Effect of Feature Smoothing Methods in Text Classification Tasks

David Vilar[1], Hermann Ney[1], Alfons Juan[2], and Enrique Vidal[2]

[1] Lehrstuhl für Informatik VI
Computer Science Department
RWTH Aachen University
D-52056 Aachen (Germany)
{vilar, ney}@i6.informatik.rwth-aachen.de
[2] Institut Tecnològic d'Informàtica
Universitat Politècnica de València
E-46071 València (Spain)
{ajuan, evidal}@iti.upv.es

Abstract. The number of features to be considered in a text classification system is given by the size of the vocabulary and this is normally in the range of the tens or hundreds of thousands even for small tasks. This leads to parameter estimation problems for statistical based methods and countermeasures have to be found. One of the most widely used methods consists of reducing the size of the vocabulary according to a well defined criterion in order to be able to reliably estimate the set of parameters. In the field of language modeling this problem is also encountered and several smoothing techniques have been developed. In this paper we show that using the full vocabulary together with a suitable choice of the smoothing technique for the text classification task obtains better results than the standard feature selection techniques.

Key words: Text Classification, Naive Bayes, Multinomial Distribution, Feature Selection, Smoothing, Length Normalization

1 Introduction

Text classification systems, even for small tasks, have to deal with vocabularies of thousands or tens of thousands of words, which form the effective dimensions of the representation space of the documents to classify. This often leads to parameter estimation problems due to the sparseness of the data, as a high percentage of the words will rarely be seen and the parameters of the models can not be reliably estimated. As an example, in the 20 newsgroups data set, more than half of the words are seen two times or less. To counteract this problem, a frequent solution consists of using only a reduced subset of the vocabulary, selected according to a well defined criterion, in order to reduce the number of parameters to be estimated, and thus trying to obtain more accurate values.

Similar problems are also found in the field of language modeling, where the most widely used models, the n-gram models, are also subjected to this data sparseness problem. The most frequent solution in this area is to use feature smoothing techniques in order to redistribute the original probability mass and so to achieve a good estimate even for unseen events.

In this paper we have used some of this techniques adapted to the text classification task, and for four out of five corpora we obtain better results by using the whole vocabulary instead of a reduced set.

This paper is organized as follows. Section 2 presents the basic model we will use for our experiments. Section 3 describes the feature selection technique most widely applied and Section 4 presents the feature smoothing technique we will use. The results for the different corpora are shown in Section 5 and lastly some conclusions are drawn in Section 6.

2 The multinomial model

As representation of the documents we use the well-known bag-of-words representation, that is, each document is assigned a D-dimensional vector of word counts, where D is the size of the (possibly reduced) vocabulary. We will denote the word variable as $w = 1, \ldots, W$ and the document class variable as $c = 1, \ldots, C$. As classification model we use the *naive Bayes* text classifier in its *multinomial* event model instantiation [1]. In this model the assumption is made that the probability of each event (word occurrence) is independent of the word's context and position in the document it appears, and thus the chosen representation is justified. Given the representation of a document by its counts $\boldsymbol{x} = (x_1, \ldots, x_W)^t$ the class-conditional probability is given by the multinomial distribution

$$p(\boldsymbol{x}|c) = p(x_+|c)p(\boldsymbol{x}|c, x_+) = p(x_+|c)\frac{x_+!}{\prod_w x_w!}\prod_w p(w|c, x_+)^{x_w} , \tag{1}$$

where $x_+ = \sum_w x_w$ is the length of document $\boldsymbol{x}$, and $p(w|c, x_+)$ are the parameters of the distribution, with the restriction

$$\sum_w p(w|c, x_+) = 1 \quad \forall c, x_+ . \tag{2}$$

In order to reduce the number of parameters to estimate, we assume that the distribution parameters are independent of the length x_+ and thus $p(w|c, x_+) = p(w|c)$, and that the length distribution is independent of the class c, so (1) becomes

$$p(\boldsymbol{x}|c) = p(x_+)\frac{x_+!}{\prod_w x_w!}\prod_w p(w|c)^{x_w} . \tag{3}$$

Applying Bayes rule we obtain the classification rule

$$r(\boldsymbol{x}) = \operatorname*{argmax}_c \{\log p(c)p(\boldsymbol{x}|c)\} = \operatorname*{argmax}_c \left\{ \log p(c) + \sum_w x_w \log p(w|c) \right\} \tag{4}$$

To estimate the prior probabilities $p(c)$ of the class and the parameters $p(w|c)$ we apply the maximum-likelihood method. For a given training set $\{(x_n, c_n)\}_{n=1}^N$, where

x_n is the representation of the nth document[3], the log-likelihood function is

$$\log \mathcal{L}(\{p(c)\}, \{p(w|c)\}) = \sum_{n=1}^{N} \left(\log p(c_n) + \sum_{w} x_{nw} \log p(w|c_n) \right.$$
$$\left. + \operatorname{const}(\{p(c)\}, \{p(w|c)\}) \right) . \tag{5}$$

Using Lagrange multipliers we maximize this function under the constraints

$$\sum_{c} p(c) = 1 \qquad \text{and} \qquad \sum_{w} p(w|c) = 1, \quad \forall 1 \leq c \leq C \tag{6}$$

The resulting estimators[4] are the relative frequencies

$$\hat{p}(c) = \frac{N_c}{N} \tag{7}$$

and

$$\hat{p}(w|c) = \frac{N_{cw}}{\sum_{w'} N_{cw'}} , \tag{8}$$

where $N_c = \sum_n \delta(c_n, c)$ is the number of documents of class c and similarly $N_{cw} = \sum_n \delta(c_n, c) x_{nw}$ is the total number of occurrences of word w in all the documents of class c. In this equations $\delta(\cdot, \cdot)$ denotes the Kronecker delta function, which is equal to one if its both arguments are equal and zero otherwise.

From equation (8) it can be observed that if a word w has not been seen in training for a class c, the corresponding parameter $p(w|c)$ will be estimated as 0. If in the test phase a document belonging to this class contains this word, the conditional probability will also be 0 (see eq (3)), which will produce a classification error. This problem is known as *data sparseness*, and is caused by the fact that the amount of possible features is much larger than the available data. In our case a feature is a pair (w, c), composed by a word and a class, (compare with the distribution parameters $p(w|c)$) and, as stated in the example, many of the words will be seen only in a reduced set of classes during training. Two solutions to this problem will be discussed in the next sections: feature selection and feature smoothing.

3 Feature Selection

Feature selection techniques aim to reduce the number of features to take into consideration without degrading the performance of the system. The use of such techniques is mandatory for certain classifiers like neural networks or Bayes belief networks, where a high dimensionality of the input space implies an intractable number of parameters to estimate. Nevertheless it is reported that for some corpora and several classification

[3] Analogously x_{nw} is the count of word w for document x_n.

[4] We will denote parameter estimations with the hat ($\hat{\ }$) symbol.

techniques, reducing the size of the vocabulary effectively improves classification accuracy by considering only those parameters which can be reliably estimated [1–3]. For the classifier we are considering the efficiency consideration is not crucial except in some special cases, if strict efficiency requirements must be met.

The most widely used feature selection technique which obtains the best results is known as *information gain* [4], based on the mutual information[5] concept of information theory [5]. It is a measure of the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. For a word w, the information gain is defined as

$$\begin{aligned} G(w) &= \sum_{c=1}^{C} \sum_{\omega \in \{w, \overline{w}\}} p(c, \omega) \log \frac{p(c, \omega)}{p(c)p(\omega)} \\ &= -\sum_{c=1}^{C} p(c) \log p(c) + p(w) \sum_{c=1}^{C} p(c|w) \log p(c|w) \\ &\quad + p(\overline{w}) \sum_{c=1}^{C} p(c|\overline{w}) \log p(c|\overline{w}) \,, \end{aligned} \tag{9}$$

where $\overline{w}$ denotes the absence of word w. Having computed this value for each word of the vocabulary, we use as classification features only those with information gain above a predefined threshold or, more frequently, the highest scoring $\mathcal{W}$ words.

Nevertheless, reducing the amount of features in general can not guarantee a solution for the "zero-frequency" problem. A frequent approach to solve it within this context is to use the so called *Laplace* estimator [1], where the effect of including one document with each word appearing exactly once is simulated for each class. As we will see in the next section, this is a simple form of smoothing and better results can be obtained using more refined approaches.

4 Feature Smoothing

Parameter smoothing is required to counteract the effect of statistical variability of the training data, particularly when the number of parameters to estimate is relatively large in comparison with the amount of available data. A clear example of this effect are the multinomial parameters whose value are set to 0 according to the maximum likelihood estimation.

One simple case of parameter smoothing, known as Laplace smoothing, consists simply of adding a pseudo-count to every word-count

$$\hat{p}(w|c) = \frac{N_{cw} + \epsilon}{\sum_{w'} (N_{cw'} + \epsilon)} \,. \tag{10}$$

[5] It is important to distinguish between the information theoretical concept of mutual information and the (related but different) criterion of mutual information for feature selection (see [4]).

The Laplace estimator mentioned in Section 3 is the special case of (10) when $\epsilon = 1$. Also, this special case can be seen as the result of a Bayesian estimation method in which a Dirichlet prior over word probabilities is used [6]. Although this approach avoids zero probabilities, we find that it can not achieve an effective redistribution of the probability mass. This problem has been extensively studied in the context of *statistical language modeling* [7] and the application to text classification tasks is presented in [8]. In this paper four different techniques are studied on the well known 20 newsgroups corpus (see also Section 5). Further experiments on more corpora have shown that the technique known as *unigram interpolation* usually achieves the best results and, in order to focus our exposition, we will only reproduce the derivation of this method here.

The base of this method is known as *absolute discounting* and it consist of gaining "free" probabilities mass from the seen events by discounting a small constant b to every (positive) word count. The idea behind this model is to leave the high counts virtually unchanged, with the justification that for a corpus of approximately the same size, the counts will not differ much, and we can consider the "average" value, using a non-integer discounting. The gained probability mass[6] for each class c is

$$M_c = \frac{b \cdot \left|\{w' : N_{cw'} > b\}\right|}{\sum_{w'} N_{cw'}}\,, \tag{11}$$

and is distributed in accordance to a *generalized distribution*, in our case, the *unigram distribution*

$$p(w) = \frac{\sum_c N_{cw}}{\sum_{w'} \sum_c N_{cw'}}\,. \tag{12}$$

The final estimation thus becomes

$$\hat{p}(w|c) = \max\left\{0, \frac{N_{cw} - b}{\sum_{w'} N_{cw'}}\right\} + p(w) M_c\,. \tag{13}$$

The selection of the discounting parameter b is crucial for the performance of the classifier. A possible way to estimate it is using the so called *leaving-one-out* technique. This can be considered as an extension of the cross-validation method [9, 10]. The main idea is to split the N observations (documents) of the training corpus into $N - 1$ observations that serve as training part and only 1 observation, the so called hold-out part, that will constitute the simulated training test. This process is repeated N times in such a way that every observation eventually constitutes the hold-out set. The main advantage of this method is that each observation is used for both the training and the hold-out part and thus we achieve and efficient exploitation of the given data. For the actual parameter estimation we again use maximum likelihood. For further details the reader is referred to [7].

No closed form solution for the estimation of b using leaving-one-out can be given. Nevertheless, an interval for the value of this parameter can be explicitly calculated as

$$\frac{n_1}{n_1 + 2n_2 + \sum_{r \geq 3} n_r} < b < \frac{n_1}{n_1 + 2n_2}\,. \tag{14}$$

[6] Normally the numerator of (11) would be $b \cdot |\{w' : N_{cw'} > 0\}|$. Allowing the generalization presented in the main text allows us to use discounting parameters greater than 1, which will be specially interesting when we consider document length normalization (see 4.1).

where $n_r = \sum_w \delta(\sum_c N_{cw}, r)$ is the number of words that have been seen exactly r times in the training set. Since in general leaving-one-out tends to underestimate the effect of unseen events we choose to use the upper bound as the leaving-one-out estimate

$$\hat{b}_{llo} \cong \frac{n_1}{n_1 + n_2} \tag{15}$$

Comparing the results with this estimation and with the optimum parameter determined on the test set for full vocabulary, in which can be considered a "cheating" experiment, we observed that this estimate performs very well on every corpus, as nearly no classification accuracy, if any at all, is lost.

4.1 Document length normalization

The multinomial naive Bayes text classifier is biased towards correctly classifying long documents due to the unrealistic assumption that the class-conditional word posterior probabilities are independent of the document length. Because of this assumption the estimate (8) is dominated by the word counts coming from long documents.

One possible solution to this problem is to normalize the word counts of each document with respect to its length

$$\tilde{x}_w = L \frac{x_w}{x_+}, \quad \forall 1 \leq w \leq W \,, \tag{16}$$

where L can be any arbitrary constant, such as the average document length.

Multinomial distributions with fractional counts are ill-defined. Nevertheless the derivations made in section 2 are extensible to fractional counts and so the estimate (8) is still valid. Another point to note is that the classification rule (4) is invariant to length normalization, so test documents can be classified without prior normalization. The smoothing techniques presented in 4 can also be directly applied, but the leaving-one-out estimate can not be easily adapted to this situation.

5 Experiments

For our experiments we used five different corpora: the 20 Newsgroups data set, the Industry Sector data set, the 7 Sectors data set, the WhizBang! Job Categorization data set and the 4 Universities data set.

The Industry Sector data set, made available by Market Guide Inc., and the 7 Sectors data set from World Wide Knowledge Base (Web→KB) project of the CMU Text Learning Group [11], consist both of collections of web pages from different companies, divided into a hierarchy of classes. In our experiments, however, we have "flattened" this structure, assigning each document a class consisting of the whole path to the document in the hierarchy tree.

The WhizBang! Job Categorization data set consist of job titles and descriptions, also organized in a hierarchy of classes. This corpus contains labeled and unlabeled samples and only the former were used in our experiments.

Table 1. Corpus statistics

Corpus	#Documents	#Classes	Vocabulary	Avg. doc. length
Industry Sector	9 637	105	59 132	118.4
7 Sectors	4 572	48	36 056	101.8
Job Category	131 643	65	77 267	67.5
20 Newsgroups	19 974	20	93 508	86.4
4 Universities	4 199	4	40 509	130.2

The 20 Newsgroups data set is a collection of approximately 20 000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. We used the original version of this data as provided in www.al.mit.edu/~jrennie, in which document headers are discarded, but the "From:" and "Subject:" header files are retained. The documents were sorted by their posting date, the first 800 documents of each class were used for training and the rest for testing.

The 4 Universities data set, also available from the CMU Web→KB project, consists of a set of web pages from the computer science departments of 4 different universities. There is a total of 7 classes defined, but according to the usual procedure only the four most populated ones[7] were used in the experiments. It is also usual practice to train with data of three universities and test with the data of the remaining university. The results presented here are therefore the average values of the four experiments.

The statistics of each corpus are shown in Table 1. Unless stated otherwise (i.e. in the 20 newsgroups and 4 Universities data sets) the corpora were randomly split into a training set consisting of approximately 80% of the samples for training and the remaining 20% for test.

Figure 1 shows the error rate as a function of the vocabulary size for all the corpora. It can be clearly seen that only for small vocabulary sizes the maximum likelihood estimator can be directly used. Using the "traditional" Laplace smoothing to avoid zero probabilities the best results are achieved using a reduced vocabulary set in three of the five corpora, as claimed in previous works. Using the unigram interpolation smoothing technique, however, better results are obtained, and in the first four corpora the best performance is achieved using the whole vocabulary set.

The remaining corpus, the 4 Universities data set, presents an anomalous behavior, as shown in figure 1(e). The best results are obtained with an extremely reduced vocabulary set (100 words) and the evolution of the error rate is rather irregular and does not correspond to the expected behavior as observed in the other corpora. In this case we hardly improve the error rate using any smoothing technique. We feel that no significant conclusions can be extrapolated from this corpus. The results are summarized in Table 2.

We also found out that length normalization increases the accuracy of the classifier, an example can be seen in Figure 2. However in this case we can not make use of Equation (13) as the "count counts" n_r are not well defined, and the parameter b was

[7] Without taking the class "others" into account.

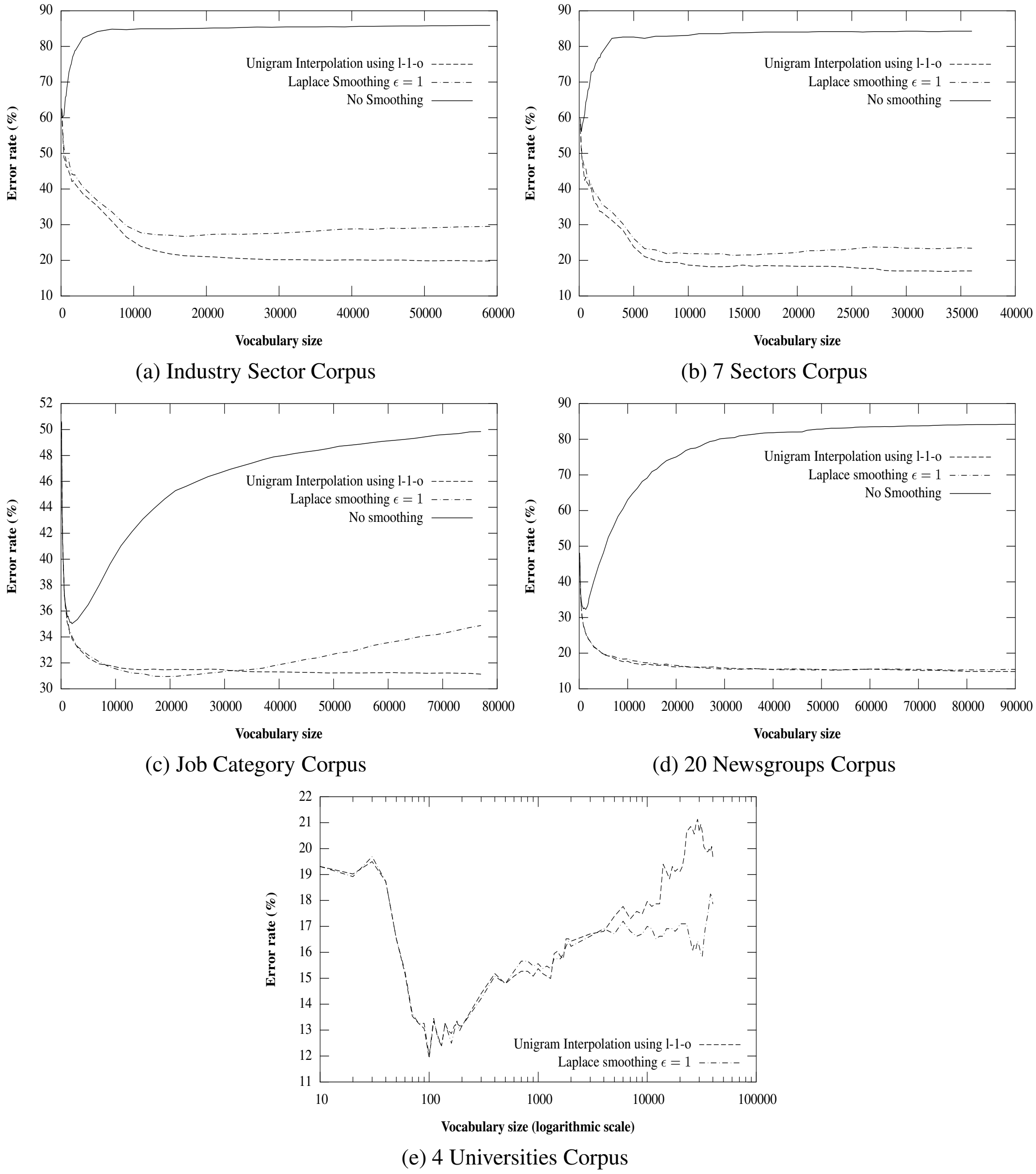

Fig. 1. Error rate as a function of the vocabulary size for different corpora

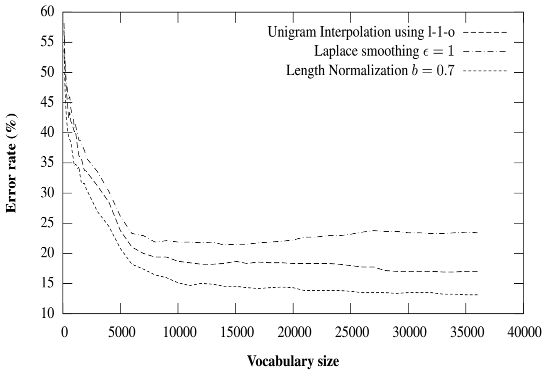

Fig. 2. Effect of length normalization on the corpus 7 sectors

empirically estimated on the test set, and therefore the results are not directly comparable. Somehow surprisingly in the Job category data set the best results are obtained using the simpler Laplace smoothing with smoothing parameter $\epsilon = 0.2$.

Table 2. Summary of classification error rates for the five tasks, using the optinum number of features.

	Smoothing method		
Corpus	None	Laplace	Abs. disc.
Industry-Sector	60.1	26.7	19.8
7 Sectors	56.2	21.4	16.9
Job category	35.0	31.2	31.1
20 Newsgroups	32.3	15.3	14.9
4 Universities	12.1	12.0	11.9

6 Concluding remarks

We have shown that for all of the corpora, using absolute discounting smoothing we obtain the best results. For four out of the five tested corpora, the best results are obtained using the whole vocabulary set. This is a satisfying result and shows that the applied smoothing techniques effectively redistribute the probability mass among the unseen events.

We also have shown that using length normalization we usually achieve better results. However the experiments we have performed were optimized on the test set, in order to try if this method could achieve better results. The next natural step is to find a well-defined estimation for the discounting parameter. Another conclusion from these results is that the length independence assumptions we made in Section 2 are too unrealistic and perhaps an explicit length model has to be included in our general formulation.

We feel that better results could be achieved by improving the feature selection techniques and perhaps including a weighting of the different terms, in a similar way as it is done in prototype selection for k nearest neighbors classifiers.

References

1. McCallum, A., Nigam, K.: A comparison of event models for naive Bayes text classification. In: AAAI/ICML-98 Workshop on Learning for Text Categorization, AAAI Press (1998) 41–48
2. Lafuente, J., Juan, A.: Comparación de Codificaciones de Documentos para Clasificación con K Vecinos Más Próximos. In: Proc. of the I Jornadas de Tratamiento y Recuperación de Información (JOTRI), València (Spain) (2002) 37–44 (In spanish).
3. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification (1999)
4. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, US (1997) 412–420
5. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA (1991)
6. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning **39** (2000) 103–134
7. Ney, H., Martin, S., Wessel, F.: Satistical Language Modeling Using Leaving-One-Out. In: Corpus-based Methods in Language and Speech Proceesing. Kluwer Academic Publishers, Dordrecht, the Netherlands (1997) 174–207
8. Juan, A., Ney, H.: Reversing and Smoothing the Multinomial Naive Bayes Text Classifier. In: Proc. of the 2nd Int. Workshop on Pattern Recognition in Information Systems (PRIS 2002), Alacant (Spain) (2002) 200–212
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, New York, NY, USA (2001)
10. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York, NY, USA (1993)
11. Group, C.T.L.: World wide knowledge base (web→kb) project. (http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/)
12. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In Joshi, A., Palmer, M., eds.: Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, San Francisco, Morgan Kaufmann Publishers (1996) 310–318