STAIRS 2014
U. Endriss and J. Leite (Eds.)

doi:10.3233/978-1-61499-421-3-290

Empirical Study of Classification Models for Web Page Categorization

Tomáš TUNYS [a], Jan ŠEDIVÝ [b]
[a] *Czech Technical University in Prague, tunystom@fel.cvut.cz*
[b] *Czech Technical University in Prague, sedivja2@fel.cvut.cz*

Abstract. We describe one part of a web page content classification system used for contextual advertising. The system classifies the content of a web page to one of many predefined ad categories, i.e. performing text classification [2,8,10]. Our goal is to identify the best performing classification model for Czech language. We present a comprehensive comparison study of selected models, text representations, and feature selection techniques on a collection of datasets with different numbers of documents, numbers of categories, and varying category sizes. We conclude the work with the recommendation for the best performing models.

Keywords. text classification, text representation, feature selection, model comparison, naive bayes model, dirichlet compound multinomial

Introduction

Text classification is a frequently studied problem mostly with documents in English. Our task was to find a good solution for Czech, which is morphologically more complex language, and despite the fact that we believe there is no strong evidence that best models for English should not work as well for Czech, we wanted to prove this assumption. We approached the problem by comparing currently successful models for text classification in English with models for Czech.

For comparison purposes we have selected publicly available datasets of different sizes and different number of categories. For validity of our results across foreign-language corpora, we paid a lot of attention to select publicly available corpora, which are comparable to our Czech datasets in the size, number of classes, and content (web pages). The chosen datasets are described in detail in Section 2.

The collection of studied models starts with a simple model such as Multinomial Naive Bayes model together with its advanced versions referred to as Transformed Weight-normalized Complement Multinomial Naive Bayes models [7], continuing with more elaborate generative models such as Dirichlet Compound Multinomial model [4]. For completeness we included the discriminative SVM model into our experiments, which is, to our knowledge, currently the best performing classification model. The selected models are described in Section 1.

Since Czech is a morphologically rich language, i.e. its words take many different forms, we decided to inspect the impact of different feature selection techniques and vocabulary reduction methods together with lemmatization on the classification perfor-

mance of the models. More details about the considered preprocessing techniques are in Section 3 and 5.

The combination of datasets, feature selection techniques, models and different parameters led to a vast number of experiments, which consumed a large number of computations resulting in even larger number of results. For the lack of space we limited the size of the reported results. Nevertheless, we show and discuss the main outcomes, with a small possibility of omitting a few details here and there.

Finally, we provide the results of the found optimal combinations of parameters and preprocessing steps over models and datasets. We conclude with providing recommendations that turns out to be valid across the different corpora, languages, and models.

1. Classification Models

This section provides a brief description of classification models and various transformations of a the simple vector space representation of documents.

1.1. Multinomial Naive Bayes Model

Multinomial Naive Bayes (MNB) is one of the most common text classification models. The model assigns probability to a document represented as a vector of word counts $\boldsymbol{x} = (x_1, x_2, \ldots, x_{\mathcal{V}})$ under the document category c according to

$$p(\boldsymbol{x}|\boldsymbol{\theta}_c) = \frac{\left(\sum_{w=1}^{\mathcal{V}} x_w\right)!}{\prod_{w=1}^{\mathcal{V}} x_w!} \prod_{w=1}^{\mathcal{V}} \theta_{cw}^{x_w} \tag{1}$$

where $\mathcal{V}$ is the size of the vocabulary and $\boldsymbol{\theta}_c$ are the word emission probabilities which can be estimated using maximum likelihood [7]. A test document $\boldsymbol{x}$ is assigned to the category c with the highest probability

$$p(c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta}_c)p(c)}{\sum_{c=1}^{\mathcal{C}} p(\boldsymbol{x}|\boldsymbol{\theta}_c)p(c)} \tag{2}$$

where the *a priori* probability $p(c)$ are commonly estimated from the training set using maximum likelihood.

To fight against the tendency of MNB to overestimate emission probabilities because of the violation of the independence assumption, a non-Bayesian alternative of the MNB model called *(Transformed) Weighted-normalized Multinomial Bayes* [7] was devised. We used both these models in our experiments to assess the relative improvement of the latter over the former.

1.2. Dirichlet Compound Multinomial Model

Dirichlet Compound Multinomial Model (DCM) is a two-level hierarchical Bayesian model that can be viewed as an extension of the MNB model and as such, it can be understood as bag-of-bag-of-words [4]. The advantage of DCM model over MNB model is that it accounts for word *burstiness*. The word burstiness refers to phenomenon which

describes the natural tendency of words to appear in documents multiple times. The a priori probability of a word appearing in a document may be quite low, but once the word appears its probability of appearing again is much higher (less surprising). The strong independence assumptions of MNB model can never capture this behaviour.

The model assigns probability to a document represented as a vector of word counts $\boldsymbol{x} = (x_1, x_2, \ldots, x_{\mathcal{V}})$ under the category c according to

$$p(\boldsymbol{x}|\boldsymbol{\alpha}_c) = \frac{\left(\sum_{w=1}^{\mathcal{V}} x_w\right)!}{\prod_{w=1}^{\mathcal{V}} x_w!} \frac{\Gamma(\sum_{w=1}^{\mathcal{V}} \alpha_w)}{\Gamma(\sum_{w=1}^{\mathcal{V}} x_w + \alpha_w)} \prod_{w=1}^{\mathcal{V}} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} \tag{3}$$

where $\mathcal{V}$ is the vocabulary size and Γ is the gamma function (see [4] for the derivation). It is obvious on the first glance that calculating the probability of a document under DCM model (Eq. 3) is much more complicated in comparison with MNB model (Eq. 1).

Sadly, no closed-form solution for the maximum likelihood estimate of the $\boldsymbol{\alpha}_c$ exists, but there are plenty of iterative gradient ascent optimization methods [6], from which the method we used to train the models in our experiments is a fixed-point iteration.

The classification of a test document is done similarly as in case of the MNB model using the class membership probability, see Eq. (2).

1.3. Complementary MNB and DCM Models

Complementary modeling with MNB was introduced to deal with corpora that contain skewed document classes [7], and it showed promising results for DCM models as well [4]. We refer to the complement alternatives of the two models as CNB and CDCM.

In regular versions of MNB and DCM models the parameters are estimated from documents of particular category c, on the contrary, the CNB and CDCM models estimate the parameters from the documents which belong to all the categories *except* c. For the explicit formulas on parameter estimation and prediction see [7,4].

1.4. Text Document Representations

To improve the classification performance of a multinomial models several heuristics based on transformation of word count vectors have been proposed [7]. We have implemented 15 different transformations which altogether with the original vector make 16 different document vector representations.

Consider a corpora containing $\mathcal{D}$ documents, represented as count vectors $\boldsymbol{x}_d = (x_{d1}, x_{d2}, \ldots, x_{d\mathcal{V}})$, we define the transformations as follows

$$x^1_{dw} = x_{dw}$$

$$x^2_{dw} = \frac{x_{dw}}{\max'_w x_{dw'}} \qquad x^{i+4}_{dw} = x^i_{dw} \log \frac{\mathcal{D}}{\sum_{d=1}^{\mathcal{D}} \delta_{dw}}, \quad i = 1, \ldots, 4$$

$$x^3_{dw} = \log(1 + x^1_{dw}) \qquad x^{i+8}_{dw} = \frac{x^i_{dw}}{\sqrt{\sum_{w=1}^{\mathcal{V}} (x^i_{dw})^2}}, \quad i = 1, \ldots, 8$$

$$x^4_{dw} = \log(1 + x^2_{dw})$$

where δ_{dw} is 1 iff word w occurs in document d. Note that the document representations $x^5, \ldots, x^{16}$ are referred to as *tf-idf* counts in Information Retrieval community.

2. Datasets

To assess the quality of the models across languages we used our Czech corpora together with commonly used corpora for text classification in English. There is to our knowledge no evidence that the performance of any of the considered model is language dependent, but to make sure we can spot this (theoretical) dependency, we selected the English datasets carefully to resemble our Czech datasets in the number of categories, content, and the size.

2.1. English Datasets

The 4 Universities dataset[1] (denoted as `webkb`) contains web pages collected at computer science departments of four universities. The 8,282 pages were manually classified into the 7 categories from which we used only 4 similarly to [3], namely `student`, `faculty`, `course`, and `project`, resulting in 4200 documents.

The 20 Newsgroups[2] (denoted as `20news`) is a popular benchmarking collection for document classification models. It consists of approximately 20k newsgroup documents, partitioned evenly across 20 different categories.

The industry sector dataset[3] (denoted as `sector`) is another popular document collection for benchmarking of text classification algorithms. The data set comprises of 95470 corporate web pages partitioned into 104 categories.

2.2. Czech Datasets

We have 2 collections of Czech web pages. The web pages from both of these collections are divided into 36 categories. We received these datasets from Czech company Seznam.cz, the 36 categories correspond to web page categories used in their Sklik.cz pay per click system. We refer to these collection as `seznam1k` and `seznam11k` where the former contains 1818 documents and the latter contains 11114 documents.

3. Dataset Preprocessing

This section summarizes the basic preprocessing steps, which were applied to datasets prior to the application of feature selection and subsequent training of the models. Note that before any of these techniques is applied, the datasets are rid of HTML tags, stopwords, numbers, special characters, and whole text is put to lowercase.

[1] Available at: `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz`

[2] Available at: `http://qwone.com/~jason/20Newsgroups/20news-18828.tar.gz`

[3] Available at: `http://people.cs.umass.edu/~mccallum/data/sector.tar.gz`

3.1. Vocabulary Pruning

A simple technique relying on the assumption that very rare and very common words are not likely to be relevant because among the former, technical terms and typos prevail in terms of word occurrences, on the other hand in the latter, words like "the", "of", "and" are the most common (in English) which are not very useful terms for classification.

Final step in preprocessing of text documents is to count the document frequency of each word, i.e. in how many documents the word appears. If the frequency is below a chosen absolute (lower) threshold (e.g. 10 documents) or more then a chosen relative (upper) threshold (e.g. 50% of documents) the word is discarded. The words that are shorter than 3 or longer than 20 characters are being discarded as well.

3.2. Lemmatization

Our target language is Czech, which is morphologically more complex than English. It is common for many Czech words (especially for nouns, verbs, and adjectives) to occur in many distinct forms, so-called inflections, therefore vector representations of the documents are usually very long and sparse even after pruning of the vocabulary. The process of lemmatization groups together different inflected forms of a word to a single instance (lemma) resulting in reduction of the dimensionality of the document vectors.

We used the Common Part-of-speech Tagger (COMPOST)[4] for lemmatization of Czech and WordNetLemmatizer from NLTK[5] to lemmatize English. Apart from the lemma we also used the information from the tagger to extract only nouns, adjectives, and verbs, to further reduce the dimensionality of the document vectors.

3.3. Database Postprocessing Statistics

The effect of the described preprocessing steps on the datasets are summarized in Table 1 in columns ADL, AUWD, and AVS. The statistics were calculated from 10 random samples from the datasets using stratified sampling with appropriate training set sizes, these are the statistical properties of the datasets seen by the different models during training. The training set sizes of the known datasets where chosen according to previous experiments made by others [4,7].

4. Feature Selection Methods

The classification performance of the models can be hugely influenced by the selection of words in the vocabulary. We searched for and decided to use two common feature (word) selection techniques: *information gain* (IG), *chi-square statistic* (χ^2) ([5,8,10]), that works well and one non-standard feature selection method known as *within class popularity* (WCP) reported to have a big improvement gap over the previous methods [9]. The impact of these techniques on the classification precision of the models is being part of our examination in the experiments.

[4]The Common POS Tagger - COMPOST has been developed by the Institute of Formal and Applied Linguistics, http://ufal.mff.cuni.cz/.

[5]NLTK Homepage: `http://www.nltk.org/`

Table 1. Database Postprocessing Statistics. Legend: Number of documents (ND), Minimum category size (mCS), Maximum category size (MCS), Average document length (ADL), average number of unique words per document (AUWD), average vocabulary size (AVS), number of categories (NC), and training set size (TSS). The 3 values reported in the columns ADL, AUWD, and AVS, are the average numbers of words in the vocabulary after: no processing, vocabulary pruning, and vocabulary pruning together with lemmatization.

Dataset	ND	mCS	MCS	ADL	AUWD	AVS	NC	TSS
20news	15076	503	800	279 112 111	142 76 74	93412 15652 14474	20	80%
sector	4795	14	53	352 183 182	155 99 97	48211 11388 10615	104	50%
webkb	2940	353	1149	273 129 130	144 82 82	33602 6864 6542	4	70%
seznam1k	916	2	56	733 590 567	375 345 297	87165 91115 59687	36	50%
seznam11k	8916	133	353	863 560 575	426 334 305	100000 41968 27702	36	80%

Note that the former two methods compute local word importances, i.e. how the word is "important" in the context of the document category, where on the contrary WCP computes the word importance globally, i.e. how the word is important in the context of the whole corpus. In order to assess the global importance for the two methods, we adopted the best performing approaches from [8], which are for IG and χ^2 the sum and a maximum over the local importances, respectively.

5. Experiments Description

The goal of the first experiments was to compare the performance of all the models trained on every document vector representation mentioned in Section 1. Our primary goal was to find out how the performance of the best classification models (found in training) are susceptible to different corpora preprocessing steps. We trained the models on corpora with pruned vocabulary (1), lemmatized pruned vocabulary (2), pruned vocabulary after feature selection (3), and lemmatized pruned vocabulary after feature selection (4), where the numbers in brackets are used as cross-reference into Figure 1 summarizing the results. When feature selection methods were applied, the words in the vocabulary were initially order by their importances and the final vocabulary was built out of only 10%, 20%, ... up to 90% of the most important words.

In our next experiments we were interested in finding what is the (relative) minimum number of words that can be selected from already pruned and lemmatized vocabulary using the feature selection methods without having substantial deteriorating effect on the performance of the classifiers. The results of these experiments are summarized in Figure 2.

In the both experiment setups we also compared the classifiers against (linear) Support Vector Machines in one-vs-all regime [2], which is to our knowledge considered the state-of-the-art text classifier.

We have implemented the classification models ourselves in Python except for SVM, in which case we used the open-source machine learning library for Python *scikit-learn*.

6. Classification Results

The performance of the models is measured using micro-averaged precision [1], defined as $\frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C} TP_c+FP_c}$ where TP_c and FP_c is the number of documents correctly classified into category c and the number of documents incorrectly classified into category c.

We have chosen to optimize precision solely for the purpose of the target domain, which is categorization of web pages for advertising, where the cost of false positives, manifesting in displaying adverts not corresponding with the web page content or worse, being inappropriate for it, can be relatively high.

The results in Table 2, Figure 1, and Figure 2 are calculated from the averages of the results from the experiments (see Section 5) run repeatedly over 5 random splits of the datasets on training and hold-out sets (for training set sizes see Table 1). We took considerable care to construct vocabularies only from the training splits in effort not to leak any information from the hold-out set into training. Also MNB and CNB models were validated with different word smoothing factors [7] and SVM with different penalty factors.

Figure 1 illustrates not only how the best combination of the model together with a corpora transformation performs (the peaks of the bars) but also what is its performance gap from the *baseline* version of the classifier (shaded bar). Note that we omitted the results for DCM and CDCM in case of reduced vocabularies. These models are naturally suited for bigger vocabularies hence their results in this setup would be inadequate and meaningless.

Figure 1 and Table 2 which provide numerical values for the best performances of the models without feature selection (left side) and the best and worst (in brackets) performances of the models per feature selector (right side). It can be concluded that the reduction of vocabularies has not a substantial, yet not insignificant ($\leq 2\%$) impact on the performances on `webkb`, `seznam1k`, and `seznam11k` datasets, but on the contrary big vocabulary reduction in case of `20news` and `sector` leads to great loss in precision (up to 6% and 13%, respectively, on average across classifiers). We think that the reason for this is given by the textual content and diversity of the topics, it is certainly not by the external characteristics of the datasets, since the two are "complementary" in terms of the basic statistics (`sector` has fewer documents and a lot of categories with skewed distribution, on the contrary `20news` has more documents, fewer categories that are uniformly distributed, both have vocabularies of comparable sizes).

Figure 2 shows how severe the loss in performance of the models is for different degrees of reduction of the vocabulary. Following the discussion above, we reduced the vocabulary in case of `webkb`, `seznam1k`, and `seznam11k` to 1%, up to 9% to find when the performance start to drop similarly to `20news` and `sector`, and from how it looks we hit just the right range. We found that in datasets of our interest we can afford to lose up to 94% of the vocabulary without paying the price in terms of precision.

We conclude this section by stating that from the results on the different datasets none of the considered models matches the performance of SVM even with all the possible transforms and feature selections. On the other hand DCM and CDCM models performed worse, these models are hard to train and according to our result does not prove useful for a simple categorization task as ours. But considering the computational demands for training and demonstrated robustness, we conclude that the model of our choice is from the category of Multinomial Naive Bayes models.

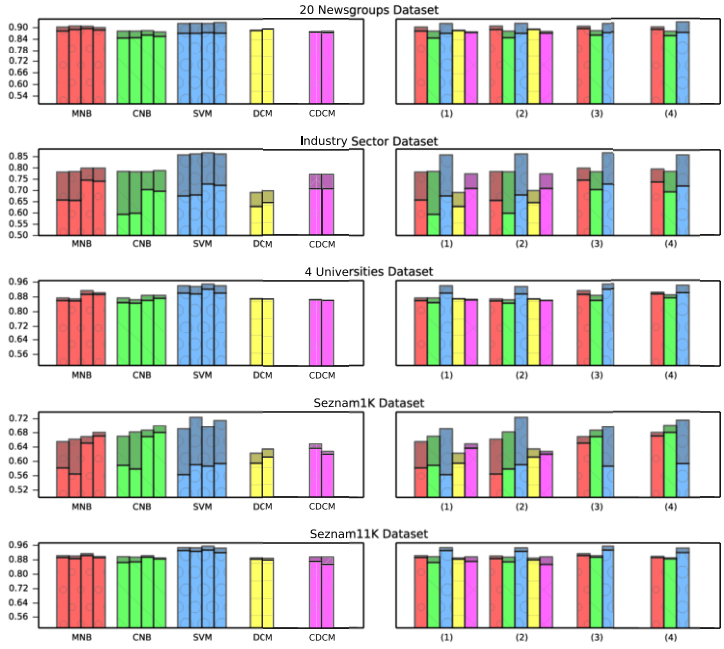

Figure 1. Impact of different preprocessing steps on the precision for individual classification models (*left*) and across different classification models (*right*).

Finally, since the results of the different classification models over different datasets and languages exhibit the same behaviour we conclude that we found no evidence that the performance of the models should be language dependent in case of English and Czech.

7. Conclusion

We presented results of an extensive empirical study of known Bayesian models and enhanced non-Bayesian alternatives to Multinomial Bayes models for text categorization. The study shows what is the best achievable performance for web documents classified to a given number of categories. The optimum combination of the model, the feature extraction method and preprocessing steps can be read out of the tables and graphs referred from Section 6.

We found that on two out of five datasets, among which is the Czech dataset of our interest, we can afford to reduce the vocabulary size approximately to 10% of the total without any classification performance degradation. This property holds across all types of models.

We also proved, that the models perform similarly across two different languages, English and Czech. Despite the fact that Czech is especially difficult and morphologically rich language, we found that lemmatization has no significant influence on classification precision of the considered models, but in the end it does not hurt and it can substantially reduce the vocabulary size which makes the computation during training and classification much faster.

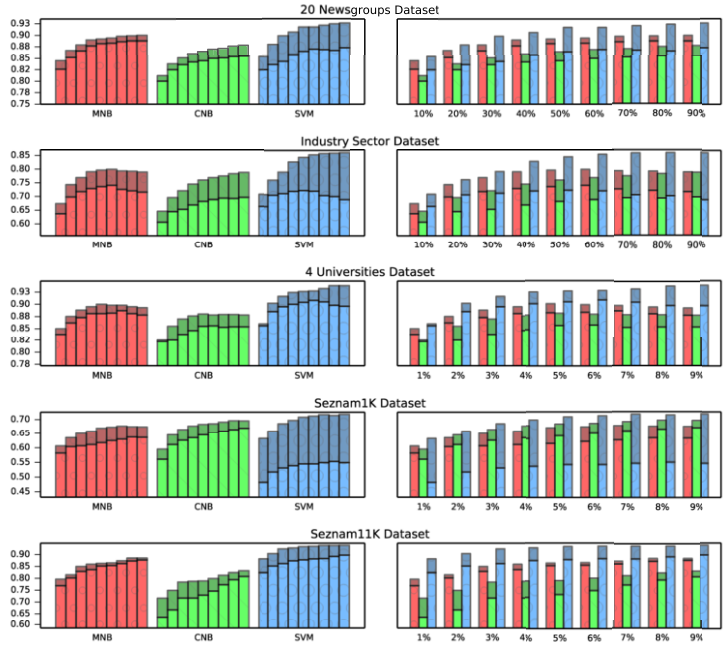

Figure 2. Impact of vocabulary reduction by feature selection methods on the precision for individual classification models (*left*) and across different classification models (*right*) for vocabulary reduced to 1%/10% up to 9%/90%.

Table 2. The maximum value of precision across different classification models and different feature selectors.

20 Newsgroups Dataset

MNB	CNB	SVM	DCM	CDCM	FS	MNB	CNB	SVM
0.91	0.88	0.92	0.89	0.88	IG	0.90 (0.83)	0.88 (0.80)	0.92 (0.84)
					CHI	0.90 (0.84)	0.89 (0.81)	0.93 (0.85)
					WCP	0.90 (0.85)	0.88 (0.81)	0.92 (0.86)

Industry Sector Dataset

MNB	CNB	SVM	DCM	CDCM	FS	MNB	CNB	SVM
0.78	0.78	0.86	0.70	0.77	IG	0.78 (0.60)	0.78 (0.52)	0.86 (0.71)
					CHI	0.80 (0.68)	0.78 (0.65)	0.86 (0.69)
					WCP	0.76 (0.60)	0.78 (0.51)	0.86 (0.70)

4 Universities Dataset

MNB	CNB	SVM	DCM	CDCM	FS	MNB	CNB	SVM
0.87	0.86	0.94	0.87	0.86	IG	0.89 (0.88)	0.88 (0.87)	0.94 (0.94)
					CHI	0.90 (0.88)	0.88 (0.87)	0.94 (0.94)
					WCP	0.90 (0.88)	0.87 (0.86)	0.94 (0.90)

Seznam1K Dataset

MNB	CNB	SVM	DCM	CDCM	FS	MNB	CNB	SVM
0.66	0.68	0.72	0.64	0.63	IG	0.68 (0.67)	0.70 (0.68)	0.72 (0.71)
					CHI	0.69 (0.65)	0.70 (0.65)	0.72 (0.65)
					WCP	0.68 (0.67)	0.69 (0.68)	0.72 (0.71)

Seznam11K Dataset

MNB	CNB	SVM	DCM	CDCM	FS	MNB	CNB	SVM
0.90	0.90	0.95	0.89	0.90	IG	0.89 (0.88)	0.90 (0.84)	0.95 (0.94)
					CHI	0.89 (0.89)	0.90 (0.82)	0.95 (0.93)
					WCP	0.89 (0.89)	0.90 (0.85)	0.95 (0.93)

Furthermore our results show that none of the feature selection techniques considered in this study is bringing significant improvement. Also, based on our empirical results, we can surely recommend the document vector representation $\boldsymbol{x^{15}}$ (see Section 1.4), which is the only document vector representation that led most often (almost all the time) to improvement in the performance of the models.

To conclude, we came to similar result as previous studies that SVM is consistently delivering the top performance. It is certainly the best choice when the training computational requirements are not of a concern, training SVM model in comparison with training of MNB model, which is as easy as simple counting, is a completely different (quadratic optimization) story. Yet the runtime number of operations for all considered models is linearly dependent on the size of the vocabulary and the number of categories.

Finally, note that the best MNB is performing almost on par with SVM, which means in practice, when the best classification performance is not necessary, picking up the right configuration for MNB is probably the right way to go.

Acknowledgements

This work was supported by the Decision Making and Control for Manufacturing III, Research programme MSM 6840770038, funded by the Czech Ministry of Education.

References

[1] George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, March 2003.

[2] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK, 1998. Springer-Verlag.

[3] Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In *Proceedings of the 17th Australian Joint Conference on Advances in Artificial Intelligence*, AI'04, pages 488–499, Berlin, Heidelberg, 2004. Springer-Verlag.

[4] Rasmus E. Madsen, David Kauchak, and Charles Elkan. Modeling word burstiness using the dirichlet distribution. In *In Proceedings of the 22nd international conference on Machine learning, volume 119 of ACM International Conference Proceeding Series*, pages 545–552, 2005.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[6] Thomas P. Minka. Estimating a dirichlet distribution. Technical report, 2000.

[7] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623, 2003.

[8] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

[9] Sanasam Ranbir Singh, Hema A. Murthy, and Timothy A. Gonsalves. Feature selection for text classification based on gini coefficient of inequality. In Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao, editors, *FSDM*, volume 10 of *JMLR Proceedings*, pages 76–85. JMLR.org, 2010.

[10] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.