T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

EMPIRICAL COMPARISON OF NAÏVE BAYES EVENT MODELS AND SMOOTHING METHODS FOR TEXT CLASSIFICATION

M.S. THESIS

ZEYNEP HİLAL KİLİMCİ

201091004

PROJECT ADVISOR: Assoc. Prof. Dr. MURAT CAN GANİZ

ISTANBUL, JANUARY 2013

PREFACE

In my thesis, I prefer to use Naïve Bayes algorithms with different event models and smoothing methods in text classification due to its easy implementation and low complexity. There are two commonly referred event models in Naïve Bayes to demonstrate text documents; multivariate Bernoulli and multinomial models. My study aims to analyze Naïve Bayes event models and smoothing methods from a different perspective by empirically. This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) grant number 111E239. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of the TÜBİTAK.

İstanbul, January 2013 Zeynep Hilal KİLİMCİ

ABSTRACT

Naïve Bayes is one of the most commonly used algorithms in text classification due to its easy implementation and low complexity. There are two commonly referred event models in Naïve Bayes for text categorization; 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 thesis aims to shed some light into this widely adopted assumption by empirically analyzing Naïve Bayes event models and smoothing methods from a different perspective. In order to clarify the difference between these event 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 at any training set size.

ÖZET

Naive bayes, kolay uygulanması ve düşük karmaşıklığı nedeniyle metin sınıflandırmada yaygın olarak kullanılan algoritmalardan biridir. Metin sınıflandırma için, Naïve bayes algoritmasının yaygın olarak kullanılan event modelleri vardır. Bunlar, Multivariate Bernoulli ve multinomial modelleridir. Çoğu çalışmada, hemen hemen her koşulda multivariate Bernoulli modele göre daha iyi performansa sahip olduğu varsayımına dayanarak model olarak multinomial model, smoothing method olarak ise Laplace seçilmiştir. Bu tez, deneysel olarak Naive Bayes event modelleri analiz etmeyi ve farklı bir bakış açısıyla yöntemleri düzgünleştirerek bu yaygın varsayıma ışık tutmayı amaçlıyor. Naive Bayes event modelleri arasındaki farkı netleştirmek için, bu modellerin metin sınıflandırma performansı İngilizce ve Türkçe olmak üzere iki farklı dildeki veri kümeleri üzerinde karşılaştırılmıştır. Kapsamlı deneyler sonucunda, multinomial modelin üstün performansının her zaman gözlenmediği görülmüştür. Multivariate Bernoulli model, farklı boyuttaki öğrenme kümelerinin olduğu koşullar altında uygun bir smoothing yöntemi ile kombine edildiğinde iyi performans gösterebilir.

ACKNOWLEDGEMENT

I would like to thank my advisor, Dr. Murat Can Ganiz for his guidance throughout this research. I am also very appreciative to all members of my family for their love, supports and prayers throughout this process. I would like to especially thank to my husband for his encouragement and understanding.

LIST OF FIGURES

Figure 4.2.1 Accuracy of MNB with LP and MVNB with LD on 20News-19997 20
Figure 4.2.2 Accuracy of MNB with LP and MVNB with LD on 20News-18828 21
Figure 4.2.3 Accuracy of MNB with LP and MVNB with GT on Mini-News..... 22
Figure 4.2.4 Accuracy of mn-NB with LP and mvb-NB with LD on Mini-News..... 23
Figure 4.2.5 Accuracy of MNB with LP and MVNB with LD on WebKB4 24
Figure 4.2.6 Accuracy of MNB with LP and MVNB with AB on Ohscal 25
Figure 4.2.7 Accuracy of MNB with LP and MVNB with LD on News3 26
Figure 4.2.8 Accuracy of MNB with LP and MVNB with LD on 1150haber 27
Figure 4.2.9 Accuracy of MNB with JM and MVNB with JM on Aahaber 28
Figure 4.2.10 Accuracy of MNB with LD and MVNB with LP on Hurriyet6c1k..... 29
Figure 4.2.11 Accuracy of MNB with LP and MVNB with JM on Milliyet9c1k 30
Figure 4.2.12 Accuracy of MNB on 20News-18828 31
Figure 4.2.13 Accuracy of MVNB on 20News-18828 32
Figure 4.2.14 Accuracy of MNB on 20News-19997 32
Figure 4.2.15 Accuracy of MVNB on 20News-19997 33
Figure 4.2.16 Accuracy of MNB on Mini-News..... 33
Figure 4.2.17 Accuracy of MVNB on Mini-News..... 34
Figure 4.2.18 Accuracy of MNB on WebKB4..... 34
Figure 4.2.19 Accuracy of MVNB on WebKB4..... 35
Figure 4.2.20 Accuracy of MNB on Ohscal..... 35
Figure 4.2.21 Accuracy of MVNB on Ohscal..... 36
Figure 4.2.22 Accuracy of MNB on News3..... 36
Figure 4.2.23 Accuracy of MVNB on News3..... 37
Figure 4.2.24 Accuracy of MNB on 1150haber..... 37
Figure 4.2.25 Accuracy of MVNB on 1150haber..... 38
Figure 4.2.26 Accuracy of MNB on Aahaber 38
Figure 4.2.27 Accuracy of MVNB on Aahaber 39
Figure 4.2.28 Accuracy of MNB on Hurriyet6c1k..... 39
Figure 4.2.29 Accuracy of MVNB on Hurriyet6c1k..... 40

Figure 4.2.30 Accuracy of MNB on Milliyet9c1k 40
Figure 4.2.31 Accuracy of MVNB on Milliyet9c1k 41
Figure 4.2.32 Accuracy of term weighting methods on 20News-18828 42
Figure 4.2.33 Accuracy of term weighting methods on 20News-19997 43
Figure 4.2.34 Accuracy of term weighting methods on Mini-News 44
Figure 4.2.35 Accuracy of term weighting methods on WebKB4 45
Figure 4.2.36 Accuracy of term weighting methods on 1150haber 46
Figure 4.2.37 Accuracy of term weighting methods on Aahaber 47
Figure 4.2.38 Accuracy of term weighting methods on Hurriyet6c1k. 48
Figure 4.2.39 Accuracy of term weighting methods on Milliyet9c1k 48

LIST OF TABLES

Table 4.1.1 Descriptions of the datasets with no preprocessing 18
Table 4.1.2 Statistics of the datasets with no preprocessing 18
Table 4.2.1 Accuracies of smoothing methods on 20News-19997 20
Table 4.2.2 Accuracies of smoothing methods on 20News-18828 21
Table 4.2.3 Accuracies of smoothing methods on Mini-News 23
Table 4.2.4 Accuracies of smoothing methods on WebKB4 24
Table 4.2.5 Accuracies of smoothing methods on Ohscal 25
Table 4.2.6 Accuracies of smoothing methods on News3 26
Table 4.2.7 Accuracies of smoothing methods on 1150haber 27
Table 4.2.8 Accuracies of smoothing methods on Aahaber 28
Table 4.2.9 Accuracies of smoothing methods on Hurriyet6c1k 29
Table 4.2.10 Accuracies of smoothing methods on Milliyet6c1k 30
Table 4.2.11 Accuracies of tfidf weighting on 20News-18828 42
Table 4.2.12 Accuracies of tfidf weighting on 20News-19997 43
Table 4.2.13 Accuracies of tfidf weighting on Mini-News 44
Table 4.2.14 Accuracies of tfidf weighting on WebKB4 45
Table 4.2.15 Accuracies of tfidf weighting on 1150haber 46
Table 4.2.16 Accuracies of tfidf weighting on Aahaber 47
Table 4.2.17 Accuracies of tfidf weighting on Hurriyet6c1k 49
Table 4.2.18 Accuracies of tfidf weighting on Milliyet9c1k 49
Table 4.2.19 Optimized values of NB smoothing methods' parameters 49
Table 4.2.20 F-measures of NB event models at 80% training set level 50
Table 4.2.21 AUC of NB event models at 80% training set level 50
Table 4.2.22 Accuracies of NB event models at 80% training set level 50
Table 4.2.23 Accuracies of NB event models at 30% training set level 51
Table 4.2.24 Accuracies of NB event models at 10% training set level 51
Table 4.2.25 Accuracies of NB event models at 5% training set level 51
Table 4.2.26 Comparison with the state of the art results 53

TABLE OF CONTENTS

PREFACE iii
ABSTRACT iv
ÖZET v
ACKNOWLEDGEMENT vi
LIST OF FIGURES vii
LIST OF TABLES ix
1. INTRODUCTION 2
2. LITERATURE REVIEW 5
3. METHODOLOGY 9
    3.1 Naïve Bayes Event Models 9
    3.2 Smoothing Methods 10
        3.2.1 Laplace Smoothing 10
        3.2.2 Good Turing Smoothing 11
        3.2.3 Jelinek-Mercer Smoothing 12
        3.2.4 Absolute Discounting Smoothing 13
        3.2.5 Linear Discounting Smoothing 14
4. CONCLUSION 16
    4.1 Experiment Setup 16
    4.2 Experiment Results 19
    4.3 Discussion 52
REFERENCES 55
BIOGRAPHY 59

1. INTRODUCTION

A Naive Bayes is a simple probabilistic classifier based on applying Bayes' theorem with naïve independence assumptions. Bayes assumption says that all attributes are independent of each other within each class. Although this assumption usually does not hold in most real-world datasets, Naive Bayes often performs well for text classification (McCallum and Nigam, 1998).

There are two commonly referred event models in Naïve Bayes to demonstrate text documents; multivariate Bernoulli and multinomial models. The first one is also known as binary independence model. In this model presence and absence of the terms is represented respectively "1" and "0" as a 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. Multinomial model is actually known as a unigram language model (McCallum and Nigam, 1998).

Most of the studies about Naïve Bayes text classification employ multinomial model based on the recommendation of the well-known paper (McCallum and Nigam, 1998). In this influential paper, authors empirically compare two event models by varying vocabulary size on five different datasets including 20 Newsgroups and WebKB. In ongoing work, they report negative influence of stemming and stopword list usage. They decide to not utilize these preprocessing methods so that the classification performance raise. 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. However, in the real world settings it is usually not the number of features but the number of documents is varying. As put forward by many studies in semi-supervised

learning literature; it is difficult, time consuming and usually expensive to obtain labeled instances as this process requires human effort. Consequently, real world applications usually start with small amount of labeled data and the amount of the labeled data increase over time. Mainly because of this problem, machine learning methods that can make inference with relatively small training data such as semi-supervised learning are gaining importance. In this study, we focus on the performance of one of the most commonly used text categorization algorithm, Naïve Bayes, under scarce 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 (Schneider, 2004; Metsis et al., 2006; Rennie et al., 2003; Juan and Ney, 2002; Kolcz and Yih, 2007; Eyheramendy et al., 2003; Peng et al., 2004) 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 (Nigam et al., 2000; Su et al., 2011).

However, this widely adopted assumption requires more investigation. From this point of view, the main motivation of our work is investigating and empirically analyzing these two approaches from a different perspective. We perform extensive experiments by varying training set size instead of vocabulary size in order to better simulate real world settings. Additionally, we utilize several different smoothing methods with these event models. Due to sparsity in training data,

unseen terms (terms that do not exist in training data) can cause zero probability problem during classification. In order to avoid this problem, some of the probability mass from existing terms is distributed over unseen terms.

We employ ten 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 (McCallum and Nigam, 1998), classification performance results can be comparable for large vocabulary sizes. After analyzing experimental results, we observe that multivariate Bernoulli model usually delivers better performance than multinomial model with large vocabulary sizes.

Besides the fact that multinomial model exhibits the superior performance on literature, our results depict 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 realized that multivariate Bernoulli event model demonstrates the best performance while number of documents are evenly or nearly distributed among categories.

2. LITERATURE REVIEW

There are several studies on two generative models of Naïve Bayes (NB) classification. McCallum and Nigam (1998) compared multivariate Bernoulli and multinomial model on five different data sets. These data sets were collected from various web pages such as Yahoo science, companies, newsgroups, university computer science departments, and Reuters. Their experimental results show that the multivariate Bernoulli event model represents better performance at smaller vocabulary sizes, whereas the multinomial model generally performs well with large vocabulary sizes.

There are several studies which try to improve upon NB based on multinomial event model. In one of these (Eyheramendy et al., 2003) 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 (Kim et al., 2006) 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 experimental results on both datasets, including 20 newsgroups and Reuters, present that the new model outperforms the traditional multinomial classifiers.

Another study (Rennie et al., 2003) also bases on multinomial model by referencing to the well-known study (McCallum and Nigam, 1998) 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 (Schneider, 2004) on NB event models, authors show that when all term frequency information is eliminated from the document vectors the multinomial model performs even better. In other words, they argue that including term frequency information is not the main reason in order to distinguish the multinomial NB from multivariate Bernoulli NB whereas it is oppositely claimed by previous studies (McCallum and Nigam, 1998; Eyheramendy et al., 2003). Furthermore, authors investigate that the main factor for the difference in performance between the two probabilistic models is negative evidence, i.e. evidence from terms that do not occur in a document. This suggests that the better performance of multinomial model compared to multivariate Bernoulli model may not stem from extra information on term frequencies. They perform classification experiments on three publicly available datasets including 20 Newsgroups, Webkb, and Ling-spam datasets. 20 Newsgroups dataset consists of 19,997 documents distributed evenly across 20 different categories. The Ling-spam corpus is formed messages from linguistics mailing list on spam messages. The last dataset, Webkb, includes web pages collected from computer science departments. Experiment results show that the multinomial NB performs better classification accuracy than the multivariate Bernoulli model due to negative evidence from documents.

The similar study (Metsis et al., 2006) 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. They also examine impact of the number of attributes on the effectiveness of the five NB versions. They find that the best results are accomplished with 3000 attributes thereby experimenting with 500, 1000, and 3000 attributes. Based on Schneider study (2004), they conclude that multinomial event model with binary attributes is one of the best performing NB versions.

Most of the studies on NB text classification including the ones mentioned above employ multinomial event model and Laplace smoothing by default. There are a few studies that attempt to use different smoothing methods. For instance, one of these studies (Juan and Ney, 2002) uses conventional multinomial model and reversed multinomial model, covering length normalization, with several different smoothing techniques which origin from statistical language modeling field and generally used with n-gram language models. These include absolute discounting with unigram backing-off and absolute discounting with unigram interpolation and Laplace smoothing. They state that absolute discounting with unigram interpolation gives better results than Laplace smoothing. They also consider document length normalization provides getting better results similar to previous studies (Rennie et al., 2003; Eyheramendy et al., 2003) on 20 Newsgroups dataset.

In a similar study (Vilar et al., 2004) 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 (Peng et al., 2004) follows the same tradition and base their study on only multinomial NB model. Instead of using unigram term probabilities, they augment NB by using n-grams and consequently advanced smoothing methods from language modelling domain such as Absolute smoothing (Ney et al., 1994), Good-Turing smoothing (Katz, 1987) and Witten–Bell smoothing (Witten and Bell, 1991). 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 (Feng and Ding, 2007) on text classification states that multinomial model almost uniformly better than Bernoulli model by following the same tradition and as a result focus solely on multinomial model. They introduce smoothing methods from language modeling area including Absolute smoothing, Good-Turing smoothing, Linear smoothing and Witten-Bell smoothing (Peng et al., 2004). However, they conduct experiments only on Chinese text documents.

The recent study (Yuan et al., 2012) applies several smoothing methods to multinomial model for short texts on Yahoo webscope 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 large margin classifier. This machine learning method aims to find a decision boundary that separates points into two classes thereby maximizing margin (Joachims, 1998; Burges, 1998). SVM projects data points into a higher dimensional space so that the data points become linearly separable by using kernel techniques. There are several kernels that can be used SVM algorithm. Linear kernel is known to perform well on text classification (Yang and Liu, 1999). We include SVM results in our experiments for comparison reasons.

3. METHODOLOGY

In this section, we focus on the Naïve Bayes event models and smoothing methods.

3.1 Naïve Bayes Event Models

Naïve Bayes is one of the most popular and commonly used machine learning algorithm in text classification due to its easy implementation and low complexity. In order to demonstrate the text documents, there are two commonly referred event models used with Naïve Bayes (NB) for text classification. First and the less popular one is multivariate Bernoulli event model which is also known as binary independence NB model. In this model, documents are considered as events and presence and absence of the terms is represented respectively "1" and "0" as a vector of binary attributes indicating occurrence of terms in the document. In other words, occurrence of the terms is placed as one within binary vector and otherwise zero. This model does not make use of 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 (McCallum and Nigam, 1998):

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

Second NB event model is multinomial model which, regardless of the terms' position, can make use of 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) (McCallum and Nigam, 1998):

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

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

3.2 Smoothing Methods

Because of sparsity in training data, missing terms (unseen events) in the document can cause "zero probability problem" in NB. To eliminate this, we need to distribute some probability mass thereby discounting the probabilities of existing terms and assigning the extra probability mass to unseen terms. This process is known as smoothing.

3.2.1 Laplace Smoothing

Laplace smoothing is one of the simplest and oldest smoothing method used in practice. This method is referred as adding one owing to adding a pseudo count to every term count. Though Laplace smoothing is one of influential method in order to avoid zero probability problem, the main disadvantage of Laplace is to give too much probability mass to previously unseen events for sparse sets of data over large vocabularies. Probability of term given class with Laplace for multinomial Naïve Bayes (Manning and Schütze, 1999):

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

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

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

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.

3.2.2 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 estimator is based 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 (Gale, 1995). 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:

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

where $P(c_j \mid d_i)$ is 1 when the document i belongs to class j. For multinomial model, T is:

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

where N_{is} is the frequency of the number of times each term w_s occurs in document d_i.

3.2.3 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 (Chen and Goodman, 1998).

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

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 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 problem, we again need to smooth maximum likelihood estimation of the term t in collection D. For multivariate Bernoulli event model $P(w_t \mid D)$ is:

$$P(w_t \mid D) = \frac{1 + N(w_t, D)}{2 + N(D)} \tag{11}$$

For multinomial event model $P(w_t \mid D)$ is:

$$P(w_t \mid D) = \frac{1 + N(w_t, D)}{|V| + N(D)} \tag{12}$$

where $N(w_t, D)$ is the total number of term counts, and $N(D)$ represents the total number of documents in collection D.

3.2.4 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 (Chen and Rosenfeld, 2000). The idea behind of this smoothing technique is that it gains free probability mass from the seen terms thereby discounting a small constant to every term count (Vilar et al., 2004). In other words, non-zero frequencies are figured out by a small constant amount δ and the frequency is uniformly redistributed over unseen terms (Manning and Schütze, 1999).

$$\delta = \frac{N_1}{N_1 + 2N_2} \tag{13}$$

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.

$$P(w_t \mid c_j) = \begin{pmatrix} \frac{r-\delta}{N} & r > 0 \\ \delta \times \frac{\sum_{i>0} N_i}{N_0 \times N} & r = 0 \end{pmatrix} \tag{14}$$

In Eq. (14), 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. For multivariate Bernoulli event model, N is:

$$\mathrm{N} = \sum_{i=1}^{|D|} P(c_j \mid d_i) \tag{15}$$

where $P(c_j \mid d_i)$ is 1 when the document i belongs to class j. For multinomial model, N is:

$$\mathrm{N} = \sum_{s=1}^{|V|} \sum_{k=1}^{|D|} N_{ks} P(c_j \mid d_k) \tag{16}$$

where N_{ks} is the frequency of the number of times each term w_s occurs in document d_k.

3.2.5 Linear Discounting Smoothing

In the linear discounting method, the non-zero maximum likelihood estimation frequencies are discounted by a constant less than one, and the remaining probability mass is again redistributed

across novel terms. That is, this estimator rescales the probability of unseen terms thereby subtracting or multiplying by a small constant instead of zero (Manning et al., 2009).

$$P_{ml}(w_t \mid c_j) = \frac{N(c_t, D)}{N(c, D)} \tag{17}$$

$$P(w_t \mid c_j) = \begin{pmatrix} (1-\gamma) \times P_{ml}(w_t \mid c_j) & r > 0 \\ \frac{\gamma}{N_0} & r = 0 \end{pmatrix} \tag{18}$$

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.

4. CONCLUSION

4.1 Experiment Setup

We use ten benchmark datasets with different sizes and properties to analyze performance of the algorithms for text classification. These include three variants of 20 Newsgroups[1] dataset. First one is the original 20 Newsgroups dataset and it is named as 20News-19997. Second one is called 20News-18828 and it has fewer documents since duplicates postings are removed. Additionally for each posting headers are deleted except “From" and "Subject" headers. Third one is a small subset of the original dataset composed of 100 postings per class and it is called as Mini-Newsgroups[2]. All of them are divided into twenty different categories. Majority of studies (McCallum and Nigam, 1998; Schneider, 2004; Rennie et al., 2003; Juan and Ney, 2002; Eyheramendy et al., 2003; Kim et al., 2006; Peng et al., 2003; Peng and Schurmans, 2003; Hall et al., 2009) also choose 20 Newsgroups for text classification experiments.

Our fourth dataset is the WebKB[3] dataset which includes web pages collected from computer science departments of different universities. There are seven categories which are student, faculty, staff, course, project, department and other. We use four class version of the WebKB dataset which is used in some studies (McCallum and Nigam, 1998; Schneider, 2004; Vilar et al., 2004). This dataset is named as WebKB4.

We also use two large text datasets in WEKA (Hall et al. 2009): News3 and Ohscal28. These are previously used in semi-supervised NB experiments (Su et al., 2011). One of the most important differences between 20 Newsgroups variants and others including WebKB is the class distribution. While 20 Newsgroups variants have almost equal number of documents per class,

[1] http://people.csail.mit.edu/people/jrennie/20Newsgroups
[2] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html
[3] http://www.cs.cmu.edu/~textlearning

others have highly skewed class distribution. Additionally, News3 has exceptionally high number of classes given its size in terms of the number of document.

The other four datasets consist of Turkish text documents. Milliyet9c1k dataset includes text from columns of Turkish newspaper Milliyet[4] from years, 2002 to 2011. It contains nine categories and 1000 documents for each category. The categories of this dataset are café (cafe), dünya (world), ege (region), ekonomi (economy), güncel (current), siyaset (politics), spor (sports), Türkiye (Turkey), yaşam (life).

Hurriyet6c1k dataset includes news from 2010 to 2011 on Turkish newspaper Hürriyet[5]. It contains six categories and 1000 documents for each category. Categories in this dataset are dünya (world), ekonomi (economy), güncel (current), spor (sports), siyaset (politics), yaşam (life). 1150haber dataset is obtained from a study done in a study (Amasyalı and Beken, 2009). It consists of 1150 Turkish news texts in five classes (economy, magazine, health, politics, sports) and 230 documents for each category.

Aahaber (Tantuğ, 2010) is a dataset consists of newspaper articles broadcasted by Turkish National News Agency, Anadolu Agency[6]. This dataset includes eight categories and 2,500 documents for each category. Categories are Turkey, world, politics, economics, sports, education science, culture art and environment health. Milliyet9c1k and 1150haber include the writings of the column writers therefore they are longer and more formal. On the other hand, Hurriyet6c1k datsaset contains traditional news articles. They are more irregular, much shorter than documents of the other datasets. Descriptions of the datasets, when no preprocessing is applied, are given in Table 6.1 including number of classes (|C|), number of documents (|D|) and the vocabulary size (|V|).

[4] www.milliyet.com.tr
[5] www.hurriyet.com.tr
[6] www.aa.com.tr

We only filter infrequent terms whose document frequency is less than three. We don't apply any stemming or stopword filtering in order to avoid any bias that can be introduced by stemming algorithms or stop list.

Table 4.1.1 Descriptions of the datasets with no preprocessing

DATA SET	|C|	|D|	|V|
20NEWS-18828	20	18,828	50,570
20NEWS-19997	20	19,997	43,553
MINI-NEWS	20	2,000	13,943
WEBKB4	4	4,199	16,116
OHSCAL	10	11,162	11,466
NEWS3	44	9,558	26,833
1150HABER	5	1,150	11,040
AAHABER	8	20,000	14,395
HURRIYET6C1K	6	6,000	18,280
MILLIYET9C1K	9	9,000	63,371

Table 4.1.2 Statistics of the datasets with no preprocessing

DATA SET	Avg. #of terms per document	Avg. term length
20NEWS-18828	144,40	6,48
20NEWS-19997	127,07	6,53
MINI-NEWS	179,78	6,12
WEBKB4	151,79	6,46
OHSCAL	60,31	7,36
NEWS3	234,52	6,42
1150HABER	126,82	7,29
AAHABER	23,36	7,65
HURRIYET6C1K	101,54	7,74
MILLIYET9C1K	318,12	8,43

As mentioned before we vary the training set size by using following percentages of the data for training and the rest for testing: 1%, 5%, 10%, 30%, 50%, 70%, and 90%. These percentages are indicated with "ts" prefix to avoid confusion with accuracy percentages. We also conduct experiments using 80% of the data and no document frequency filtering in order to compare our results with the ones in the literature.

We run 10 random splits for each of the training set percentages above per class. This approach is similar to famous studies (McCallum and Nigam, 1998; Rennie et al., 2003) where they use 80% training data and 20% for test.

4.2 Experimental Results

There are two generative event models that are in our experiments augmented by standard deviations. We include F-measure (F1) and Area Under Curve (AUC) results only for 80% training data level due to length restrictions. We also provide statistical significance tests in several places by using Student's t-Test especially when results of different algorithms are close to each other (e.g. about 2-3% difference). We use $\alpha = 0.05$ significance level and consider the difference is statistically significant if the probability associated with Student's t-Test is lower. It is important to note that we always use the accuracy of best performing 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) (with the exception of Mini-Newsgroups dataset in which Good-Turing (GT) performs much better with MVNB).

We start with the original 20 Newsgroups dataset which is named as 20News-19997. Overall, for all training set percentages LD is the best performing smoothing method for multivariate Bernoulli Naïve Bayes (MVNB). For the multinomial Naïve Bayes (MNB), LP smoothing performs best for most of the cases. For training set percentages 90 (ts90) and 70 (ts70) MNB has slightly higher accuracy than MVNB however the difference is not statistically significant. Starting from 50% training data, accuracy of MVNB exceeds MNB by approximately 2% at ts30, 19% at ts10, and 26% at ts5 levels. The differences are statistically significant. This trend can be seen in Figure 4.2.1 and Table 4.2.1.

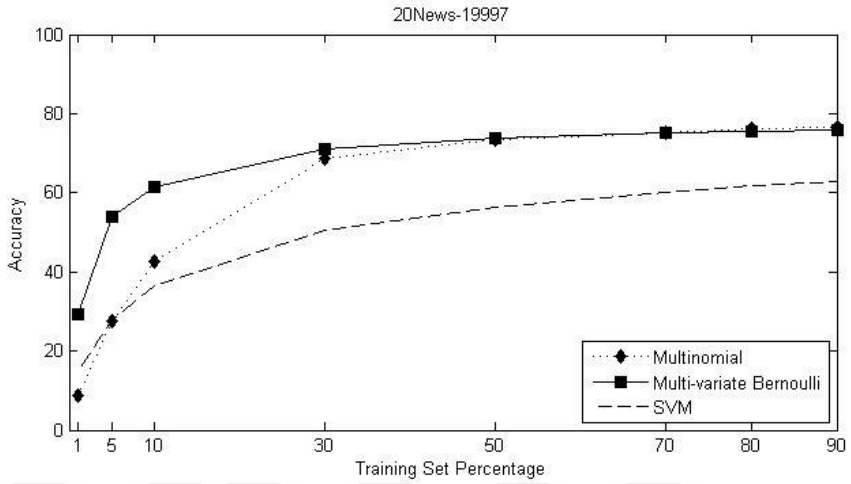

Figure 4.2.1 Accuracy of MNB with LP and MVNB with LD on 20News-19997.

Table 4.2.1 Accuracies of smoothing methods on 20News-19997.

20NEWS-19997		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	70,680	70,380	69,675	68,681	66,300	58,561	52,336	27,363
	GT	70,655	70,503	69,572	68,127	65,151	56,205	49,698	26,311
	JM	69,750	69,768	69,358	68,335	65,901	57,921	51,741	29,885
	LP	63,595	63,183	62,157	59,979	55,922	43,612	34,535	15,111
	LD	75,820	75,680	75,145	**73,942**	**70,992**	**61,353**	**53,912**	**29,238**
MNB	*AB*	55,655	55,395	55,560	51,444	47,125	33,311	29,270	10,525
	GT	57,675	57,013	56,872	52,070	47,235	32,102	28,169	9,488
	JM	55,095	55,263	56,270	53,269	50,669	39,307	35,193	13,432
	LP	**76,515**	**76,058**	**75,197**	73,477	68,808	42,764	27,541	8,608
	LD	67,390	66,888	66,875	64,248	60,348	47,421	40,473	14,895
SVM	*Linear*	62,765	61,838	60,225	56,399	50,371	36,463	27,802	14,880

For 20News-18828, we have even better results for MVNB. Again, 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 4.2.2, accuracy difference is 15% between MVNB and MNB at ts10 level and at ts5 level it increases to about 29%. This trend can be clearly seen in Figure 4.2.2 and Table 4.2.2.

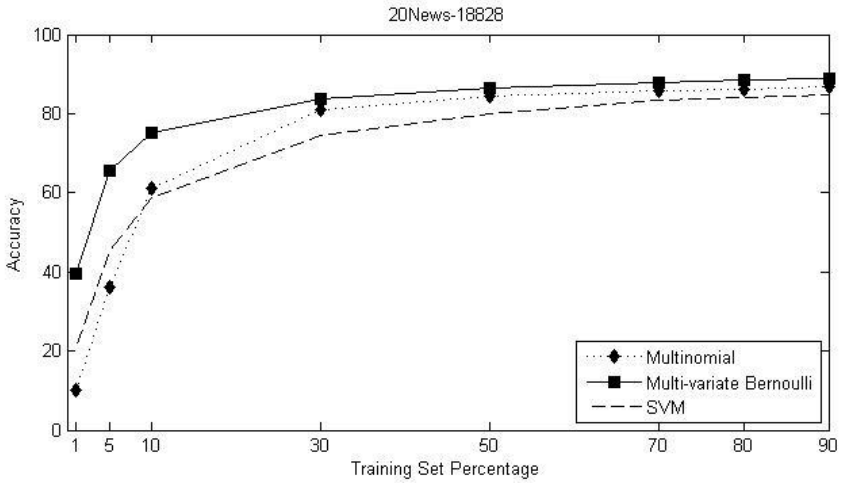

Figure 4.2.2 Accuracy of MNB with LP and MVNB with LD on 20News-18828.

Table 4.2.2 Accuracies of smoothing methods on 20News-18828.

20NEWS-18828		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	84,072	83,932	83,069	82,025	79,520	72,494	63,685	36,815
	GT	84,299	84,006	83,111	81,806	78,706	70,233	60,719	34,786
	JM	83,527	83,499	82,915	81,738	79,116	71,141	62,728	36,834
	LP	76,705	75,801	74,920	71,790	65,549	50,167	37,175	16,073
	LD	**88,757**	**88,518**	**87,723**	**86,363**	**83,605**	**75,139**	**65,550**	**39,418**
MNB	*AB*	76,235	74,875	73,981	69,620	67,202	48,589	32,727	12,952
	GT	78,361	76,795	75,694	71,047	67,848	48,486	30,586	11,813
	JM	76,282	75,130	75,038	71,722	70,746	57,280	42,016	19,718
	LP	86,679	86,259	85,864	84,566	80,832	60,971	36,066	10,180
	LD	84,870	84,046	83,424	81,193	78,168	64,883	49,255	22,088
SVM	*Linear*	84,929	84,183	83,419	80,029	74,401	58,639	45,462	20,886

In Mini-Newsgroups, which is the third variant of 20 Newsgroups dataset we have unusual results in terms of smoothing method performances of MVNB. In contrast to the previous datasets, Good-Turing smoothing (GT) yields by far highest accuracies when combined with MVNB. It is followed by absolute discounting (AB) and after that by Jelinek-Mercer (JM) in some cases. In any case, Laplace smoothing yields the lowest values in MVNB and highest values in MNB. Except these differences, we see the same pattern in Figure 4.2.4 and Table 4.2.3.

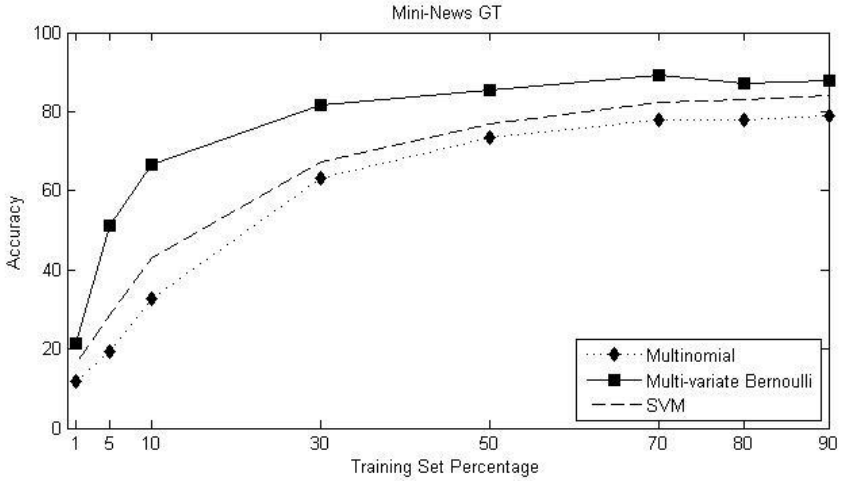

Figure 4.2.3 Accuracy of MNB with LP and MVNB with GT on Mini-News.

MVNB with LD still outperforms MNB with Laplace. The difference is about 1% in ts70, 2% in ts50, 5% in ts30, and rise to 19% in ts10 and % 20 percent in ts5. These differences are statistically significant. The difference between MVNB with GT and MNB with Laplace is much more remarkable as can be seen in Figure 4.2.3 where MVNB with GT significantly outperforms MNB and SVM by a wide margin in all cases.

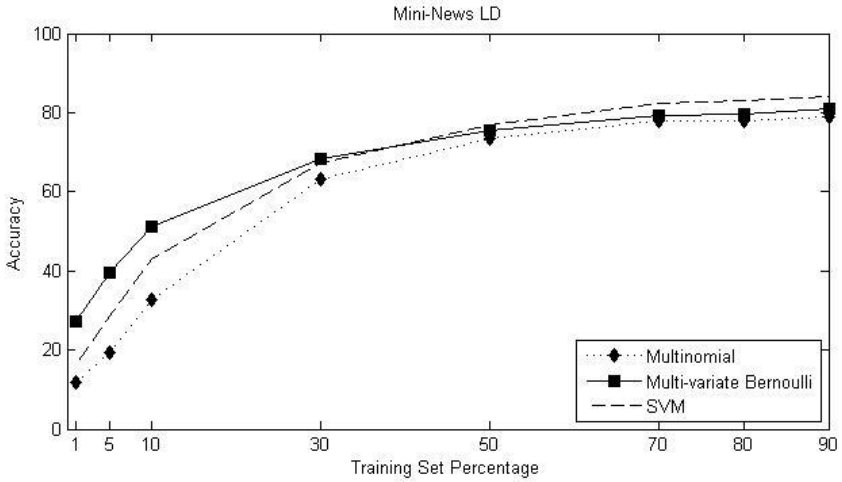

Figure 4.2.4. Accuracy of mn-NB with LP and mvb-NB with LD on Mini-News.

Table 4.2.3. Accuracies of smoothing methods on Mini-News.

MINI-NEWS		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	85,700	83,875	84,750	81,150	75,607	57,556	42,242	26,934
	GT	**87,800**	**87,250**	**89,083**	**85,510**	**81,750**	**66,639**	**51,163**	**21,301**
	JM	80,450	79,825	80,617	76,000	70,750	54,267	41,253	26,673
	LP	77,850	76,700	77,267	69,650	61,721	37,294	25,279	16,301
	LD	80,850	79,525	79,283	75,410	68,393	51,139	39,616	27,112
MNB	*AB*	58,800	56,350	55,117	50,780	41,000	25,939	18,647	12,434
	GT	59,100	56,825	55,633	50,750	39,321	23,689	15,226	11,255
	JM	62,900	61,075	59,983	56,110	46,871	31,039	23,221	16,235
	LP	79,000	77,900	78,033	73,510	63,329	32,800	19,468	11,852
	LD	65,200	64,075	63,300	59,660	50,829	34,056	25,689	16,429
SVM	*Linear*	84,150	83,000	82,467	76,810	67,171	42,950	28,616	16,408

In WebKB4 which has different properties such as skewed class distribution we observe somewhat different patterns. Similar to the other datasets, 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 4.2.5 and Table 4.2.4 summarizes these results.

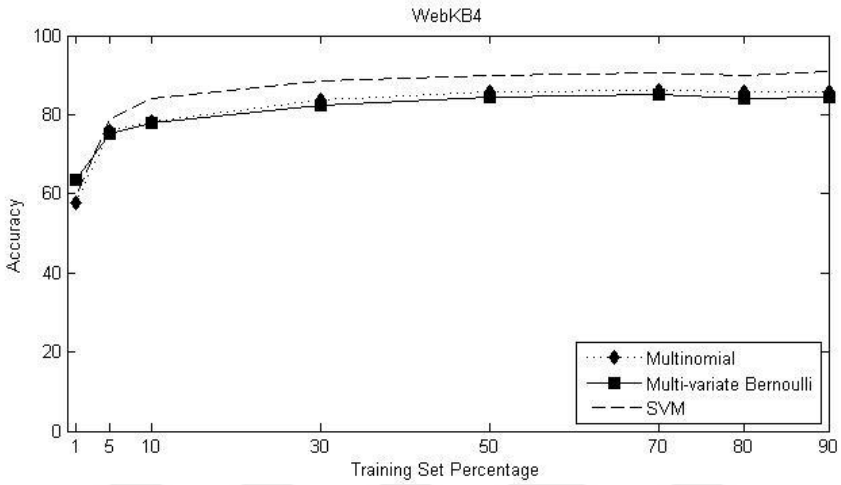

Figure 4.2.5 Accuracy of MNB with LP and MVNB with LD on WebKB4.

Table 4.2.4 Accuracies of smoothing methods on WebKB4.

WEBKB4		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	81,848	81,605	82,456	81,924	80,592	78,397	74,331	63,968
	GT	82,322	82,247	82,623	82,124	80,490	78,034	73,812	61,746
	JM	81,445	81,605	82,060	81,790	80,187	78,370	74,546	61,943
	LP	79,360	79,180	78,368	76,576	70,755	54,847	45,321	39,086
	LD	84,550	84,102	84,968	84,319	82,371	77,857	75,120	**63,521**
MNB	*AB*	63,318	62,996	62,448	66,457	62,895	60,772	55,158	44,555
	GT	64,076	63,769	63,193	67,990	64,442	61,696	56,140	44,853
	JM	64,597	64,233	63,859	68,900	66,361	65,368	63,396	55,089
	LP	**85,687**	**85,636**	**86,014**	**85,648**	**83,895**	**78,143**	**75,975**	57,665
	LD	66,398	66,635	67,124	73,943	70,711	68,217	67,288	56,371
SVM	*Linear*	90,782	89,845	90,475	89,795	88,537	84,238	78,815	59,577

In Ohscal, the best performing smoothing method for MVNB is AB, and LP for MNB. Yet, accuracy of MVNB is slightly lower (about 1%) than MNB for all training set percentages. Because of these small differences, accuracy results are not statistically significant.

SVM generally outperforms MVNB and MNB except ts5 and ts1. Performance of SVM reaches maximum 3% in all training set percentages except ts1 where MVNB and MNB have about 7% higher accuracy. Ohscal shows similar behaviour to WebKB4. This pattern can be seen in Figure 4.2.6 and Table 4.2.5.

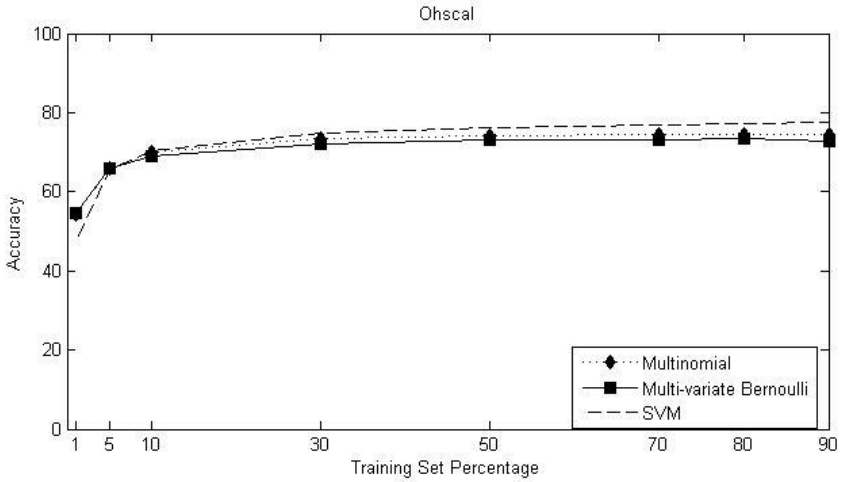

Figure 4.2.6 Accuracy of MNB with LP and MVNB with AB on Ohscal.

Table 4.2.5. Accuracies of smoothing methods on Ohscal.

OHSCAL		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	72,830	73,511	73,259	72,978	72,235	69,095	65,912	**54,468**
	GT	72,705	73,408	73,208	72,919	72,130	68,834	65,648	54,428
	JM	72,705	73,175	72,719	72,518	71,776	68,909	66,296	54,740
	LP	72,884	73,305	73,286	72,290	68,548	47,392	25,716	14,518
	LD	71,723	72,160	71,768	70,777	68,982	63,978	59,994	49,205
MNB	*AB*	72,777	73,104	72,761	71,927	70,956	67,400	63,286	50,751
	GT	72,768	72,898	72,624	71,750	70,659	66,822	62,908	50,606
	JM	72,589	72,616	72,340	71,572	70,493	66,207	62,115	51,060
	LP	**74,384**	**74,468**	**74,517**	**74,162**	**73,364**	**69,875**	**65,922**	54,248
	LD	71,696	71,691	71,225	69,959	68,316	63,343	59,855	49,930
SVM	*Linear*	77,714	77,223	76,840	76,039	74,795	70,407	65,519	47,298

In News3, LD is the best performing smoothing method at ts90, ts80, ts70 and JM shows higher accuracy performance at remaining training set percentages for MVNB. For MNB, LP is the best performing smoothing method. However, MVNB with AB outperforms MNB Laplace at ts10, ts5, and ts1 levels. Accuracy difference is 3% at ts10 level, 8% at ts5 level and 10% at ts10 level. This trend can be clearly seen in Figure 4.2.7 and Table 4.2.6. Moreover, SVM yields by far highest accuracies in all training set percentages except ts1 where MVNB has about 3% higher accuracy.

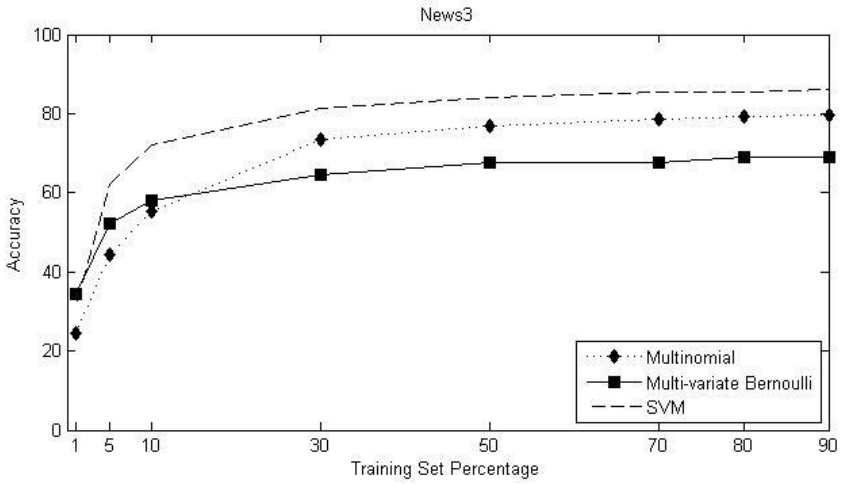

Figure 4.2.7 Accuracy of MNB with LP and MVNB with LD on News3.

Table 4.2.6. Accuracies of smoothing methods on News3.

NEWS3		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	69,138	68,876	67,753	67,415	64,893	55,793	47,150	30,165
	GT	68,388	67,986	66,630	66,346	63,236	55,561	48,529	26,185
	JM	68,984	68,928	67,743	67,638	64,459	**58,004**	**52,218**	**34,546**
	LP	50,595	48,115	44,809	37,901	25,261	13,127	12,103	7,297
	LD	70,626	69,943	68,790	67,024	62,908	49,857	41,204	28,220
MNB	*AB*	42,947	40,192	39,411	38,549	35,419	28,010	21,821	15,154
	GT	42,741	39,694	38,981	38,102	34,861	26,502	19,220	11,523
	JM	44,476	40,601	40,142	39,528	37,469	34,357	25,737	18,920
	LP	**79,538**	**79,430**	**78,499**	**77,047**	**73,330**	55,160	44,188	24,444
	LD	50,924	47,773	47,386	46,630	43,254	37,426	27,468	18,597
SVM	*Linear*	86,263	85,282	85,392	83,915	81,187	72,059	62,011	31,229

In 1150haber, for all training set percentages LD is the best performing smoothing method for MVNB. For MNB, Laplace smoothing performs best for most of the cases. The accuracy of MVNB with LD exceeds MNB with Laplace by approximately 3% at ts10, 10% at ts5, and 13% at ts1 levels. The differences are statistically significant. At remaining training set levels, MNB has slightly higher accuracy than MVNB yet the difference is not statistically significant. This pattern can be seen in Figure 4.2.8 and Table 4.2.7.

.

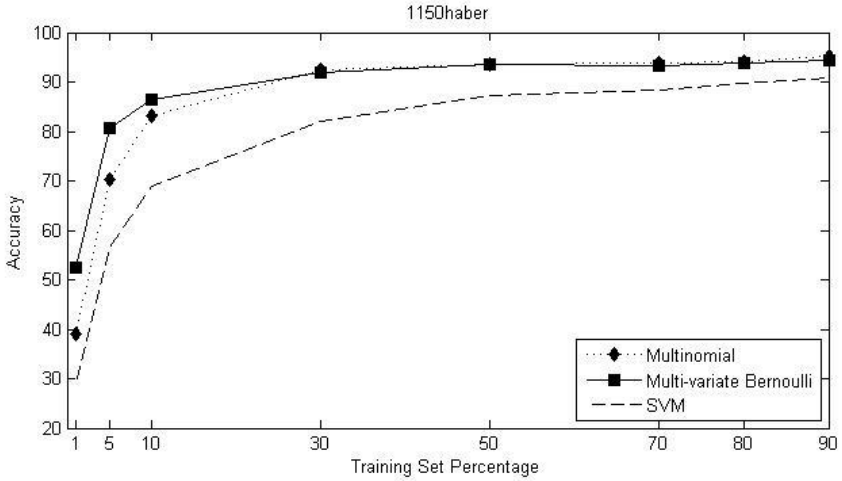

Figure 4.2.8 Accuracy of MNB with LP and MVNB with LD on 1150haber.

Table 4.2.7 Accuracies of smoothing methods on 1150haber.

1150HABER		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	91,130	90,478	89,449	88,939	87,888	81,527	75,580	41,746
	GT	91,391	90,174	89,043	88,348	85,640	76,493	68,712	27,202
	JM	91,478	90,261	88,667	87,878	84,969	73,005	59,397	29,289
	LP	87,391	85,913	83,130	80,452	73,863	52,744	36,009	24,105
	LD	94,261	93,739	93,217	93,513	91,863	**86,570**	**80,603**	**52,395**
MNB	*AB*	87,304	86,696	84,435	83,322	77,329	66,589	58,557	31,333
	GT	89,565	88,522	86,754	85,009	78,509	65,884	59,397	29,246
	JM	91,913	91,217	90,609	90,017	86,335	78,010	69,726	40,746
	LP	**95,217**	**94,000**	**93,971**	**93,617**	**92,422**	83,159	70,174	39,026
	LD	91,826	91,522	91,188	90,470	86,944	77,749	69,269	40,377
SVM	*Linear*	90,870	89,652	88,377	87,270	82,000	69,034	56,548	29,035

In Aahaber, JM smoothing yields by far highest accuracies when combined with MVNB. JM smoothing also yields the highest values in MNB at most of the training set levels except ts10, ts5, and ts1. Starting from 50% training data, accuracy of MVNB has slightly higher than MNB yet the difference is not statistically significant. This trend can be seen in Figure 4.2.9 and Table 4.2.8.

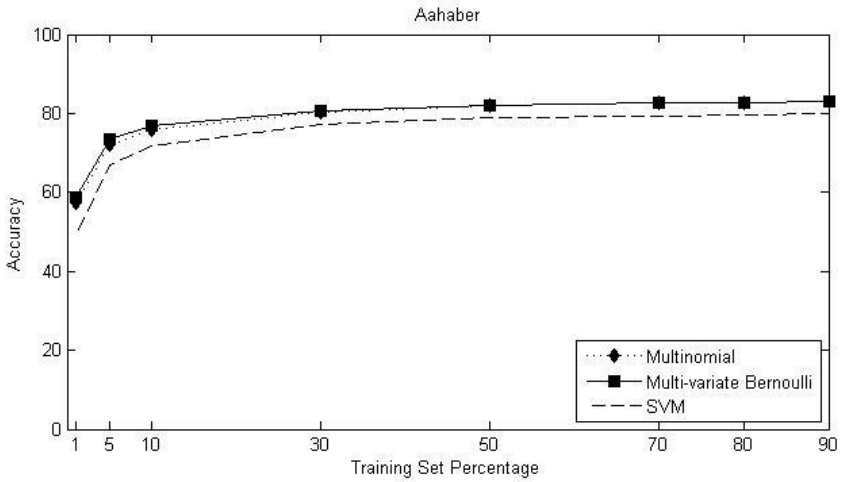

Figure 4.2.9 Accuracy of MNB with JM and MVNB with JM on Aahaber.

Table 4.2.8 Accuracies of smoothing methods on Aahaber.

AAHABER		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	82,300	81,803	81,640	80,999	79,788	76,047	72,462	57,522
	GT	82,480	82,165	82,048	81,415	80,076	76,120	72,372	56,971
	JM	82,935	82,703	82,630	**82,004**	**80,639**	**76,937**	**73,341**	**58,610**
	LP	82,160	81,623	81,600	80,929	79,689	75,542	71,118	49,808
	LD	82,495	82,075	81,948	80,950	78,944	74,252	70,231	56,186
MNB	*AB*	82,290	81,963	81,785	81,167	79,739	75,938	72,315	57,533
	GT	82,630	82,310	82,110	81,418	79,811	75,853	71,821	56,327
	JM	**83,210**	**82,798**	**82,637**	81,977	80,363	75,990	71,945	57,520
	LP	82,565	82,135	82,002	81,431	80,082	76,072	72,369	56,669
	LD	82,490	82,288	81,982	81,057	79,063	74,397	70,361	56,625
SVM	*Linear*	79,900	79,623	79,377	78,834	77,176	71,725	66,882	49,114

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 4.2.10 and Table 4.2.9.

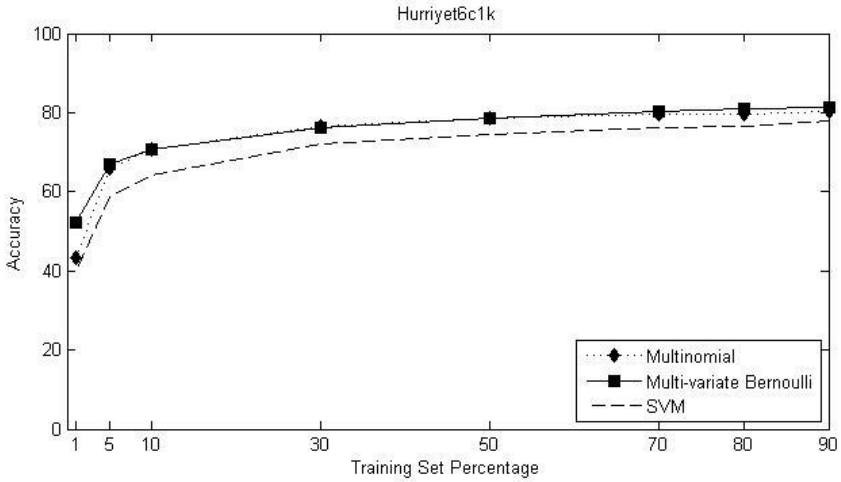

Figure 4.2.10 Accuracy of MNB with LD and MVNB with LP on Hurriyet6c1k.

Table 4.2.9 Accuracies of smoothing methods on Hurriyet6c1k.

HURRIYET6C1K		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	77,517	77,433	76,689	76,257	74,760	70,941	67,730	49,923
	GT	78,650	78,008	77,617	76,983	75,202	70,565	67,184	46,899
	JM	79,050	78,792	78,233	77,780	75,948	71,004	66,889	43,672
	LP	77,050	76,600	75,983	75,217	72,914	65,428	58,232	30,135
	LD	**81,500**	**81,158**	**80,217**	**78,687**	76,240	70,557	**66,902**	**52,298**
MNB	*AB*	75,467	74,625	74,194	73,483	70,940	65,565	63,325	44,047
	GT	76,517	75,875	75,250	74,393	71,488	64,994	62,607	38,613
	JM	78,383	77,700	77,372	76,397	74,214	68,098	65,549	48,167
	LP	80,250	79,775	79,672	78,753	**76,629**	**70,644**	65,889	43,455
	LD	79,150	78,658	78,106	76,923	74,310	67,969	65,165	48,030
SVM	*Linear*	77,933	76,583	76,289	74,603	72,124	64,119	58,809	39,753

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 traning set percentages. 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 4.2.11 and Table 4.2.10 summarizes these results in detail.

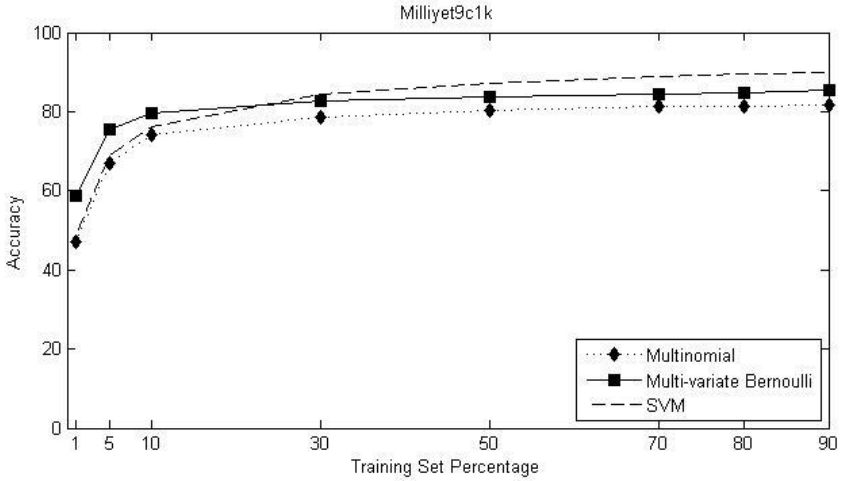

Figure 4.2.11 Accuracy of MNB with LP and MVNB with JM on Milliyet9c1k.

Table 4.2.10 Accuracies of smoothing methods on Milliyet6c1k.

MILLIYET9C1K		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*AB*	82,933	82,739	82,444	81,689	80,783	78,269	74,855	62,181
	GT	84,011	83,600	83,459	82,507	81,627	78,899	75,650	63,836
	JM	**85,311**	**84,639**	**84,496**	**83,573**	**82,551**	**79,584**	**75,677**	**58,758**
	LP	83,811	83,561	83,404	82,480	81,319	76,491	66,794	34,811
	LD	83,933	83,239	82,807	81,296	78,956	74,631	70,450	59,768
MNB	*AB*	72,556	71,717	71,763	71,213	70,070	67,642	61,584	48,604
	GT	72,456	71,689	71,426	70,529	68,821	64,644	57,571	43,448
	JM	75,711	74,994	74,893	74,273	72,803	69,870	64,556	52,740
	LP	81,733	81,483	81,233	80,322	78,711	74,164	66,986	47,149
	LD	76,178	75,578	75,089	74,404	72,679	69,370	64,389	52,533
SVM	*Linear*	89,778	89,406	88,856	87,020	84,316	76,254	68,860	48,686

The accuracy differences between event models of Naïve Bayes and SVM can be observed in detail in the appendix section for each dataset.

Figure 4.2.12 and 4.2.13 shows 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. These patterns can be seen in Figure 4.2.12 and 4.2.13. 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, 20 News-19997, 1150haber, Hurriyet6c1k, WebKB4. GT and JM methods have the highest results for Mini-Newsgroups and Milliyet9c1k datasets respectively. Accuracy graphics of smoothing methods can be found in detail for each dataset below.

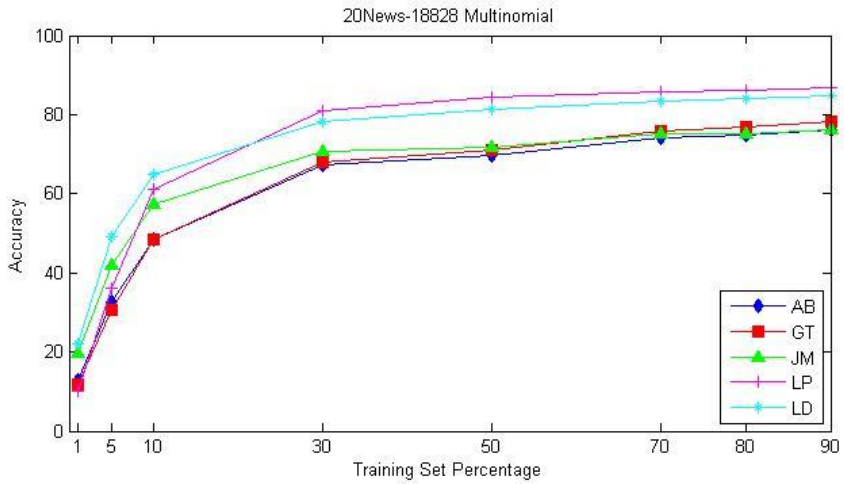

Figure 4.2.12 Accuracy of MNB on 20News-18828.

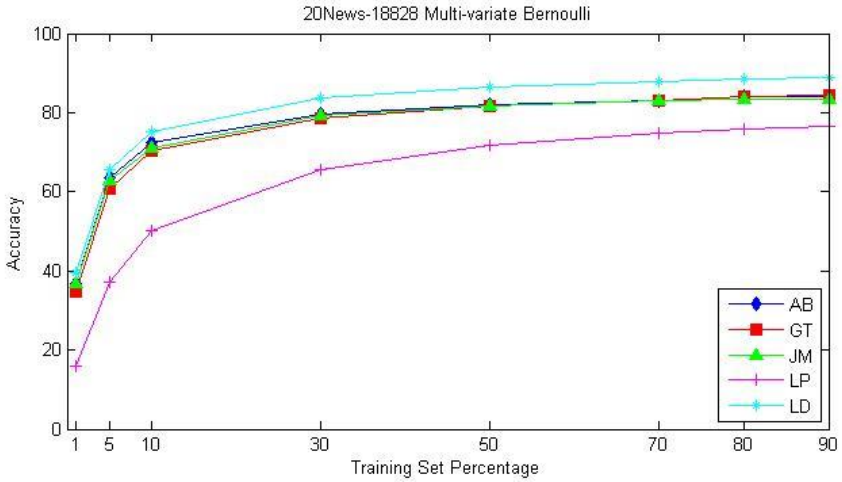

Figure 4.2.13 Accuracy of MVNB on 20News-18828.

Figure 4.2.14 shows the performance of the smoothing methods on 20News-19997 dataset for MNB. Laplace smoothing yields by far highest accuracies in all training percentages except ts10, ts5, ts1 where LD has about 5%, 20%, 6% higher accuracy, respectively.

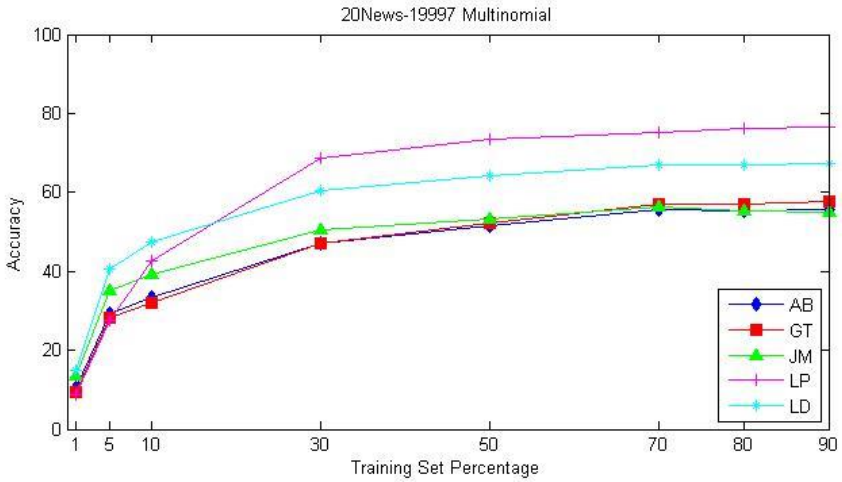

Figure 4.2.14 Accuracy of MNB on 20News-19997.

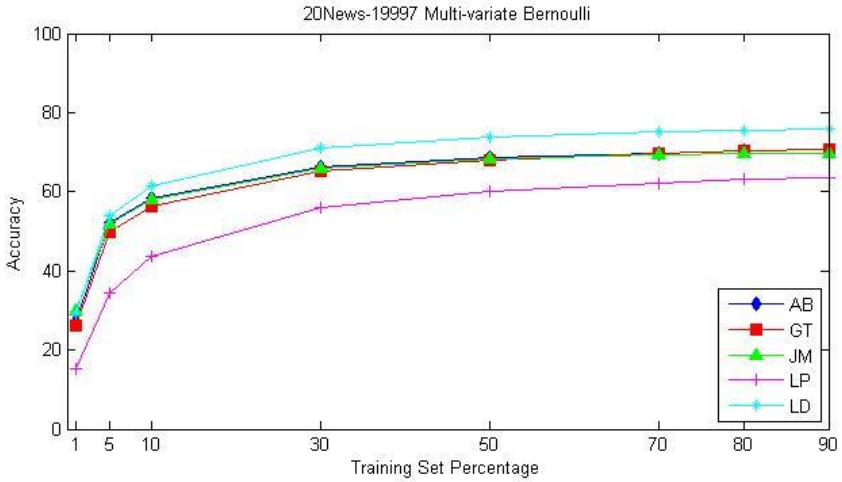

Figure 4.2.15 Accuracy of MVNB on 20News-19997.

In Figure 4.2.15, MVNB with LD outperforms MNB with Laplace on 20News-19997. It is followed by AB, GT and JM and after that by LP in some cases. The highest and lowest smoothing values are observed in a significant way although the others are very close to each other.

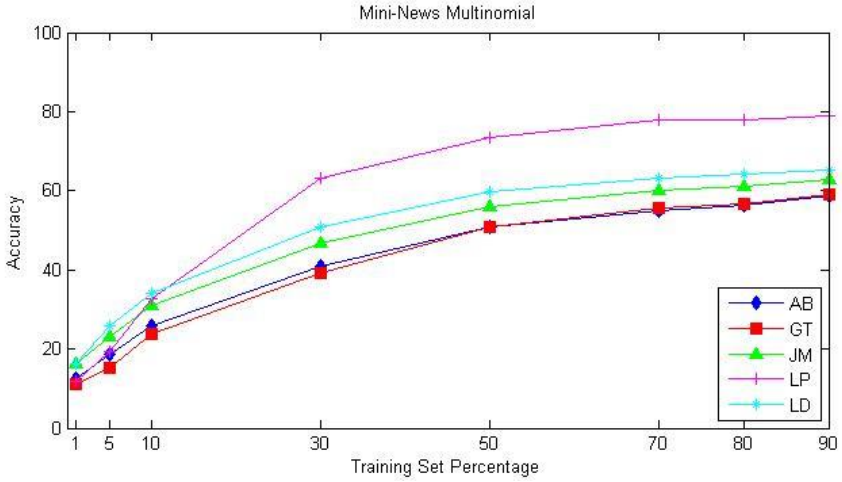

Figure 4.2.16 Accuracy of MNB on Mini-News.

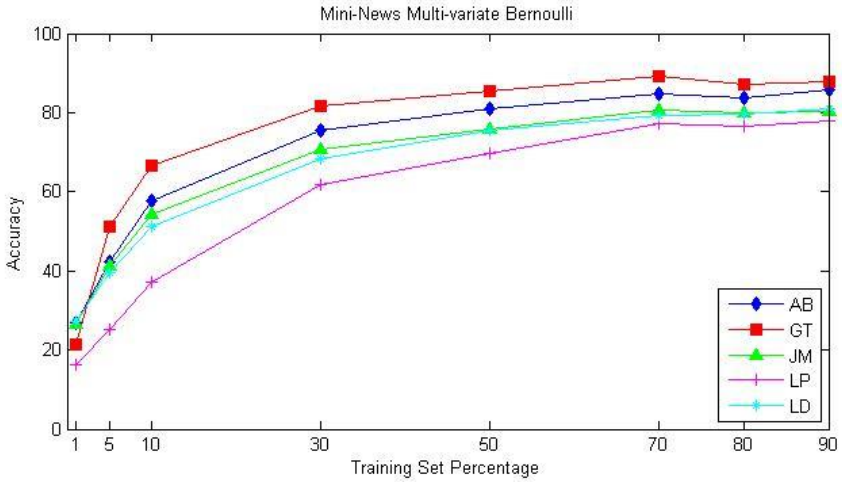

Figure 4.2.17 Accuracy of MVNB on Mini-News.

For Mini-News, we have even better results for MVNB. The best performing smoothing method for MVNB is GT and it is followed by AB, JM, LD, and LP respectively. GT surpases other smoothing methods by at least 2% in all training set percentages. For MNB, again LP smoothing performs best for most of the cases. Starting from 30% training data, accuracy of LP exceeds other smoothing methods by at least 13% in all training set percentages except ts1, ts5 and ts10 where LD has about 5%, 6% and 2% higher accuracy, respectively. Figure 4.2.16 and Figure 4.2.17 summarize these results.

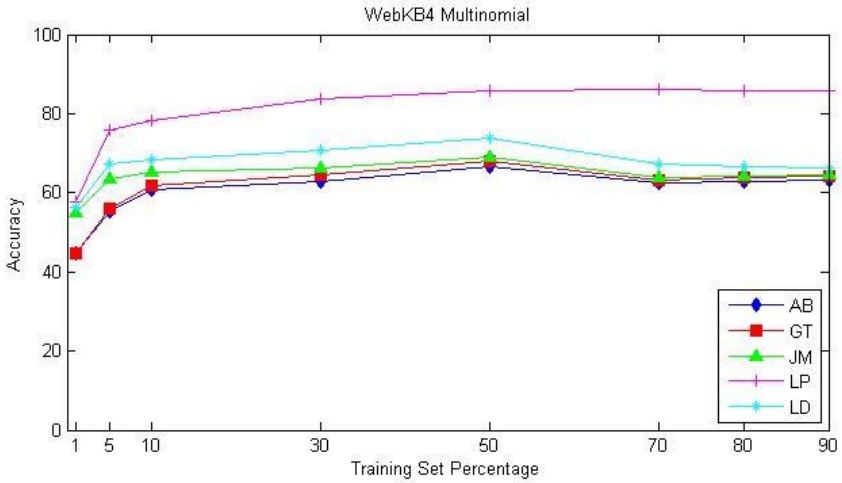

Figure 4.2.18 Accuracy of MNB on WebKB4.

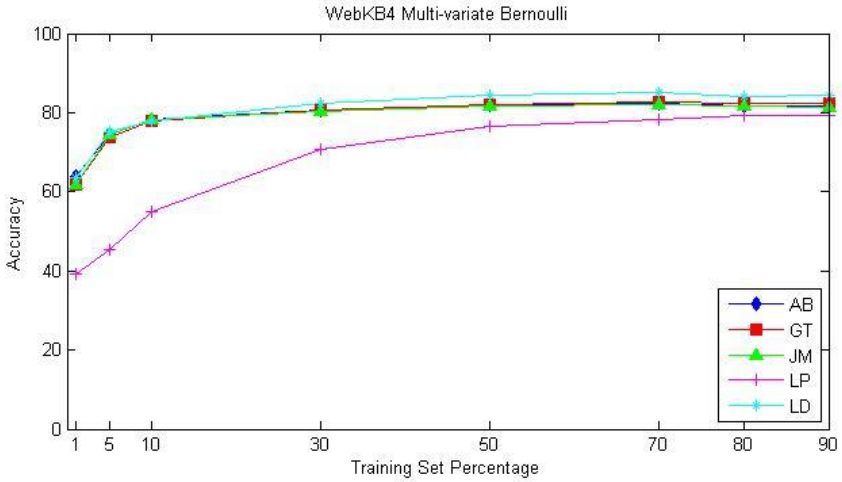

Figure 4.2.19 Accuracy of MVNB on WebKB4.

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.

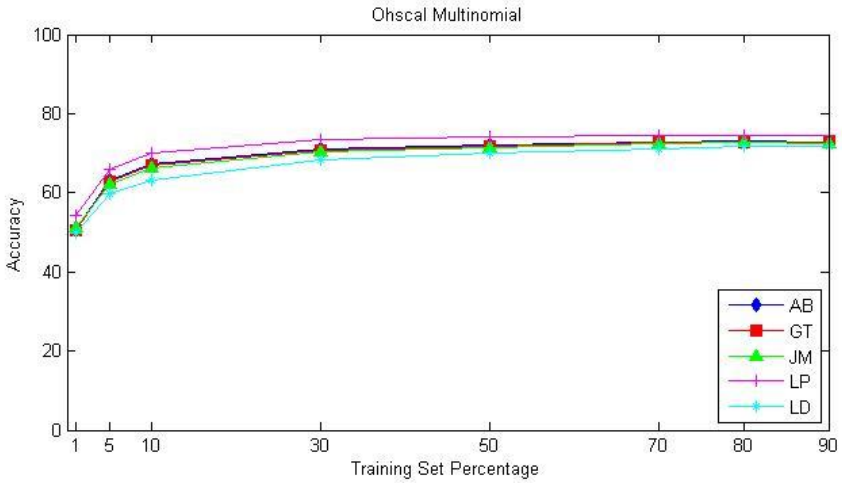

Figure 4.2.20 Accuracy of MNB on Ohscal.

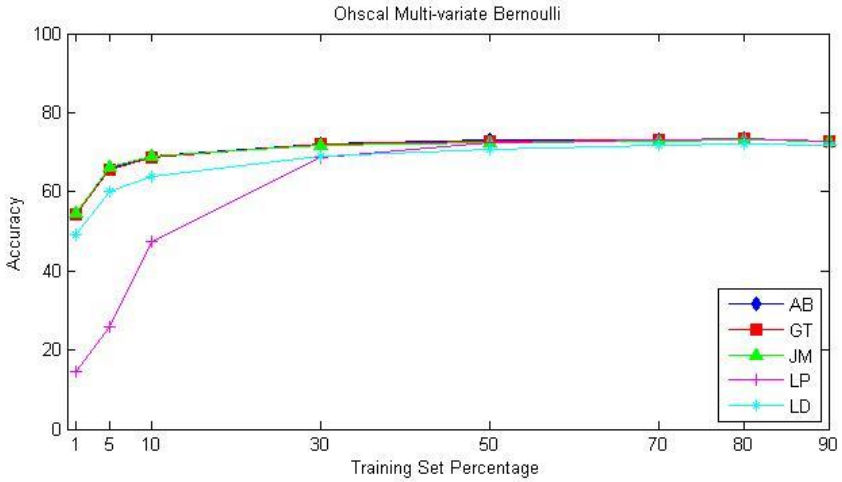

Figure 4.2.21 Accuracy of MVNB on Ohscal.

In Ohscal, the best performing smoothing method for MVNB is AB, and LP for MNB. Ohscal shows similar behaviour to WebKB4. For MNB, the lowest and highest accuracy results are especially close to each other. Accuracy results for MVNB also demonstrate similar pattern except LP and LD smoothing at ts1, ts5, ts10. These results can be seen in Figure 4.2.20 and Figure 4.2.21.

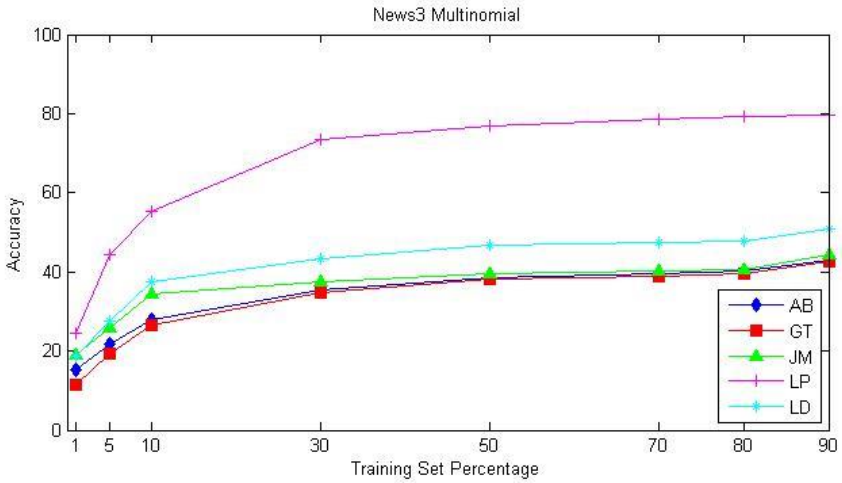

Figure 4.2.22 Accuracy of MNB on News3.

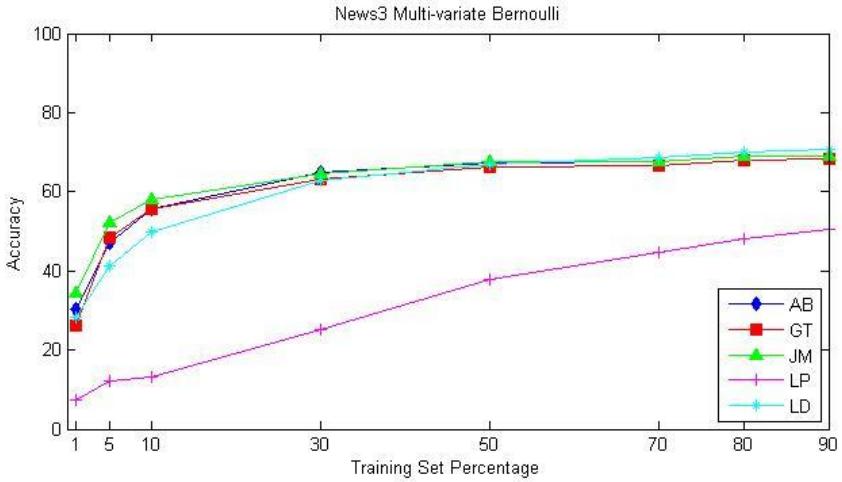

Figure 4.2.23 Accuracy of MVNB on News3.

In News3, LD is the best performing smoothing method at ts90, ts80, ts70 and JM shows higher accuracy performance at remaining training set percentages for MVNB. For MNB, LP is the best performing smoothing method. It is followed by LD and after that JM in some cases. This trend can be clearly seen in Figure 4.2.22 and Figure 4.2.23.

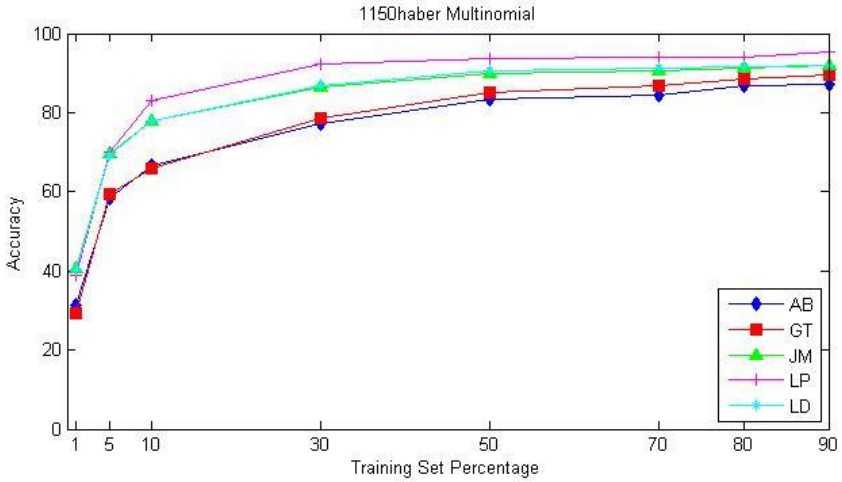

Figure 4.2.24 Accuracy of MNB on 1150haber.

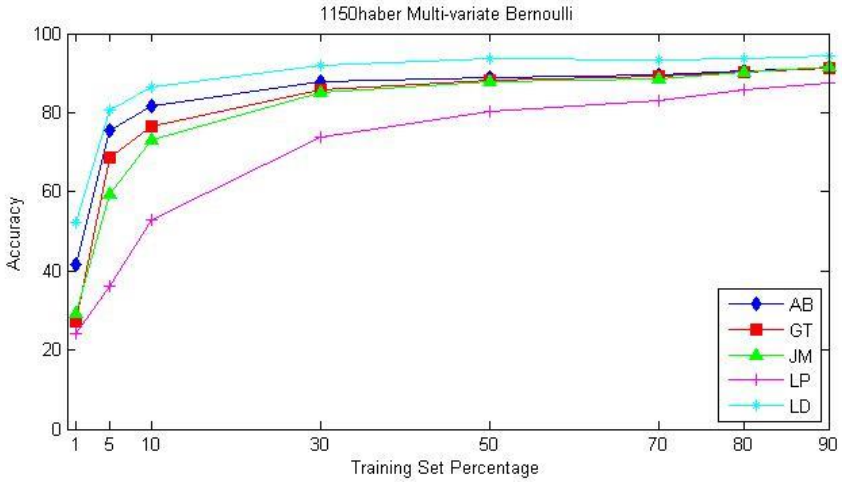

Figure 4.2.25 Accuracy of MVNB on 1150haber.

In any case, Laplace smoothing yields the lowest values in MVNB and highest values in MNB. LP and LD yields 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. Although 1150haber dataset is collected from Turkish news, it demonstrates similar behavior with 20Newsgroups dataset. These results can be observed in Figure 4.2.24 and Figure 4.2.25.

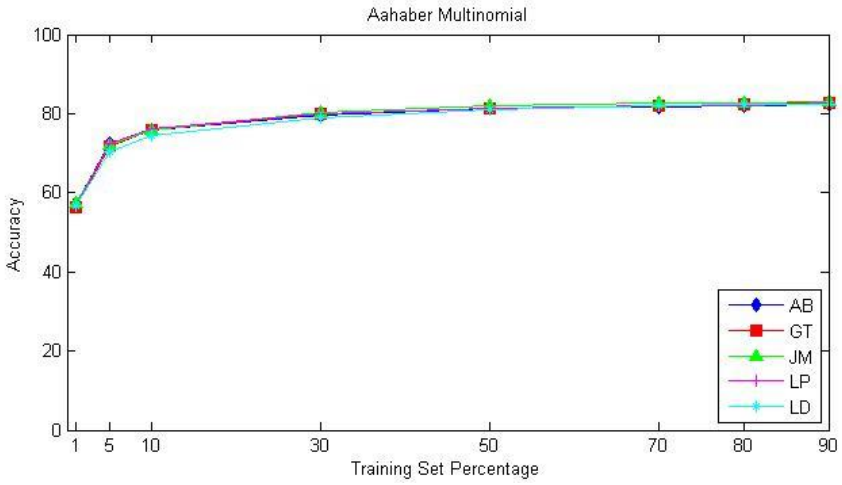

Figure 4.2.26 Accuracy of MNB on Aahaber.

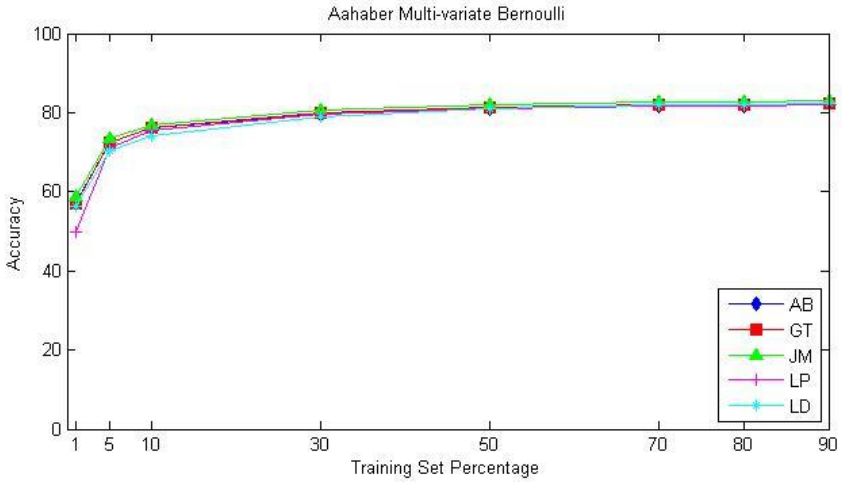

Figure 4.2.27 Accuracy of MVNB on Aahaber.

In Aahaber, all smoothing method accuracies are close to each other both MNB and MVNB event models. Moreover, JM has slightly higher accuracy than the others but the difference is not statistically significant. This trend can be clearly seen in Figure 4.2.26 and Figure 4.2.27.

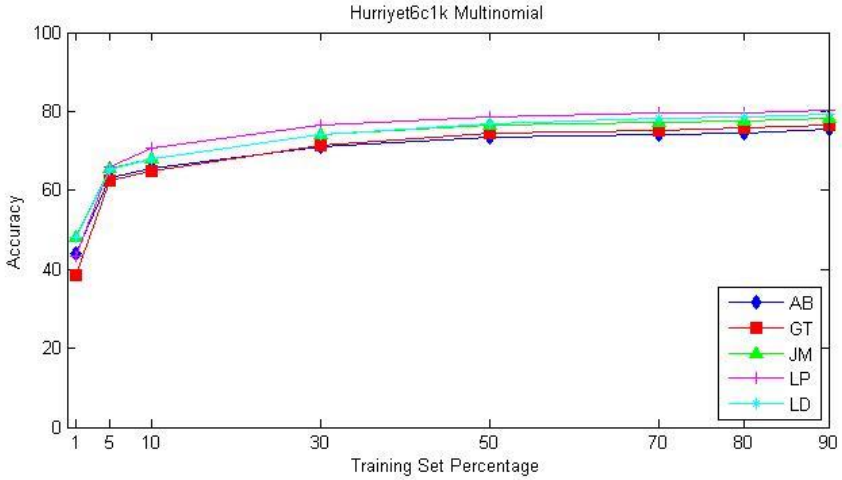

Figure 4.2.28 Accuracy of MNB on Hurriyet6c1k.

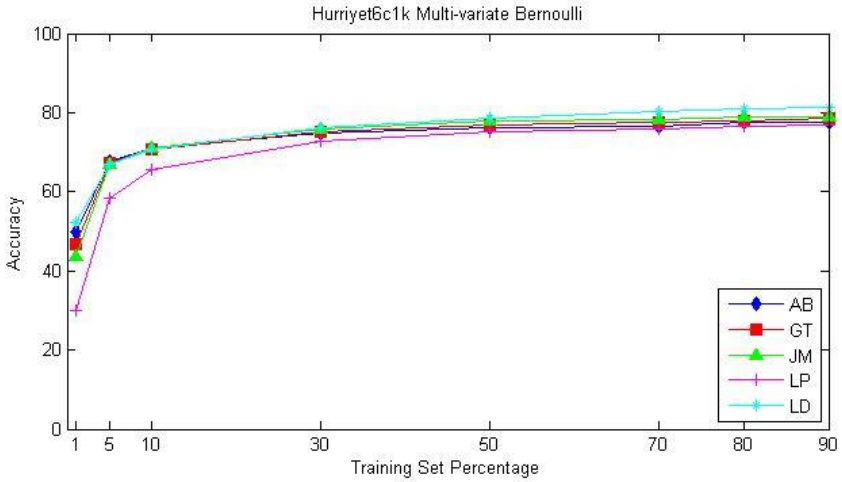

Figure 4.2.29 Accuracy of MVNB on Hurriyet6c1k.

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 close to each other similarly, the highest and lowest accuracies is more distinctive here. This pattern can be observed in Figure 4.2.28 and Figure 4.2.29.

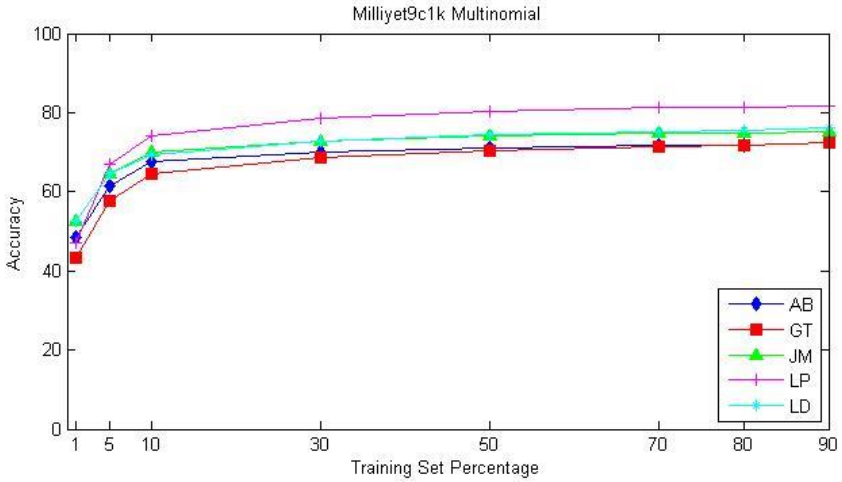

Figure 4.2.30 Accuracy of MNB on Milliyet9c1k.

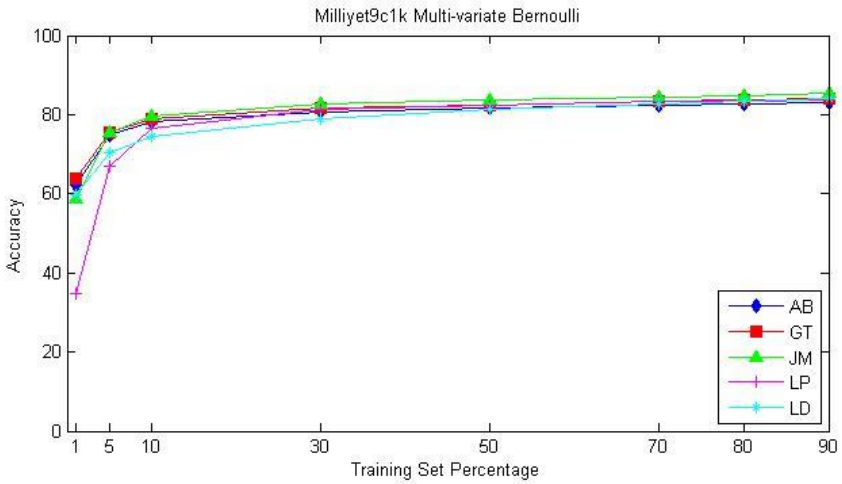

Figure 4.2.31 Accuracy of MVNB on Milliyet9c1k.

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 percentages. 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. Figure 4.2.30 and Figure 4.2.31 summarize these results in detail.

We carry out our experiments with term weighting methods on all datasets. These are binary, tf, and tfidf weighting. Binary and tf weighting are known as MVNB and MNB, respectively. Tfidf weighting is combined term frequency and inverse document frequency in order to produce a composite weight for each term in each document. Tfidf weighting method with LP smoothing yields the highest values among the other term weighting methods. It is followed by binary weighting with LD smoothing and after that by tf weighting with Laplace smoothing on 20News-18828.

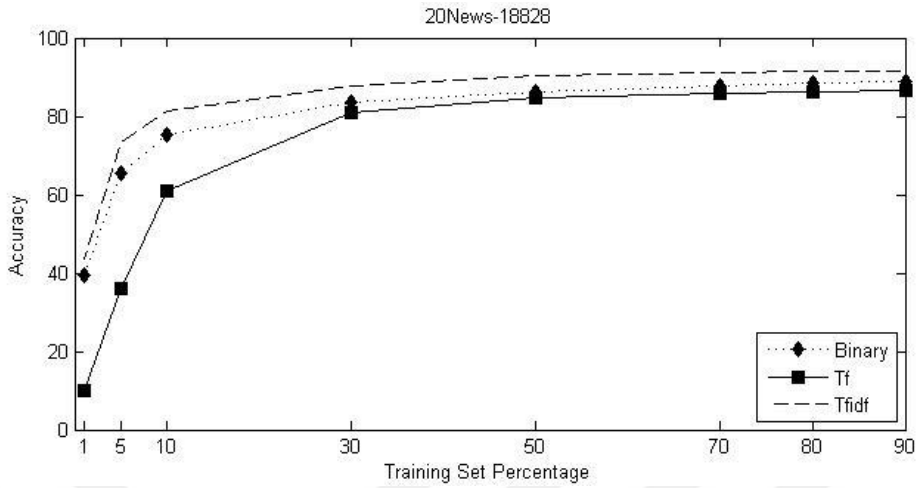

Figure 4.2.32 Accuracy of term weighting methods on 20News-18828.

Table 4.2.11 Accuracies of tfidf weighting on 20News-18828.

20NEWS-18828		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*LP*	76,684	75,885	74,476	71,389	65,245	50,655	36,771	17,633
MNB	*LP*	**91,692**	**91,487**	**91,025**	**90,192**	**87,737**	**81,276**	**73,576**	**43,664**
SVM	*Linear*	91,650	91,498	90,730	89,521	86,907	76,819	65,385	26,616

Previous experiments rely on tf and binary weighting. That is, we take into account either the count of the term or terms' occurrence or non-occurrence briefly. This time, the combination of term frequency and inverse document frequency information is applied on datasets. Table 4.2.11 demonstrates tfidf weighting results on Naïve Bayes event models and SVM. We observe that there is no significant difference between tf and tfidf weighting for MVNB with Laplace smoothing.

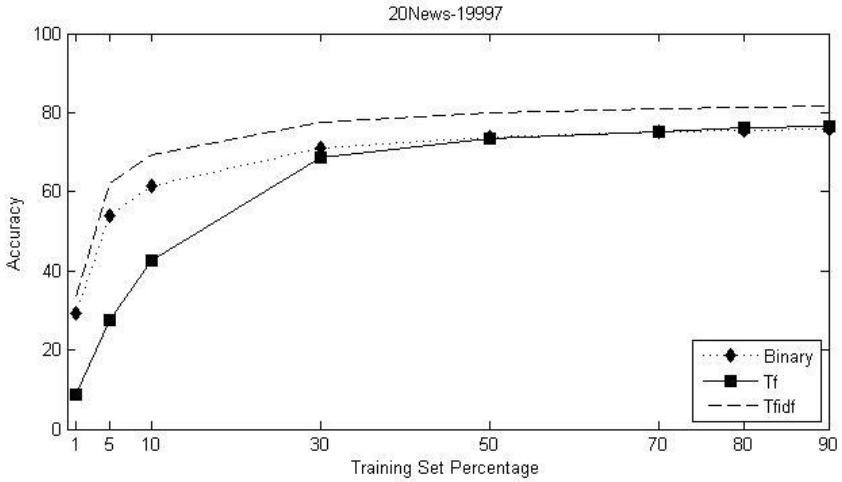

Figure 4.2.33 Accuracy of term weighting methods on 20News-19997.

Table 4.2.12 Accuracies of tfidf weighting on 20News-19997.

20NEWS-19997		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*LP*	63,760	63,908	62,402	60,619	56,759	44,559	35,529	15,993
MNB	*LP*	**81,520**	**81,288**	**80,978**	**79,941**	**77,417**	**69,398**	**62,107**	**33,237**
SVM	*Linear*	64,430	63,140	62,488	59,379	55,737	47,301	40,123	15,846

On the other hand, tfidf weighting of MNB with Laplace smoothing and SVM surpass tf weighting of them by at least 4% and 7% in all training set percentages, respectively. Similarly, tfidf accuracy of MNB with Laplace and SVM outperform tf accuracy of them by at least 5% and 2%, respectively on 20News-19997.

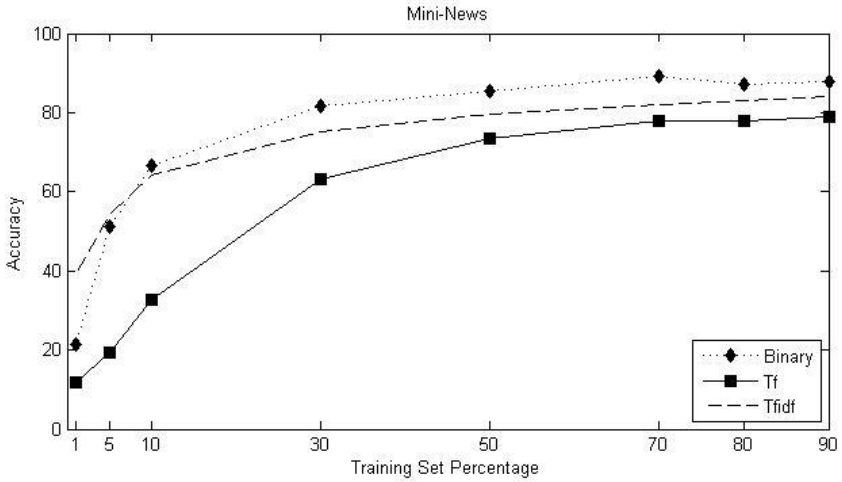

Figure 4.2.34 Accuracy of term weighting methods on Mini-News.

Table 4.2.13 Accuracies of tfidf weighting on Mini-News.

MINI-NEWS		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*LP*	77,550	77,775	74,533	70,960	61,971	39,011	24,821	18,607
MNB	*LP*	84,000	82,925	82,067	79,640	75,271	**64,111**	**54,242**	**39,102**
SVM	*Linear*	**93,200**	**92,400**	**92,017**	**90,720**	**84,114**	55,439	31,242	14,036

For Mini-News, the pattern can be observed in a way similar to the others. That is, tfidf MNB and SVM again exceed tf version of them by approximately at least 5% and 9% in all training set percentages, respectively.

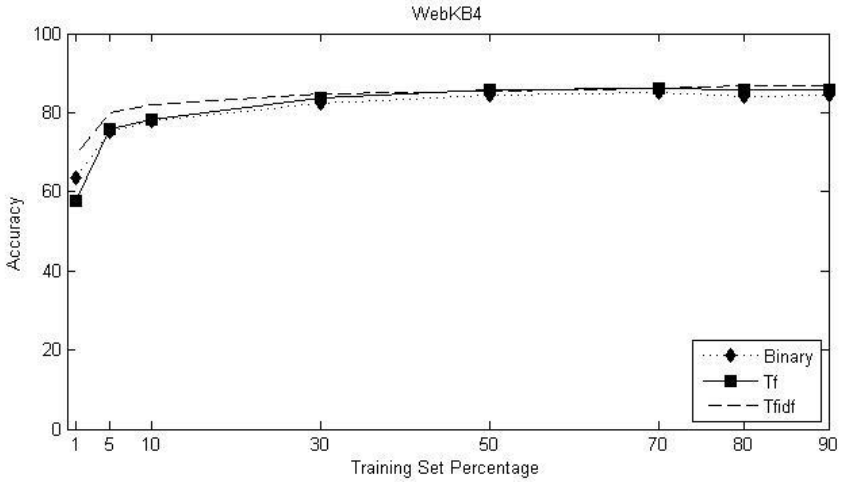

Figure 4.2.35 Accuracy of term weighting methods on WebKB4.

Table 4.2.14 Accuracies of tfidf weighting on WebKB4.

WEBKB4		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*LP*	79,858	79,298	78,550	76,200	70,340	55,619	45,170	39,098
MNB	*LP*	86,777	86,742	86,236	85,571	84,745	**82,116**	**79,905**	**69,449**
SVM	*Linear*	**91,161**	**91,153**	**90,967**	**89,686**	**88,432**	80,283	72,865	49,764

In WebKB4, tfidf accuracy of MNB and SVM has slightly higher accuracy than tf accuracy of them for training set percentages 90, 80, 70, 50 and 30 yet the difference is not statistically significant. In remaining training set percentages, MNB and SVM outperform tf version of them by approximately 4% for both at ts10, 4% and 6% for MNB and SVM at ts5, 12% and 10% for MNB and SVM, respectively. We also apply tfidf weighting on Turkish datasets in order to sight impacts of it.

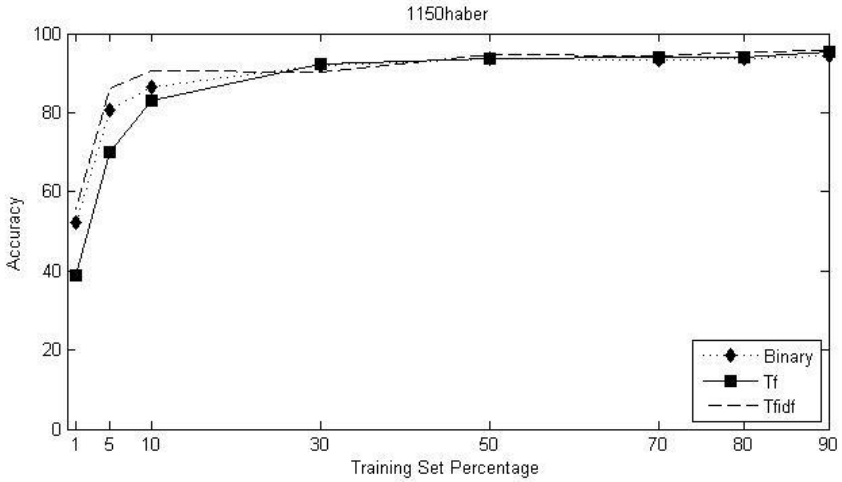

Figure 4.2.36 Accuracy of term weighting methods on 1150haber.

Table 4.2.15. Accuracies of tfidf weighting on 1150haber.

1150HABER		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*LP*	83,913	85,174	83,014	79,583	74,236	57,420	41,306	20,904
MNB	*LP*	**95,826**	**95,304**	**94,493**	**94,574**	**93,565**	**90,696**	**86,046**	**55,570**
SVM	*Linear*	92,609	92,522	92,145	89,670	84,807	72,377	52,009	23,465

In 1150haber and Aahaber, there is not seen significant increases of tfidf accuracy results than the performance of the tfidf results of English datasets at high training set levels.

The accuracy difference between tfidf MNB and tf MNB is approximately 1% for Aahaber in all training set percentages and for 1150haber at high training set levels. At the remaining training set percentages, the accuracy difference at ts10 increases to about 7%, at ts5 and ts1 16% on 1150haber. Tfidf SVM accuracy results are close to tf SVM results which has slightly lower (about 2%) on 1150haber and Aahaber datasets. These results can be seen in Figure 4.2.36, Figure 4.2.37, Table 4.2.15 and Table 4.2.16.

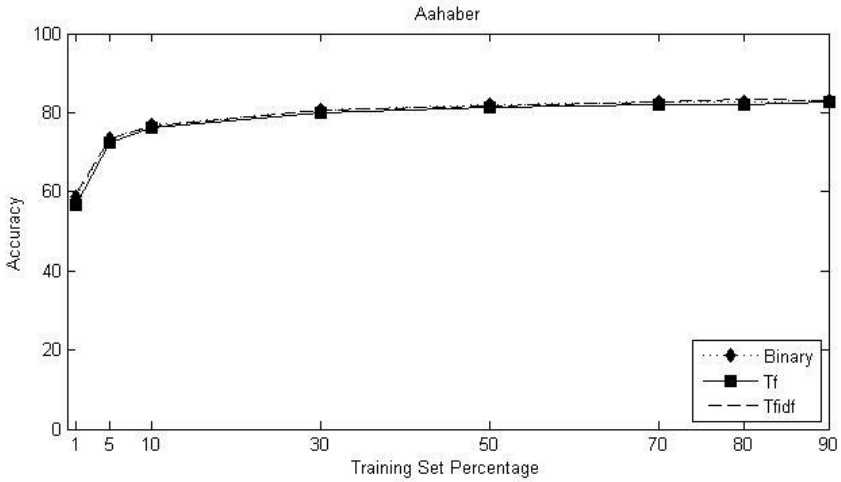

Figure 4.2.37 Accuracy of term weighting methods on Aahaber.

Table 4.2.16 Accuracies of tfidf weighting on Aahaber.

AAHABER		Training Set Percentage							
	Method	**90**	**80**	**70**	**50**	**30**	**10**	**5**	**1**
MVNB	*LP*	81,990	82,055	81,593	80,817	79,534	75,555	71,007	50,148
MNB	*LP*	**82,900**	**83,263**	**82,752**	**81,753**	**80,549**	**76,633**	**73,345**	**59,557**
SVM	*Linear*	80,885	80,888	80,897	80,189	79,189	75,706	71,696	53,768

In Hurriyet6c1k and Milliyet9c1k, tfidf accuracies exceeds tf results of both MNB with LP and SVM approximately by at %3 in some training set percentages. However, tfidf results of MNB and SVM statistically significant outperform tf versions of them by about at least 5% increase in the accuracy.

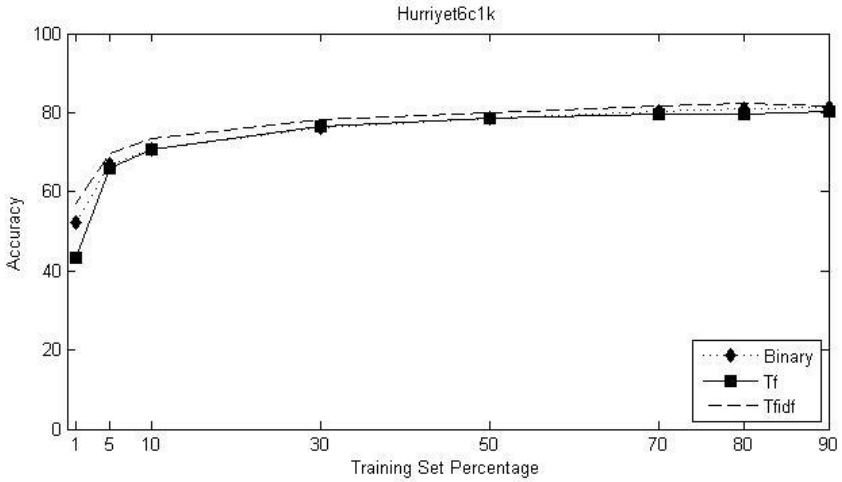

Figure 4.2.38 Accuracy of term weighting methods on Hurriyet6c1k.

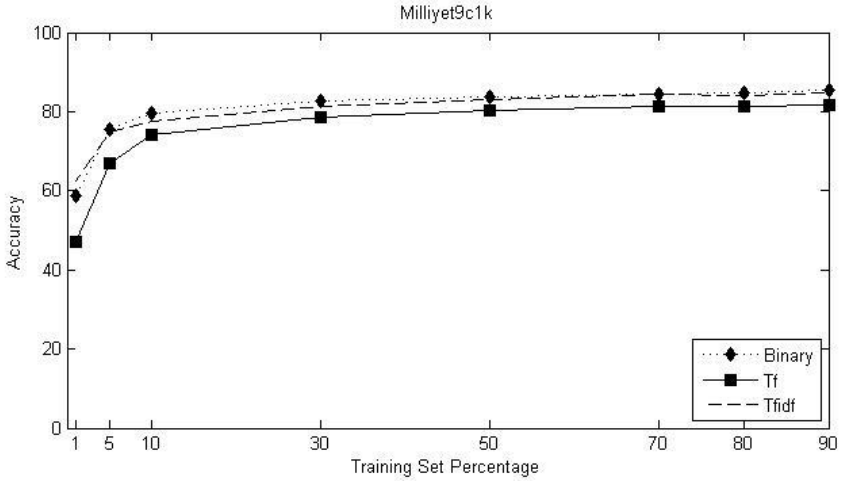

Figure 4.2.39 Accuracy of term weighting methods on Milliyet9c1k.

Table 4.2.17 Accuracies of tfidf weighting on Hurriyet6c1k.

HURRIYET6C1K		Training Set Percentage							
	Method	90	80	70	50	30	10	5	1
MVNB	*LP*	76,883	77,142	76,528	74,727	72,462	65,122	56,523	29,532
MNB	*LP*	**81,767**	**82,292**	**81,683**	**79,883**	**78,283**	**73,548**	**69,625**	**56,993**
SVM	*Linear*	80,483	81,442	80,267	79,077	77,238	71,341	64,882	36,380

Table 4.2.18 Accuracies of tfidf weighting on Milliyet9c1k.

MILLIYET9C1K		Training Set Percentage							
	Method	90	80	70	50	30	10	5	1
MVNB	*LP*	84,244	83,189	83,419	82,462	81,030	75,788	68,906	32,670
MNB	*LP*	84,844	83,906	84,319	83,164	81,352	77,473	74,816	**62,567**
SVM	*Linear*	**93,344**	**92,750**	**92,167**	**90,820**	**88,135**	**82,022**	**75,330**	29,139

Table 4.2.19 shows optimized smoothing parameter values. We experiment on the smoothing methods between 0 and 1 values. Values in the table are choosen thereby observing the best results. Table 4.2.20 and 4.2.21 show F-measure and AUC results for 80% training set level respectively. Table 4.2.22, 4.2.23, 4.2.24, and 4.2.25 demonstrate results accuracy results for ts80, ts30, ts10, 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.

Table 4.2.19 Optimized values of NB smoothing methods' parameters.

SMOOTHING METHOD	PARAMETER	VALUE
LP	ε	1.00
AB	δ	1.00
GT	k	1.00
JM	β	0.50
LD	γ	0.10

Table 4.2.20 F-measures of NB event models at 80% training set level.

DATA SET	MVNB	MNB	SVM
20NEWS-18828	**88.45±0.45 *LD***	84.73±1.84 *LP*	*84.33±0.80*
20NEWS-19997	**76.53±1.15 *LD***	74.43±1.56 *LP*	*61.94±0.91*
MINI-NEWS	**87.99±1.92 *GT***	76.38±1.48 *LP*	*83.04±1.71*
WEBKB4	83.49±1.78 *LD*	**85.10±1.44 *LP***	*89.90±1.46*
OHSCAL	72.22±1.50 *AB*	**73.16±1.56 *LP***	*76.07±1.10*
NEWS3	70.46±0.85 *LD*	**79.29±0.62 *LP***	*85.61±0.57*
1150HABER	93.76±1.35 *LD*	**94.04±1.64 *LP***	*89.65±2.62*
AAHABER	**82.61±0.35 *JM***	82.67±0.42 *JM*	*79.73±0.73*
HURRIYET6C1K	**81.06±0.96 *LD***	79.48±0.84 *LP*	*76.51±1.31*
MILLIYET9C1K	**84.40±0.99 *JM***	81.47±1.03 *LP*	*89.40±0.53*

Table 4.2.21 AUC of NB event models at 80% training set level.

DATA SET	MVNB	MNB	SVM
20NEWS-18828	**99.14±0.08 *LD***	98.20±0.17 *LP*	*91.71±0.42*
20NEWS-19997	**97.19±0.13 *LD***	95.86±0.19 *LP*	*79.79±0.44*
MINI-NEWS	**99.11±0.34 *GT***	96.87±0.49 *LP*	*91.07±0.90*
WEBKB4	**96.39±0.48** *LD*	96.05±0.53 *LP*	*93.11±0.65*
OHSCAL	**95.81±0.29** *AB*	95.67±0.30 *LP*	*86.73±0.26*
NEWS3	94.27±0.55 *LD*	**96.58±0.27 *LP***	*92.60±0.42*
1150HABER	**99.12±0.16 *LD***	98.95±0.36 *LP*	*93.53±1.63*
AAHABER	**97.24±0.09 *JM***	97.17±0.08 *JM*	*88.36±0.41*
HURRIYET6C1K	**96.06±0.39 *LD***	95.31±0.47 *LP*	*85.95±0.79*
MILLIYET9C1K	**97.81±0.24 *JM***	97.10±0.10 *LP*	*94.04±0.30*

Table 4.2.22 Accuracies of NB event models at 80% training set level.

DATA SET	MVNB	MNB	SVM
20NEWS-18828	**88.52±0.42 *LD***	86.26±0.36 *LP*	*84.18±0.79*
20NEWS-19997	**75.68±0.62 *LD***	**76.06±0.38 *LP***	*61.84±0.84*
MINI-NEWS	**87.25±1.91 *GT***	77.90±1.36 *LP*	*83.00±1.71*
WEBKB4	**84.10±1.12 *LD***	**85.64±1.17 *LP***	*89.85±0.96*
OHSCAL	**73.51±0.74 *AB***	**74.47±0.69 *LP***	*77.22±0.49*
NEWS3	69.94±0.85 *LD*	**79.43±0.54 *LP***	*85.28±0.56*
1150HABER	**93.74±1.35 *LD***	**94.00±1.64 *LP***	*89.65±2.62*
AAHABER	**82.70±0.33 *JM***	**82.80±0.44 *JM***	*79.62±0.72*
HURRIYET6C1K	**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 4.2.23 Accuracies of NB event models at 30% training set level.

DATA SET	MVNB	MNB	SVM
20NEWS-18828	**83.60±0.46 *LD***	80.83±0.78 *LP*	*74.40±0.49*
20NEWS-19997	**70.99±0.38 *LD***	68.81±0.90 *LP*	*50.37±0.81*
MINI-NEWS	**81.75±1.60 *GT***	63.33±3.28 *LP*	*67.17±1.15*
WEBKB4	**82.37±1.67 *LD***	**83.89±1.79 *LP***	*88.54±0.86*
OHSCAL	**72.24±0.37 *AB***	**73.36±0.42 *LP***	*74.80±0.42*
NEWS3	64.89±0.68 *AB*	**73.33±0.83 *LP***	*81.19±0.47*
1150HABER	**91.86±0.77 *LD***	**92.42±0.56 *LP***	*82.00±1.69*
AAHABER	**80.64±0.16 *JM***	**80.36±0.29 *JM***	*77.18±0.19*
HURRIYET6C1K	**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 4.2.24 Accuracies of NB event models at 10% training set level.

DATA SET	MVNB	MNB	SVM
20NEWS-18828	**75.14±0.72 *LD***	64.88±3.20 *LD*	*58.64±0.82*
20NEWS-19997	**61.35±0.52 *LD***	47.42±3.40 *LD*	*36.46±1.13*
MINI-NEWS	**66.64±1.84 *GT***	34.06±5.42 *LD*	*42.95±1.07*
WEBKB4	**78.40±2.42 AB**	**78.14±3.04 *LP***	*84.24±1.07*
OHSCAL	**69.09±0.40 AB**	**69.87±0.39 *LP***	*70.41±0.55*
NEWS3	**58.00±1.26 *JM***	55.16±1.66 *LP*	*72.06±0.81*
1150HABER	**86.57±1.58 *LD***	83.16±3.67 *LP*	*69.03±3.20*
AAHABER	**76.94± 0.34 *JM***	**76.07±0.39 *LP***	*71.73±0.31*
HURRIYET6C1K	**71.00±0.56 *JM***	70.64±0.68 *LP*	*64.12±1.22*
MILLIYET9C1K	**79.58±0.65 *JM***	74.16±1.15 *LP*	*76.25±0.86*

Table 4.2.25 Accuracies of NB event models at 5% training set level.

DATA SET	MVNB	MNB	SVM
20NEWS-18828	**65.55±1.42 *LD***	49.25±7.27 *LD*	*45.46±0.95*
20NEWS-19997	**53.91±1.31 *LD***	40.47±5.43 *LD*	*27.80±1.17*
MINI-NEWS	**51.16±4.19 *GT***	25.69±4.30 *LD*	*28.62±2.41*
WEBKB4	**75.12±1.05 *LD***	**75.97±1.72 *LP***	*78.81±2.21*
OHSCAL	**66.30±0.83 *JM***	65.92±0.84 *LP*	*65.52±0.89*
NEWS3	**52.22±2.32 *JM***	44.19±2.28 *LP*	*62.01±1.49*
1150HABER	**80.60±3.26 *LD***	70.17±4.71 *LP*	*56.55±4.26*
AAHABER	**73.34±0.36 *JM***	72.37±0.50 *LP*	*66.88±0.71*
HURRIYET6C1K	**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 visible when we have limited amount of labeled training data which usually is the case in real world applications.

4.3 Discussion

Superiority of the multinomial event model for Naive Bayes is a widely accepted assumption in literature as mentioned in related work section. Several studies argue that this is due to the incorporation of frequency information in multinomial model (Rennie et al., 2003). 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 show that when all term frequency information is eliminated from the document vectors the multinomial model performs even better (Schneider, 2004; Metsis et al., 2006). This suggests that the better performance of multinomial model compare to multivariate Bernoulli model may not stem from extra information on term frequencies. This empirical study discusses the validity of these assumptions by empirically testing event models and smoothing methods under scarce labeled data conditions. 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. Mini-Newsgroups dataset is an exception for MVNB since GT performs much better. 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 4.2.23.

Please note that our 20News-18828 accuracy results for MNB in Table 4.2.22 compares to the state of the art results (McCallum and Nigam, 1998; Rennie et al., 2003) (85% vs our 86.3%) and (84.8% vs. our 86.3%), respectively. Our macroF1 results in Table 4.2.20 consistent with the another studies (Eyheramendy et al., 2003; Kim et al., 2006; Peng and Schurmans, 2003) (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, macroF1 results of 20 Newsgroups dataset are also comparable with the literature (Eyheramendy et al., 2003). Moreover, macroF1 results for MNB compare favorably to the literature (Tantuğ, 2010) result on Turkish Aahaber dataset (77.2% vs. our 82.6%) at ts90 level. On the other hand, our SVM accuracy is 84.2% but the well-known study (Rennie et al., 2003) 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 4.2.26 demonstrates results, briefly.

Table 4.2.26 Comparison with the state of the art results

Authors & Year	**MNB**	**MNB-our**	**MVNB-our**	**Details**
McCallum and Nigam, 1998	85.0% **(LP)**	86.3% **(LP)**	88.5% **(LD)**	20newsgroups,ts80,in terms accuracy, no preprocessing
Rennie et al., 2003	84.8% **(LP)**	86.3% **(LP)**	88.5% **(LD)**	20newsgroups,ts80,in terms accuracy, no preprocessing
Eyheramendy et al., 2003	86.0% **(LP)**	84.7% **(LP)**	88.5% **(LD)**	20newsgroups, in terms macroF1
Kim et al., 2006	86.8% **(LP)**	84.7% **(LP)**	88.5% **(LD)**	20newsgroups, in terms macroF1
Peng and Schurmans, 2003	84.4% **(LP)**	84.7% **(LP)**	88.5% **(LD)**	20newsgroups, in terms macroF1
Tantuğ, 2010	77.2% **(GT)**	82.7% **(JM)**	82.6% **(JM)**	Aahaber,macroF1,ts90 vs our ts80,unigram

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.

We specifically choose three variants of 20 Newsgroups dataset in order to show that algorithms can demonstrate indeed very different behaviours on them. In other words these datasets are not alternatives to each other. It is important to note this since, although not widespread, there are studies in the literature which report or compare results between different variants.

As a future work, we will analyze the reasons behind the better performance of MVNB with appropriate smoothing methods in more details. Several studies (Rennie et al., 2003; Kolcz and Yih, 2007; Eyheramendy et al., 2003) emphasize the importance of document-length normalization in improving Naive Bayes performance. Consequently, we would like to analyze event models by incorporating document length normalization in the future.

REFERENCES

1. McCallum, A. and Nigam, K., (1998), "A Comparison of Event Models for Naive Bayes Text Classification", *AAAI-98 Workshop on Learning for Text Categorization*, Winconsin, 41-48.
2. Schneider, K., M., (2004), "On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification", *4th International Conference on Advances in Natural Language Processing*, Alacant, Spain, 474–485.
3. Metsis, V., Androutsopoulos, I., Paliouras, G., (2006), "Spam Filtering with Naive Bayes – Which Naive Bayes?", *3rd Conference on Email and Anti-Spam*, California, USA.
4. Rennie, J., D., M., Shih, L., Teevan, J., Karger, D., R., (2003), "Tackling the Poor Assumptions of Naive Bayes Text Classifiers", *20th International Conference on Machine Learning*, Washington, USA, 616-623.
5. Juan, A., and Ney, H., (2002), "Reversing and Smoothing the Multinomial Naive Bayes Text Classifier", *2nd International Workshop on Pattern Recognition in Information Systems*, Alacant, Spain, 200-212.
6. Kolcz, A., and Yih, W., (2007), "Raising the Baseline for High-Precision Text Classifiers", *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, California, USA, 400-409.
7. Eyheramendy, S., Lewis, D., D., and Madigan, D., (2003), "On the Naive Bayes Model for Text Categorization", *9th International Workshop on Artificial Intelligence and Statistics*, Key West, Florida, 332-339.
8. Peng, F., Schuurmans, D., and Wang, S., (2004), "Augmenting Naive Bayes Classifiers with Statistical Language Models", *Information Retrieval*, 7, 317–345 (2004).

9. Nigam, K., Mccallum, A., K., Thrun, S., and Mitchell, T., (2000), "Text Classification from Labeled and Unlabeled Documents using EM", *17th International Conference on Machine Learning*, California, USA, 103-134.
10. Su, J., Sayyad-Shirabad, J., and Matwin, S., (2011), "Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes", *28th International Conference on Machine Learning*, Washington, USA, 97-105.
11. Kim, S-B., Han, K-S., Rim, H-C., and Myaeng, S. H., (2006), "Some Effective Techniques for Naïve Bayes Text Classification", *IEEE Transactions on Knowledge and Data Engineering*, 8(11) 1457-1465.
12. Vilar, D., Ney, H., Juan, A., and Vidal, E., (2004), "Effect of Feature Smoothing Methods in Text Classification Tasks", *4th International Workshop Pattern Recognition in Information Systems*, Porto, Portugal, 108-117.
13. Ney, H., Essen, U., and Kneser, R., (1994), "On structuring probabilistic dependencies in stochastic language Modeling", *Computer Speech and Language*, 8(1) 1–28.
14. Katz, S., M., (1987), "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", *IEEE Transactions on Acoustics Speech and Signal Processing*, 35(3) 400–401.
15. Witten, I., and Bell, T., (1991), "The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression", *IEEE Transactions on Information Theory*, 37(4) 1085-1094.
16. He, F., and Ding, X., (2007), "Improving Naive Bayes Text Classifier using Smoothing Methods", *29th European Conference on Information Retrieval Research*, Springer, 703.

17. Yuan, Q., Cong, G., and Thalmann, N., M., (2012), "Enhancing Naïve Bayes with Various Smoothing Methods for Short Text Classification", *21st World Wide Web Conference*, Lyon, France.
18. Joachims, T., (1998), "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", *10th European Conference on Machine Learning*, Chemnitz, Germany, 137-142.
19. Burges, C., J., C., (1998), "A Tutorial on Support Vector Machines for Pattern Recognition", *3rd International Conference on Knowledge Discovery and Data Mining*, New York, USA, 121-167.
20. Yang, Y., and Liu, X., (1999), "A Re-examination of Text Categorization Methods", *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 42-49.
21. Manning, C., and Schütze, H., (1999), "Foundations of Statistical Natural Language Processing", MIT Press.
22. Gale, W., A., (1995), "Good-Turing Smoothing Without Tears", *Journal of Quantitative Linguistics*, 2, 217-237.
23. Chen, S., F., and Goodman, J., (1998), "An Emprical Study of Smoothing Techniques for Language Modeling", Technical Report, Harvard University Center for Research in Computing Technology.
24. Chen, S., F., and Rosenfeld, R., (2000), "A Survey of Smoothing Techniques for Maximum Entropy Models", *IEEE Transactions on Speech and Audio Processing*, 8(1) 37-50.

25. Peng, F., Schuurmans, D., and Wang, S., (2003), "Language and Task Independent Text Categorization with Simple Language Models", *Human Language Technology Conference*, Edmonton, Canada, 110-117.

26. Peng, F., and Schurmans, D., (2003), "Combining Naïve Bayes and n-Gram Language Models for Text Classification", *25th European Conference on Information Retrieval Research*, Pisa, Italy, 335-350.

27. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I., H., (2009), "The WEKA Data Mining Software An Update", *ACM SIGKDD Explorations Newsletter*, 11(1).

28. Forman, G., and Cohen, I., (2004), "Learning from Little: Comparison of Classifiers Given Little Training", *15th European Conference on Machine Learning*, Pisa, Italy.

29. Amasyalı, M., F., and Beken, A., (2009), "Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Sınıflandırmada Kullanılması", *IEEE 17th Signal Processing and Communications Applications Conference*, Antalya, Turkey.

30. Tantuğ, A., C., (2010), "Document Categorization with Modified Statistical Language Models for Agglutinative Languages*",International Journal of Computational Intelligence Systems*, 3(5) 632-645.

31. Manning, C., D., Raghavan, P., and Schütze, H., (2009), "An Introduction to Information Retrieval, in Text Classification and Naïve Bayes", chapter 13, 263-270, Cambridge UP, Cambridge, England.

CV

Place and Date of Birth		Erzurum, 10.09.1985
High School	2001-2003	İzmir Anatolian High School
B.S Degree	2003-	Aegean University
		Physics Department
	2005-2008	Doğuş University
		Computer Engineering Department
M.S Degree	2010-	Doğuş University
		Computer Engineering Department

Work Experiences

	2009-2011	Denizbank
		Datawarehouse Department, CRM and Data Mining Team
		Software Engineer
	2011-	Doğuş University Research Asistant
		Computer Engineering Department

Publications

Poyraz, M., Kilimci, Z. H., Ganiz, M. C., (2012), "A Novel Semantic Smoothing Method Based on Higher Order Paths for Text Classification", *IEEE International Conference on Data Mining (ICDM 2012)*, December 10-13, 2012, Brussels, Belgium.

Poyraz, M., Ganiz, M. C., Akyokuş, S., Gorener, B., Kilimci, Z. H., (2012), "Exploiting Turkish Wikipedia as a Semantic Resource for Text Classification", *INISTA 2012*, July 2-4, 2012, Trabzon, Turkey.