney smoothing

The extended KN smoothing [7] method reduces the number of free parameters by using an absolute discounting model for counts larger than a given threshold S

$$p_r = \frac{(r-d)}{N} \qquad \forall r \geq S \tag{13}$$

where the parameter d is the so-called discounting parameter. Obviously, this method does not guarantee that the remaining probabilities p_r for $r = 0, 1, ..., S-1$ are monotonic. Whether monotonicity is satisfied or not depends on the training data and the given threshold S.

This estimation technique was initially presented with a fixed threshold, $S = 1$ [8], and afterwards extended to $S = 3$ [1]. Nevertheless, no exact solution was given for the estimation if $S > 1$. In this section, we analyse the exact solution for this estimation approach using LOO.

Since the probabilities, p_r, with r larger than $S-1$ depend on d as expressed in Eq. (13); the LOO log-likelihood function in Eq. (4) is rewritten to

$$F(p_0^{S-1}, d) = \sum_{r=1}^{S} r n_r \log p_{r-1} + \sum_{r=S+1}^{R} r n_r \log \frac{r-1-d}{N} \tag{14}$$

subject to the normalisation constraint rewritten as

$$\sum_{r=0}^{S-1} n_r p_r + \sum_{r=S}^{R} n_r \frac{r-d}{N} = 1 \tag{15}$$

The solution that maximises Eq. (14) constrained by Eq. (15) is given by

$$p_r(d) = \frac{1}{\lambda(d)} \frac{(r+1)n_{r+1}}{n_r}, \quad r = 0, \ldots, S-1 \tag{16}$$

where the normalisation constant depends on d as follows

$$\lambda(d) = \left(\sum_{r=S+1}^{R} \frac{r n_r}{r-1-d} \right) \left(\sum_{r=S}^{R} \frac{n_r}{N} \right)^{-1} \tag{17}$$

Table 1. Some basic statistics for the Wall Street Journal.

Training	WSJ
sentences	1.62M
avg. length	26.0
running words	42.12M
vocab. size	200.1K
n_1/N (1-gram)	0.17%
n_1/N (2-gram)	18.1%
n_1/N (3-gram)	28.5%

Table 2. Out of vocabulary words (OOV) in test for each training partitions.

	50K	100K	$1,000$K	full size
OOV Rate [%]	1.7	1.1	0.3	0.3

Similarly to section 4.1, we reformulate the normalisation constraint in Eq. (15) by defining $Q''(d)$ as follows

$$Q''(d) = \sum_{r=0}^{S-1} n_r p_r(d) + \sum_{r=S}^{R} n_r \frac{r-d}{N} \tag{18}$$

and requiring it to be equal to 1, $Q''(d) = 1$.

The function $Q''(d)$ is again monotonically decreasing, and therefore it is straightforward to find the optimal value $\hat{d}$ such that $Q''(\hat{d}) = 1$

Unlike original and modified KN, we have not made any approximation in order to obtain the exact value for $\hat{d}$ and p_0. Moreover, the threshold count S is not fixed beforehand to be neither 1 (KN), nor 3 (mKN).

5. EXPERIMENTS

In this section, the practical performance of the proposed estimation techniques is analysed in a language modeling task. The perplexity [10] on a test set will be used to compare among the techniques. The less the perplexity is, the better the model is. In order to quantify the impact of the techniques, we have compared all the techniques with the baseline perplexity given by the mKN smoothing [1] and original KN [7]. In order to obtain this baseline, we have used the standard tool SRILM [11].

Table 1 summarises some statistics about the corpus used in the experiments: the Wall Street Journal (WSJ) [8]. For the test set, we have selected an small percentage of paragraphs from all the years, in order to gain independence on the test set with respect to time factors. The test set is made up of 12.5K sentences with an average length of 26.1 comprising 326.3K running words. Finally, in order to analyse the behaviour of all the techniques as a function of the training size, we have splitted the training into increasing sizes ranging from 50K sentences to the full corpus.

We used all the training vocabulary in order to carry out all the experimentation. For modelling the *out of vocabulary*

Table 3. Perplexities for trigram language models on the corpus. *Sk OOV* column stands for the perplexity skipping the OOV, while the *All* column accumulates all the events (OOV and known).

N. of trainig sentences	50K		100K		1M		1.62M	
	All	Sk OOV	All	Sk OOV	All	Sk OOV	All	Sk OOV
modified Kneser-Ney	154.0	154.3	137.7	136.7	94.2	93.4	87.3	86.7
Kneser-Ney	150.5	149.2	134.6	132.7	92.6	91.7	85.9	85.2
Interval	**150.0**	**148.0**	**134.2**	**131.9**	92.4	91.4	85.7	84.9
extended exact KN ($S = 3$)	151.4	149.6	134.8	132.4	**90.8**	**89.8**	**83.8**	**83.0**

words (OOV) we reserved the smoothing probability mass for the unigram unseen events. In order to do so, the full vocabulary size must be known. Although, any sensible estimation of the size suffices, we have extrapolated the number of unseen words in the vocabulary from the seen words. Moreover, we also report perplexities skipping OOV in order to quantify the influence of the unknown events. Table 2 depicts the percentage of OOV in test as a function of the training size of our corpus partition.

The influence of the threshold count S (for the eeKN) in the perplexity is not significant since moderate values obtain similar perplexities as long as the training data is not scarce. For instance, if the training is performed on the full corpus, then the perplexity is within the range $[83.75, 83.91]$, for all $S = 1, 2, \ldots, 128$.

Table 3 summarises the perplexities obtained using a trigram language model. We have obtained results for bigrams and fourgrams, as well, showing a similar behaviour. From the table we conclude that all the proposed techniques perform at least as baseline techniques, being better under some circumstances. The best technique for small training data is the interval constrained LOO, which obtain the best results for scarce training sizes (50k and 100K). On the other hand, as the size of the corpus increases, the best technique is the exact extended Kneser-Ney.

6. CONCLUSIONS

Conventional smoothing models based on leaving one out estimates represent two extremes. On the one extreme the absolute discounting (KN) reduces the number of parameters to estimate to one. On the other extreme the Turing-Good estimates all the LOO probabilities, producing over-fitted probabilities.

In this paper we have proposed novel discounting methods that are less restrictive than absolute discounting approaches, but more restrictive than Turing-Good method. Therefore, we explore the gap in which we try to optimise the tradeoff between the number of parameters and data scarcity.

Specifically, we have proposed two novel discounting methods based on constraining leaving one out estimates: interval constraints and the exact extended Kneser-Ney smoothing. We have also presented the associated estimation algorithms needed to compute the discounted estimates in an efficient way. Systematic experiments have also been performed in order to compare the proposed methods with other standard discounting methods. This experimentation reports improvements over the baseline smoothing under some circunstances.

7. REFERENCES

[1] Stanley Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modelling," *In Proc. annual meeting of the ACL*, pp. 310–318, 1996.

[2] T. Lwin and J. S. Maritz, *Empirical Bayes methods*, Chapman and Hall, 1989.

[3] Herbert Robbins, "An empirical bayes approach to statistics," *Proceeding of the Third Berkeley Symposium on Mathematical Statistics*, vol. 1, pp. 157–163, 1956.

[4] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *In IEEE Transactions on ASSP*, vol. ASSP-35, pp. 400–401, 1987.

[5] I. J. Good, "Population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, pp. 237–264, 1953.

[6] A. Nadas, "On turing's formula for word probabilities," *IEEE Trans. ASSP*, vol. 33, pp. 1,414–1,416, 1985.

[7] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling.," *IEEE Int. Conf. on ASSP*, vol. II, pp. 181–184, 1995.

[8] H. Ney, U. Essen, and R. Kneser, "On the estimation of 'small' probabilities by leaving-one-out," *IEEE Trans. PAMI.*, vol. 17, no. 12, pp. 1202–1212, 1995.

[9] Stephen Boyd and Lieven Vandenberghe, "Convex optimization," pp. 244–254, March 2004.

[10] P. F. Brown, V. J. Della Pietra, R. L. Mercer, S. A. Della Pietra, and J. C. Lai, "An estimate of an upper bound for the entropy of english," *Comput. Linguist.*, vol. 18, no. 1, pp. 31–40, 1992.

[11] A. Stolcke, "Srilm - an extensible language modeling toolkit," *In Proc. International Conference on Spoken Language Processing*, vol. 2, pp. 901–904, Sep. 2002.