Evaluation of Classification Models for Language Processing

Zeynep Hilal Kilimci
Computer Engineering Department
Dogus University
Istanbul, Turkey
hkilimci@dogus.edu.tr

Murat Can Ganiz
Computer Engineering Department
Dogus University
Istanbul, Turkey
mcganiz@dogus.edu.tr

***Abstract*—Naïve Bayes is a commonly used algorithm in text categorization because of its easy implementation and low complexity. Naïve Bayes has mainly two event models used for text categorization which are multivariate Bernoulli and multinomial models. A very large number of studies choose multinomial model and Laplace smoothing just based on the assumption that it performs better than multivariate model under almost any conditions. This study aims to shed some light into this widely adopted assumption by analyzing Naïve Bayes event models and smoothing methods from a different perspective. To clarify the difference between events models of Naïve Bayes, their classification performance are compared on different languages -English and Turkish- datasets. Results of our extensive experiments demonstrate that superior performance of multinomial model does not observed all the time. On the other hand, multivariate Bernoulli model can perform well when combined with an appropriate smoothing method under different training data size conditions.**

Keywords—Naïve Bayes; event models; smoothing methods; text categorization; language processing.

I. INTRODUCTION

Simplification of Bayes' theorem by independence assumption provides a basic classifier called Naïve Bayes. Bayes assumption says that all features are independently discrete of each other within each class. Although this assumption usually does not hold in practice, Naive Bayes almost always has a significant performance for text categorization [1].

Multivariate Bernoulli (MVNB) and multinomial event models (MNB) are used in text classification domain to indicate text documents. MVNB and MNB are called respectively binary independence model and unigram language model in literature. In MVNB, terms being present and absent have an important role in representation of binary vector. In other words, whether the term occurs in the document, it is placed as one within binary vector and otherwise zero. Frequencies of terms in documents are not captured by this model. In contrast to multivariate Bernoulli model, documents are denoted by integer term counts vector in multinomial model. On account of multinomial distribution of each class, this model is called multinomial event model [1].

Majority of the studies regarding Naïve Bayes text categorization employ multinomial model depended on the suggestion of the well-known research [1]. In this influential paper, authors empirically compare two event models by varying vocabulary size on five different datasets including 20 Newsgroups and WebKB. They conclude that multivariate Bernoulli works weakly on large vocabularies and multinomial event model is a better fit for this type of challenging classification problems. The experiments are designed to observe the performance of event models under different feature sizes by using average mutual information as feature selection method. In this research, we concentrate on the performance of one of the most commonly used algorithm, Naïve Bayes, under sparse training data conditions. In our opinion, smoothing methods in Naïve Bayes are of critical importance especially under these conditions since majority of the events or parameters could not be inferred from training data. Smoothing methods distribute some of the probability mass to these previously unseen events. We experiment with different Naïve Bayes event models combined with a wide range of smoothing methods.

Majority of studies which employs Naïve Bayes algorithm for text classification choose multinomial model and Laplace smoothing by default. This is mainly based on the landmark study of unlike multivariate Bernoulli model; multinomial model captures the order of the terms and frequency information of them. Along with, previous studies on event models [2-8] show experimentally that multinomial model outperforms multivariate Bernoulli model. Surprisingly several studies that uses Naïve Bayes in semi-supervised settings where labeled training set sizes are usually very small follows the same pattern [9].

However, this widely adopted assumption requires more investigation. From this point of view, the main motivation of our work is investigating and analyzing these two approaches from a different perspective. Extensively experiments are carried out with various training set sizes instead of vocabulary size to better reflect real world settings. Additionally, we utilize several different smoothing methods with these event models. Due to scarcity in training data, unseen terms (terms that do not exist in training data) can cause zero probability problems during classification. To avoid this problem, some of the probability mass from existing terms is distributed over unseen terms.

We employ four different datasets Turkish and English which are frequently used in related studies. Moreover, the vocabulary size of these datasets is sufficiently large. Provided that the training set size is fixed as 80% on our datasets like the well-known study [1], classification performance results can be comparable for large vocabulary sizes. After analyzing experimental results, we observe that multivariate Bernoulli model can significantly outperforms when combined with an appropriate smoothing method at any training set size as well.

Furthermore, this combination can even outperform the Support Vector Machines (SVM) with linear kernel under certain conditions. We also realized that multivariate Bernoulli event model demonstrates the best performance while numbers of documents are evenly or nearly distributed among categories.

II. Related Work

There are several studies on two generative models of Naïve Bayes (NB) classification. [1] analyzed the performances of multivariate Bernoulli and multinomial model on five different datasets. These datasets were collected from various web pages such as Yahoo science, companies, newsgroups, university computer science departments and Reuters. It is clearly stated that giving smaller vocabulary sizes has remarkable performance on the multivariate Bernoulli event model. Looking at it the other way, multinomial model usually notably acts if large vocabulary sizes are given. Their results indicate that the multivariate Bernoulli model shows better performance at smaller vocabulary sizes, whereas the multinomial model usually acts well out with large vocabulary sizes.

There are several studies which try to improve upon NB based on multinomial event model. In one of these [7], empirically compares the performance of four NB event models for text classification. These models are multivariate Bernoulli model and variants of multinomial model including a Poisson model, a negative binomial and models explicitly incorporate document length. Their results on several datasets including 20 Newsgroups suggest that multinomial model often outperforms the others.

The other study [10] proposes empirical heuristic to improve poor results in the text classification domain. They also utilize Poisson NB text classification model with weight enhancing method. This model assumes that a document is composed by a multivariate Poisson model. Unlike parameter estimation of traditional classifiers, their new model suggests per-document term frequency normalization in order to estimate Poisson parameter. In other words, they propose to develop an alternative text classification model to traditional NB classifier by enhancing classification weights with multivariate Poisson model. Their experiment results on both datasets, including 20 newsgroups and Reuters, present that the new model outperforms the traditional multinomial classifiers.

Another study [4] also bases on multinomial model by referencing to the well-known study [1] and by stating that multinomial NB shows improved performance compared to multivariate Bernoulli model due to the incorporation of frequency information. They provide simple yet effective modifications, including various normalization techniques, to multinomial NB to improve the accuracy by using three well-known datasets covering 20 Newsgroups, Industry sector and Reuters. These modifications address especially skewed data bias.

In an interesting study [2] on NB event models, authors argue that including term frequency information is not the fundamental factor in order to distinguish the multinomial NB from multivariate Bernoulli NB whereas it is oppositely claimed by previous studies [1, 7]. Furthermore, authors investigate that negative evidence is the main reason for differences between the two probabilistic models' performances. This suggests that the better performance of multinomial model compared to multivariate Bernoulli model may not stem from extra information on term frequencies. They carry out classification experiments on three accessible datasets in public including 20 Newsgroups, WebKB and Ling-spam datasets. Experiment results show that the multinomial NB classification accuracy exceeds the multivariate Bernoulli model performance because of negative evidence from documents.

The similar study [3] also investigates the classification performance of different versions of NB classifier in a spam filtering context. These are multivariate Bernoulli NB, Multinomial NB with term frequency attributes, Multinomial NB with binary attributes, Multivariate Gauss NB, Flexible NB. In order to measure performance of NB classifiers, authors evaluate their experiments on six datasets, collectively called Enron-Spam. Based on the study [2], they conclude that multinomial event model with binary attributes has outstanding performance among NB versions.

Majority of the studies on NB text categorization including the ones mentioned above employ multinomial event model and Laplace smoothing. There are a few studies that aim to measure the performance of different smoothing methods with event models. For instance, one of these studies [5], uses conventional multinomial model and reversed multinomial model, covering length normalization, with several smoothing methods which base from modeling of statistical language domain and usually employed with n-gram language models. These cover absolute discounting combined with unigram backing-off and combination of absolute discounting, unigram interpolation and Laplace smoothing. They specify that absolute discounting with unigram interpolation surpasses Laplace smoothing. They also bear in mind document length normalization provides getting better results similar to previous studies [4, 7] on 20 Newsgroups dataset.

In a similar study [11] with shared authors they conduct experiments on 20 Newsgroups dataset using NB with unigram interpolation smoothing and show error rate with increasing vocabulary size. Results show a slightly better error rate for unigram interpolation smoothing. Another study [8] follows the same tradition and bases their study on only multinomial NB model. Instead of using unigram term probabilities, they augment NB by using n-grams and consequently sophisticated smoothing methods from language modeling field containing Absolute smoothing [12], Good-Turing smoothing [13] and Witten-Bell smoothing [14]. They also use 20 Newsgroups

dataset by setting training set size to 80%. They achieve best accuracy value by using bigram models with linear discounting. In our study, we use the similar smoothing methods but only for unigrams.

The interesting study [15] on text classification states that multinomial model just about regularly better than Bernoulli model by following the same tradition and as a result focuses solely on multinomial model. They introduce smoothing methods from language modeling area involving absolute smoothing, Good-Turing smoothing, linear smoothing and Witten-Bell smoothing [8]. However, they conduct experiments only on Chinese documents.

The recent study [16] applies several smoothing methods to multinomial model for short texts on Yahoo web scope dataset comprising 3.9 million questions. These methods involve absolute discounting (AB), Jelinek-Mercer (JM) smoothing, Dirichlet smoothing and two - stage (TS) smoothing. They generate two-stage smoothing thereby blending Jelinek-Mercer and Dirichlet smoothing. They also apply different preprocessing methods including stop words removal and stemming on their datasets. Experimental results indicate that smoothing methods significantly raise the performance of multinomial model. They observe that all smoothing methods raise the classification performance. Especially, AB and TS contribute to enhance performance of Naïve Bayes among the four smoothing methods by representing the best performance at different scales of training data.

Support Vector Machines (SVM) is a popular machine learning classifier. This method seeks to obtain a decision border that splits points into two classes by making bigger boundary [17, 18]. SVM aims to project data points into a higher dimensional space. Since data points become linearly distinguishable thereby applying kernel methods. SVM algorithm can be combined with several kernel techniques. Among these, linear kernel has almost always the best classification performance on text classification field [19]. We mention SVM accuracy results in our extensive experiments to compare performances of the others.

III. Naïve Bayes Event Models

In order to demonstrate the text documents, there are two event models combined with Naïve Bayes (NB) for text classification. First is multivariate Bernoulli event model is called binary independence NB model because of representation of terms. This representation indicates that occurrence of the terms in binary vector is referred as one, otherwise zero. This model does not utilize term frequency information and thus usually considered as an inferior model.

In Eq. (1), occurrence of term t in document i is indicated by B_{it} which can be either 1 or 0. $|D|$ indicates the number of labeled training documents. Formula is given using Laplace smoothing. $P(c_j|d_i)$ is 1 if document i is in class j. Finally, the probability of term w_t in class c_j is [1]:

$$P(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} B_{it} P(c_j \mid d_i)}{2 + \sum_{i=1}^{|D|} P(c_j \mid d_i)} \quad (1)$$

The other NB event model is multinomial model which regardless of the terms' position can utilize term frequencies. In contrast to multivariate Bernoulli model, documents are denoted by a vector of term counts. The class conditional probability of the term w_t in class c_j is given by multinomial distribution (using Laplace smoothing) [1]:

$$P(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} N_{it} P(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{is} P(c_j \mid d_i)} \quad (2)$$

where $|V|$ is vocabulary (total number of terms), N_{it} is the number of the count of times term w_t occurs in document d_i.

IV. Smoothing Methods

Owing to scarcity in training data, unseen terms in the document can cause "zero probability problems" in Naïve Bayes assumption. To avoid this, probability mass of existing terms is redistributed and unseen terms take extra probability mass. As a result, this process called smoothing.

A. Laplace Smoothing

Laplace smoothing is one of the most common and oldest smoothing method employed in practice. This method is referred as adding one owing to adding a constant to every term count. Though Laplace smoothing is one of influential method in order to prevent zero probability problem, Laplace smoothing is to assign too much probability mass to unseen terms for sparse sets of data over large vocabularies. Probability of term given class with Laplace for multinomial Naïve Bayes [20]:

$$\varphi_{c,t} = \frac{1 + N(c_t, D)}{|V| + N(c, D)} \quad (3)$$

Probability of term given class with Laplace for multivariate Bernoulli Naïve Bayes:

$$\varphi_{c,t} = \frac{1 + N(c_t, D)}{2 + N(c, D)} \quad (4)$$

where $\varphi_{c,t}$ is the probability of term given class. $N(c_t, D)$ represents number of documents in class c that contain term t and $N(c, D)$ denotes number of documents in class c.

B. Good Turing Smoothing

The Good Turing smoothing method assigns the probabilities for observed terms that are consistent with the total probability assigned to the unseen terms. This is established on a theorem about the probability of an existing

term is replaced with a smaller probability. The total number of the smaller probabilities is subtracted from 1 and this difference is redistributed equally among the unseen terms [21]. In this method, r symbolizes term frequency, r^* is called an adjusted frequency which is the total probability for all terms adds to one. In other words, the probability of the terms, which were seen to be less than one, must be reduced to be consistent with non-zero probabilities for unseen terms, N_r is the number of terms with a frequency of r.

$$r^* = (r+1) \times \frac{N_{r+1}}{N_r} \tag{5}$$

$$P(w_t \mid c_j) = \begin{cases} \frac{r^*}{T} & 0 < r \le k \\ \frac{r}{T} & r > 0 \\ \frac{N_1}{N_0 \times T} & r = 0 \end{cases} \tag{6}$$

where T is the total number of training documents that belongs to class j. N_0 refers to the frequency of zero term counts in class j, N_1 is the frequency of one time term counts in class j and k is some constant. We combine the event models with this smoothing method thereby calculating T. For multivariate Bernoulli model, T is :

$$\mathrm{T} = \sum_{i=1}^{|D|} P(c_j \mid d_i) \tag{7}$$

where is 1 when the document i belongs to class j. For multinomial model, T is :

$$\mathrm{T} = \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{is} P(c_j \mid d_i) \tag{8}$$

where N_{is} is the count of the number of times each term w_s takes place in document d_i.

C. Jelinek- Mercer Smoothing

There are fundamental distinctions between smoothing methods whether the method is interpolated or backed-off. Jelinek-Mercer smoothing method is fall under interpolated models since the maximum estimate is interpolated with the smoothed lower-order distribution [22].

$$P_{ml}(w_t \mid c_j) = \frac{N(c_t, D)}{N(c, D)} \tag{9}$$

where $P_{ml}(w_t|c_j)$ is the probability with maximum likelihood estimate and $P(w_t|D)$ is the maximum likelihood estimation of term t in collection D where D is referred as the total number of documents.

$$P(w_t \mid c_j) = (1-\beta) \times P_{ml}(w_t \mid c_j) + \beta \times P(w_t \mid D) \tag{10}$$

where β can be set to some constant. Due to zero probability problems, we again need to smooth maximum likelihood estimation of the term t in collection D. Both formulas of multivariate Bernoulli and multinomial event models are mentioned before at Laplace smoothing section.

D. Absolute Discounting Smoothing

Absolute discounting method is included backed-off models because of determining the probability estimation with non-zero count by ignoring information from lower-order distributions [23]. The idea behind of this smoothing technique is that the seen terms are gained free probability mass thereby discounting a small pseudo count to every term count [11]. In other words, non-zero frequencies are figured out by a small constant amount δ and the frequency is uniformly redistributed over unseen terms [20].

$$\delta = \frac{N_1}{N_1 + 2N_2} \tag{11}$$

where N_1 is the frequency of the number of one time term w_t occurs in class c_j, and N_2 denotes the frequency of two times term count in class c_j.

$$P(w_t \mid c_j) = \begin{cases} \frac{r-\delta}{N}, & r > 0 \\ \delta \times \frac{\sum_{i>0} N_i}{N_0 \times N}, & r = 0 \end{cases} \tag{12}$$

In Eq. (12), r is term frequency and total number of training documents in class j is denoted by N, N_0 is the frequency of zero term counts and N_i refers to frequency of non - zero term counts in class j. We also distinguish two event models thereby computing N, N_i, and N_0. Both formulas of multivariate Bernoulli and multinomial event models are mentioned before at Good-Turing smoothing section.

E. Linear Discounting Smoothing

In this smoothing method, the non-zero frequencies are processed by a constant less than one, and the remaining probability mass is again distributed among novel terms. That is, this estimator rescales the probability of unseen terms thereby subtracting or multiplying by a small constant instead of zero [24].

$$P_{ml}(w_t \mid c_j) = \frac{N(c_t, D)}{N(c, D)} \tag{13}$$

$$P(w_t \mid c_j) = \begin{cases} (1-\gamma) \times P_{ml}(w_t \mid c_j) & r > 0 \\ \frac{\gamma}{N_0} & r = 0 \end{cases} \tag{14}$$

where $P_{ml}(w_t|c_j)$ is the probability with maximum likelihood estimate, N_0 demonstrates the frequency of zero term counts in

class j and γ can be set to some constant in this smoothing method.

V. Experiment Setup

We use four benchmark datasets with different sizes and properties to investigate performance of the algorithms for language processing. First one is called 20News-18828[1] and it contains less number of documents than the original dataset since identical postings are eliminated. This dataset is divided into twenty different categories. Majority of studies [1, 2, 4, 5, 6, 7, 10, 25, 26, 27] also choose 20 Newsgroups for text classification experiments. Second dataset is the WebKB[2] which contains web pages gathered from computer science departments of various universities. These web pages are composed of seven categories which are student, faculty, staff, course, project, department and other. We analyze reduced category version of WebKB dataset which is experimented in some studies [1, 2, 11]. This dataset is called as WebKB4.

The other datasets consist of Turkish documents. Milliyet9c1k dataset contains texts from Turkish newspaper Milliyet[3]. It includes nine classes and 9000 documents. The classes are café (cafe), dünya (world), ege (region), ekonomi (economy), güncel (current), siyaset (politics), spor (sports), Türkiye (Turkey), yaşam (life). Another Turkish dataset is comprised of Turkish newspaper Hürriyet[4]. Hurriyet6c1k similarly covers 1000 documents per each class and six classes. These are dünya (world), ekonomi (economy), güncel (current), spor (sports), siyaset (politics), yaşam (life). Descriptions of the datasets, when no preprocessing is applied, are given in Table I including number of classes (|C|), number of documents (|D|) and the vocabulary size (|V|).

We take into account frequent terms whose document frequency is greater than two. We do not apply any stemming or stop word filtering in order to avoid any bias that can be introduced by stemming algorithms or stop list.

We implement experiments by varying the training set size and making use of following percentages of the data for training and the remain for testing: 1%, 5%, 10%, 30%, 50%, 70%, and 90%. These percentages are demonstrated with "ts" prefix to prevent perplexity with accuracy percentages.

TABLE I. Descriptions of the datasets with no preprocessing

Dataset	|C|	|D|	|V|
20News-18828	20	18828	50570
WebKB4	4	4199	16116
Hurrıyet6c1k	6	6000	18280
Milliyet9c1k	9	9000	63371

[1] http://people.csail.mit.edu/people/jrennie/20Newsgroups

2 http://www.cs.cmu.edu/~textlearning

3 www.milliyet.com.tr

4 www.hurriyet.com.tr

We also conduct experiments using 80% of the data and no document frequency filtering to assess our results with the ones in the literature. We run our experiments by arranging ten random splits for each of the training set percentages above per class. This approach is similar to famous studies [1, 4] where they use 80% training data and 20% for test.

VI. Experiment Results

It is important to notice that we always use the accuracy results to obtain the best carrying out smoothing method for each event model when drawing conclusions. It is usually Laplace (LP) in multinomial Naïve Bayes (MNB) and linear discounting (LD) in multivariate Bernoulli Naïve Bayes (MVNB).

We also present statistical significance test thereby utilizing Student's t-Test particularly when results of different algorithms are close to each other (e.g. about 2–3 % difference). Whether the probability associated with Student's t-Test is lower, we consider the difference is statistically significant if it is arranged as $\alpha = 0.05$ significance level.

We start with the original 20 Newsgroups dataset. The best performing smoothing method for MVNB is LD and LP for MNB (except for ts10, ts5, ts1 on which LD performs better). For all training set sizes, accuracy of MVNB statistically significantly outperforms both MNB and Support Vector Machine (SVM). As can be seen in Figure 1, accuracy difference is 15% between MVNB and MNB at ts10 level and at ts5 level it increases to about 29%.

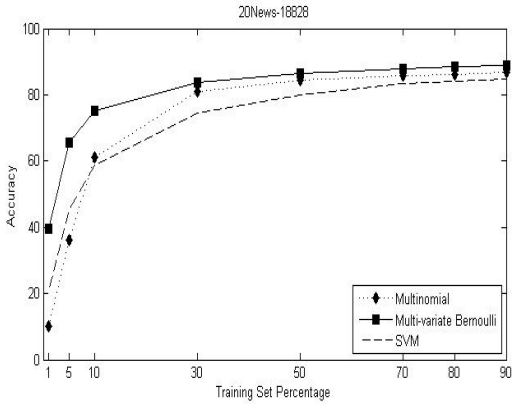

Fig. 1. Accuracy of MNB with LP and MVNB with LD on 20News-18828.

WebKB4 has different properties such as skewed class distribution we observe somewhat different patterns. Similar to the other dataset, in general best performing smoothing methods are LD and LP for MVNB and MNB respectively.

However, accuracy of MVNB is slightly lower (about 1%) than MNB for all training set percentages except ts1. Yet, these small differences are not statistically significant at ts30, ts10 and ts5 levels (probabilities are 0.65, 0.81 and 0.19 respectively). At ts1 MVNB statistically significantly outperforms MVNB by about 6% increase in the accuracy.

SVM has exceptionally good performance in this dataset. SVM surpasses others by at least 4% in all training set percentages except ts1 where MVNB has about 4% higher accuracy. Figure 2 summarizes these results.

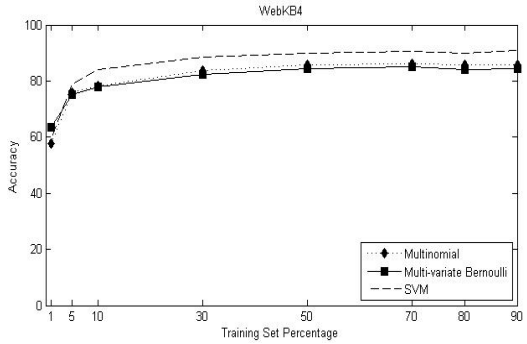

Fig. 2. Accuracy of MNB with LP and MVNB with LD on WebKB4.

In Hurriyet6c1k, for most of cases LD is the best performing smoothing method in MVNB. Furthermore, MVNB with LD outperforms MNB with LP in all training set percentages. Similar to other datasets, in general best performing smoothing method is Laplace for MNB. The pattern can be seen in Figure 3.

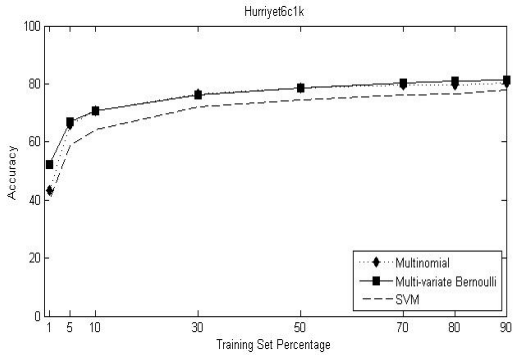

Fig. 3. Accuracy of MNB with LD and MVNB with LP on Hurriyet6c1k.

In Milliyet9c1k, for all training set levels, JM has the best accuracy performance among other smoothing methods in MVNB. Laplace is also the best performing smoothing method for MNB at every training set percentage. Moreover, MVNB surpasses MNB by at 3% in all training set percentages. The difference reaches maximum value, 11%, at ts5 and ts1 levels. SVM has also exceptionally good performance in this dataset. SVM outperforms others by at least 2% in all training set percentages except ts10, ts5, and ts1 levels where MVNB has 3%, 7%, 10% higher accuracy respectively. Figure 4 summarizes these results in detail.

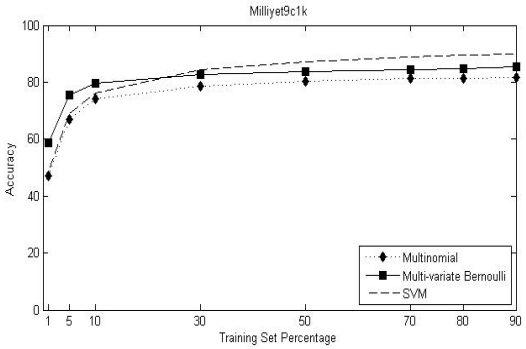

Fig. 4. Accuracy of MNB with LP and MVNB with JM on Milliyet9c1k.

Figure 5 and 6 show the accuracy of different smoothing methods in MNB and MVNB respectively on 20News-18828. Laplace smoothing is the best performing method for MNB. It is followed by LD, and after that by JM in some cases. For MVNB, LD smoothing outperforms the other smoothing methods by a least 4% in all training set percentages and LP smoothing yields the lowest values. Moreover, Laplace smoothing yields by far highest accuracies when combined with MNB for all datasets. LD surpasses the other smoothing methods for 20 News-18828, Hurriyet6c1k, WebKB4. GT and JM methods have the highest results for Milliyet9c1k dataset.

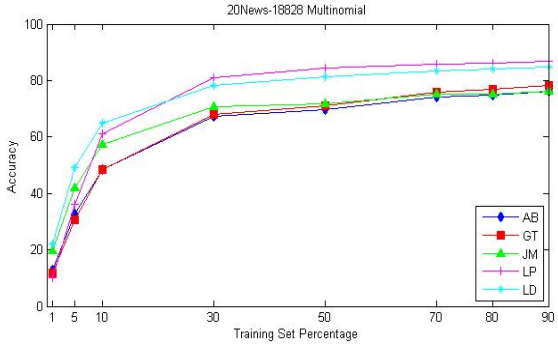

Fig. 5. Accuracy of MNB on 20News-18828.

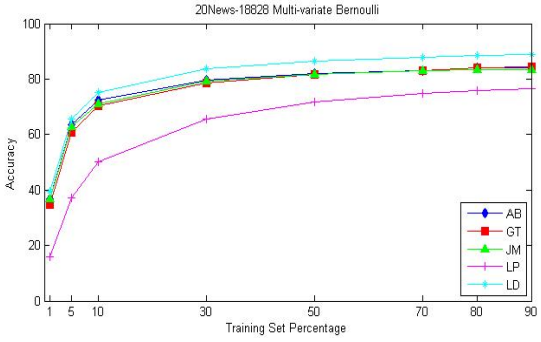

Fig. 6. Accuracy of MVNB on 20News-18828

In WebKB4 similar to the other dataset, in general best performing smoothing methods are LD and LP for MVNB and MNB, respectively. Laplace smoothing outperforms others in all training set percentages by at least 1% and 2% for MNB and MVNB, respectively. Due to skewed class distribution, we observe different patterns than the others. That is, the results of accuracies are close to each other for smoothing methods except LD and LP both MNB and MVNB.

In any case, Laplace smoothing yields the lowest values in MVNB and highest values in MNB. LP and LD yield by far highest accuracies when combined with MNB and MVNB, respectively. For MNB, it is followed by JM and after that GT and AB. LD is again the best performing smoothing method for MVNB and it is followed by AB, GT, JM and after that LP.

In Hurriyet6c1k, starting from 10% to 90% training data, accuracy of LP smoothing exceeds other smoothing methods by approximately at least 2% for MNB. For MVNB, LD is the best performing smoothing method in all training set percentages. Although all smoothing methods are similar to each other, the highest and lowest accuracies are more distinctive here.

In Milliyet9c1k, for all training set levels, JM has the best accuracy performance among other smoothing methods in MVNB. Laplace is also the best performing smoothing method for MNB at every training set percentage. Moreover, LP surpasses the others by at least 2% in all training set percentages except at ts1 level where JM has 5% higher accuracy. Similarly, JM outperforms other smoothing methods by at least 2% in all training set percentages except at ts1 level where GT has 5% higher accuracy in MVNB.

Table I, II and III demonstrate accuracy results for ts80, ts30 and ts5 levels respectively. We also indicate the best performing smoothing method for both of the event models in these tables. Additionally, best results among the NB results are indicated with bold font. If MVNB and MNB results are close to each other (e.g. about 2–3% difference) and t - test results show no difference is not significant then both values are indicated with bold font. Furthermore, we arrange smoothing parameter thereby optimizing their values. 1.00 is the parameter value for LP, AB, GT smoothing methods. Values of JM and LD are optimized as respectively 0.5 and 0.10.

TABLE I. ACCURACIES OF NB EVENT MODELS AT 80% TRAINING SET LEVEL

Dataset	MVNB	MNB	SVM
20News-18828	**88.52±0.42 *LD***	86.26±0.36 *LP*	*84.18±0.79*
WebKB4	**84.10±1.12 *LD***	**85.64±1.17 *LP***	*89.85±0.96*
Hurrıyet6c1k	**81.16±0.96 *LD***	79.78±0.78 *LP*	*76.58±1.31*
Milliyet9c1k	**84.64±0.95 *JM***	81.48±1.03 *LP*	*89.41±0.53*

TABLE II. ACCURACIES OF NB EVENT MODELS AT 30% TRAINING SET LEVEL

Dataset	MVNB	MNB	SVM
20News-18828	**83.60±0.46 *LD***	80.83±0.78 *LP*	*74.40±0.49*
WebKB4	**82.37±1.67 *LD***	**83.89±1.79 *LP***	*88.54±0.86*
Hurrıyet6c1k	**76.24±0.42 *LD***	**76.63±0.56 *LP***	*72.12±0.50*
Milliyet9c1k	**82.55±0.57 *JM***	78.71±0.55 *LP*	*84.32±0.58*

TABLE III. ACCURACIES OF NB EVENT MODELS AT 5% TRAINING SET LEVEL.

Dataset	MVNB	MNB	SVM
20News-18828	**65.55±1.42 *LD***	49.25±7.27 *LD*	*45.46±0.95*
WebKB4	**75.12±1.05 *LD***	**75.97±1.72 *LP***	*78.81±2.21*
Hurrıyet6c1k	**67.73±0.63 *AB***	65.89±1.06 *LP*	*58.81±1.24*
Milliyet9c1k	**75.68±1.12 *JM***	66.99±2.06 *LP*	*68.86±1.03*

The performance improvement of MVNB combined with an appropriate smoothing method over MNB is most noticeable when we have limited amount of labeled training data which usually is the case in real world applications.

VII. DISCUSSION AND CONCLUSION

Superiority of the multinomial event model is a widely accepted assumption in literature as mentioned before. Several studies argue that this is owing to the incorporation of frequency information in multinomial model [4]. By doing so multinomial model can exploit extra information in term frequencies and consequently perform better than multivariate Bernoulli model which uses only if the term occurred in the document or not.

However, more recent studies demonstrate that when all term frequency information is ignored, the multinomial model achieves better performance [2, 3]. This suggests that the better performance of multinomial model compare to multivariate Bernoulli model may not stem from extra information on term frequencies. Besides discussing these assumptions, we also argue that the difference of performance between event models derive from how they calculate the probabilities and how they distribute probability mass to previously unseen events.

The best performing smoothing method for each event model can change by the size of training set which also indicates the sparsity. It is usually LP in MNB and LD in MVNB. In 20 Newsgroups datasets, MNB with LD outperforms LP well when the training set size is lower than 30% as can be seen in Table III at part VI.

Please note that our 20News-18828 accuracy results for MNB in Table I compares to the state of the art results [1, 4] (85% vs. our 86.3%) and (84.8% vs. our 86.3%), respectively below. Our results are also consistent with another studies [7, 11, 27] (86% vs. our 84.7%), (86.8% vs. our 84.7%) and

(84.4% vs. our 84.7%) on 20 Newsgroups dataset, respectively. For MVNB, results of 20 Newsgroups dataset are also comparable with the literature [7]. On the other hand, our SVM accuracy is 84.2% but the well-known study [4] reports 86.2%. This difference may due to the different SVM packages used and the optimization of the parameters (c=10 vs. our c=1). Table I demonstrates results below, briefly.

TABLE I. COMPARISON WITH THE STATE OF THE ART RESULTS

Study	MNB	MNB-our	MVNB-our
[1]	85.0% **(LP)**	86.3% **(LP)**	88.5% **(LD)**
[4]	84.8% **(LP)**	86.3% **(LP)**	88.5% **(LD)**
[7]	86.0% **(LP)**	84.7% **(LP)**	88.5% **(LD)**
[10]	86.8% **(LP)**	84.7% **(LP)**	88.5% **(LD)**
[25]	84.4% **(LP)**	84.7% **(LP)**	88.5% **(LD)**

Results of our extensive experiments demonstrate that multivariate Bernoulli event model can significantly outperform or perform competitively to multinomial event model in several cases when combined with appropriate smoothing method at any training set size. That is, superior performance of the multinomial model does not observed all the time. Not only the multinomial model, but also multivariate Bernoulli model can perform good results. This is especially the case when training data is scarce. Scarcity of labeled data is a common problem in real world settings and the main motivation of semi-supervised learning algorithms. Multivariate Bernoulli event model with appropriate smoothing method can be a better fit for these types of algorithms.

REFERENCES

[1] McCallum A, Nigam KA. Comparison of Event Models for Naive Bayes Text Classification. In: AAAI-98 Workshop on Learning for Text Categorization; 1998; Wisconsin, USA: pp. 41-48.

[2] Schneider KM. On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. In: 4th International Conference on Advances in Natural Language Processing; 2004; Alacant, Spain: pp. 474–485

[3] Metsis V, Androutsopoulos I, Paliouras G. Spam Filtering with Naive Bayes – Which Naive Bayes?. In: 3rd Conference on Email and Anti-Spam; 2006; California, USA.

[4] Rennie JDM, Shih L, Teevan J, Karger DR. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In: 20th International Conference on Machine Learning; 2003; Washington, USA: pp. 616-623.

[5] Juan A, Ney H. Reversing and Smoothing the Multinomial Naive Bayes Text Classifier. In: 2nd International Workshop on Pattern Recognition in Information Systems; 2002; Alacant, Spain: pp. 200-212.

[6] Kolcz A, Yih W. Raising the Baseline for High-Precision Text Classifiers. In: 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2007; California, USA: pp. 400-409.

[7] Eyheramendy S, Lewis DD, Madigan D. On the Naive Bayes Model for Text Categorization. In: 9th International Workshop on Artificial Intelligence and Statistics; 2003; Key West, Florida, USA: pp. 332-339.

[8] Peng F, Schuurmans D, Wang S. Augmenting Naive Bayes Classifiers with Statistical Language Models. Information Retrieval 2004; 7: 317–345.

[9] Nigam K, Mccallum AK, Thrun S, Mitchell T. Text Classification from Labeled and Unlabeled Documents using EM. In: 17th International Conference on Machine Learning; 2000; California, USA: pp. 103-134.

[10] Kim SB, Han KS, Rim HC, Myaeng SH. Some Effective Techniques for Naïve Bayes Text Classification. IEEE Transactions on Knowledge and Data Engineering 2006; 8: 1457-1465.

[11] Vilar D, Ney H, Juan A, Vidal E. Effect of Feature Smoothing Methods in Text Classification Tasks. In: 4th International Workshop Pattern Recognition in Information Systems; 2004; Porto, Portugal: pp. 108-117.

[12] Ney H, Essen U, Kneser R. On structuring probabilistic dependencies in stochastic language Modeling. Computer Speech and Language 1994; 8: 1–28.

[13] Katz SM. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transactions on Acoustics Speech and Signal Processing 1987; 35: 400–401.

[14] Witten I, Bell T. The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. IEEE Transactions on Information Theory 1991; 37: 1085-1094.

[15] He F, Ding X. Improving Naive Bayes Text Classifier using Smoothing Methods. In: 29th European Conference on Information Retrieval Research; 2007; Springer: pp. 703.

[16] Yuan Q, Cong G, Thalmann NM. Enhancing Naïve Bayes with Various Smoothing Methods for Short Text Classification. In: 21st World Wide Web Conference; 2012; Lyon, France.

[17] Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: 10th European Conference on Machine Learning; 1998; Chemnitz, Germany: pp. 137-142.

[18] Burges CJC. A Tutorial on Support Vector Machines for Pattern Recognition. In: 3rd International Conference on Knowledge Discovery and Data Mining; 1998; New York, USA: pp. 121-167.

[19] Yang Y, Liu X. A Re-examination of Text Categorization Methods. In: 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999; Berkeley, CA, USA: pp. 42-49.

[20] Manning, C., Schütze, H. Foundations of Statistical Natural Language Processing. Cambridge, Massachusetts: MIT Press, 1999.

[21] Gale WA. Good-Turing Smoothing without Tears. Journal of Quantitative Linguistics 1995; 2: 217-237.

[22] Chen S. F., Goodman J. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report, Harvard University Center for Research in Computing Technology, Cambridge, USA, 1998.

[23] Chen SF, Rosenfeld R. A Survey of Smoothing Techniques for Maximum Entropy Models. IEEE Transactions on Speech and Audio Processing 2000; 8: 37-50.

[24] Manning C, D, Raghavan P, Schütze, H. An Introduction to Information Retrieval. Text Classification and Naïve Bayes. Cambridge, England: Cambridge University Press, 2009. pp. 263-270.

[25] Peng F, Schuurmans D, Wang S. Language and Task Independent Text Categorization with Simple Language Models. In: Human Language Technology Conference; 2003; Edmonton, Canada: pp. 110-117.

[26] Peng F, Schuurmans D. Combining Naïve Bayes and n-Gram Language Models for Text Classification. In: 25th European Conference on Information Retrieval Research; 2003; Pisa, Italy: pp. 335-350.

[27] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA Data Mining Software an Update. ACM SIGKDD Explorations Newsletter 2009; 11.