Text Classification using Semi-Supervised Learning

Anusha Manur
amanur@ncsu.edu

Kartik Shah
kcshah@ncsu.edu

Shailaja Mallick
smallic@ncsu.edu

1 Background

With the massive volume of data available online such as news feed, email, medical records; automatically classifying text documents is a problem. Such text classification usually requires large amount of labeled data to train any model which affects the accuracy. Hence, to improve the accuracy, unlabeled documents can be used to augment the labeled ones. We focus on semi-supervised algorithms to classify news articles into 20 newsgroups. Semi-supervised algorithms learn from a small amount of labeled data and a large pool of unlabeled data. Semi-supervised learning algorithms typically make an assumption about the data distribution which enables learning. Also, the algorithm assumes that either the points that are close to each other are likely to share a label or the data tend to form clusters and points in same cluster share the same label.

Text/document classification involves a large number of attributes such as words, which can vary from hundreds to thousands, for example, in case of news articles. Prior work shows that based on presence or absence of words, *multi-variate Bernoulli* model in the field of Bayesian networks have been applied for text classification [4, 3]. Researchers also used number of occurrences of each word in the document in classifying the text, known as the *multinomial Naive Bayes* model [5, 7]. In this model, feature vectors represent frequencies with which certain events have been generated by a multinomial distribution and it explicitly models the word counts and adjusts the underlying calculations to deal within. A combination of Expectation-Maximization algorithm and Naive Bayes classifier have also been used to improve the accuracy of the model by incorporating a small number of labeled data and a large pool of unlabeled data [8]. Apart from existing supervised algorithms (Support Vector machines (SVM)), various other semi-supervised models have also been used for text classification such as self-learning, graph based method, semi-supervised support vector machines (S3VM) [10, 9, 1]. Recent works show that by using unlabeled data, sequence learning can also be improved with recurrent networks [2]. Based on such prior work, we attempt to apply these techniques to our data set and compare the performance of the model.

As known, text classification is assigning categories/labels to text based on its context and the range of text classification varies from spam detection, topic labeling, sentiment analysis etc. We, in this project, focus on the topic labeling aspect of text classification and apply supervised and semi-supervised algorithms to compare the accuracy of the model. Apart from labeled data, we attempt to use unlabeled data to enhance model performance. We explore different semi-supervised techniques while considering the assumptions about the data.

The data set used in this project is the Twenty Newsgroups, which contains approx., 1000 text articles posted to each of 20 online newsgroups, for a total of 18846 articles. The "label" of each article is the category (newsgroups) to which each of the article belongs to. The newsgroups (labels) are hierarchically organized such as sports, hockey, religion, politics and each record in the data set is a text file. The data set has metadata such as subject, summary, keywords etc., and the body which contains the actual content of the article.

Based on this data set and the techniques available to us, there are few questions that needs to be answered.

1. How to use largely available unlabelled data to augment model performance and establish a pipeline?

2. Semi-supervised learning algorithms typically make an assumption about data distribution which enables learning. Is this suitable to our data set?
3. For our data set, how do semi-supervised techniques perform when compared to supervised approaches?

By applying supervised and semi-supervised for both labeled and unlabeled data, we, in this paper, try to answer these questions.

The rest of the paper is structured as follows. Section 2 details the methods that were applied to classify the data set. In Section 3, we discuss the set up of our experiment including the detail of the data set. The results of the model are discussed in Section 4 and finally, in Section 5 we present an overview of the findings and leave our concluding remarks.

2 Methods

In this section, we present the methods that are applied to specifically evaluate the model performance by using more of unlabeled data and less of labeled data. We also outline the data pre-processing steps that are required before applying different algorithms.

2.1 Data pre-processing

As mentioned in Section 1, we have used the Twenty Newsgroups data set which contains text data predominantly, as shown in Figure 1. The data needs to be processed before we feed it into our

```
From: lerxst@wam.umd.edu (where's my thing)
Subject: WHAT car is this!?
Nntp-Posting-Host: rac3.wam.umd.edu
Organization: University of Maryland, College Park
Lines: 15

 I was wondering if anyone out there could enlighten me on this car I saw
the other day. It was a 2-door sports car, looked to be from the late 60s/
early 70s. It was called a Bricklin. The doors were really small. In addition,
the front bumper was separate from the rest of the body. This is
all I know. If anyone can tellme a model name, engine specs, years
of production, where this car is made, history, or whatever info you
have on this funky looking car, please e-mail.

Thanks,
- IL
   ---- brought to you by your neighborhood Lerxst ----
```

Figure 1: Example of body of a article

machine learning algorithm in order to label the text based on its content. We apply the following pre-processing steps,

1. **Removing metadata** - We consider only the body of the article for text classification and hence remove the metadata from each article.
2. **Tokenization** - Next we segment the text into words (tokens). It is basically splitting group of sentences into small chunks of words which is done using python's NLTK library.
3. **Stemming** - The next step is to remove the end or start of the word since they are unlikely to add any value while training the model. It considers a list of basic prefixes and suffixes to truncate it. For example, "consigned" or "consigning" words are converted to "consign".
4. **Lemmatization** - We also apply lemmatization technique which considers the morphological analysis of the word, i.e., converts the word into its canonical form (lemma). It sees the position and part of speech before truncating anything. For example, the words "entitling" or "entitled" to "entitle". This method is slower compared to Stemming but it is more accurate since it considers context of word rather than just removal of the prefix/suffix.
5. **Vectorization** - We use term frequency-inverse document frequency (Tf-idf). This method of tokenization uses word importance in a document and then proportionally increases its weightage based on the number of times it appears in the article.

2.2 Proposed Methods

After cleaning up the data, we now apply supervised and semi-supervised algorithms to train our data set and then use to test the accuracy of the model. We start with a baseline model so as to compare the results of accuracy and improve the performance. The main aim is to achieve higher accuracy by utilizing larger number of unlabeled data. We use *Multinomial Naive Bayes* algorithm as the baseline model, which is a specific instance of Naive Bayes classifier and is typically used for document classification. A simple naive Bayes would model a document as the presence and absence of particular words, multinomial naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal within.

In the next step, we apply various semi-supervised algorithms to train our data set, which are as follows,

1. **Self Learning** - It is a technique where pseudo labeled data selected based on a confidence threshold is used along with labeled data to iteratively train a classifier.
2. **Expectation-Maximization (EM) with NB** - By using unlabeled data, this algorithm iteratively assigns label based on conditional probability and after estimating, maximises the parameter values.
3. **Graph based** - It is a representation of heterogeneous data and is dependent on the assumption that the input information has some underlying structure which can help to classify the unlabeled data.
4. **S3VM** - It is a soft margin SVM which means, when the data is not linearly separable, we allow it to cross the margins with some penalty. It maximizes the margin of labeled and unlabeled points and is also helpful to solve convex optimizations problems.

3 Experiment Set Up

In this section, we describe the details of the experiment and how various methods are applied. We also provide detailed outline of the features/attributes of the data set.

3.1 Data set

The data set used in this project is the Twenty Newsgroups, which contains approx., 1000 text articles posted to each of 20 online newsgroups, for a total of 18846 articles. The "label" of each article is the category (newsgroups) to which each of the article belongs to. The newsgroups (labels) are hierarchically organized such as sports, hockey, religion, politics (as shown in Figure 2) and each record in the data set is a text file. The labels are numbered as 1, 2, . . . and so on, which represents to which category each word belongs to, when the words are run against the chosen machine learning model. The data set has metadata such as subject, summary, keywords etc., and the body which contains the actual content of the article.

```
Total Newsgroups : ['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sy
s.mac.hardware', 'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport
.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns', '
talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']
```

Figure 2: Total Newsgroup Categories

3.2 Data Pre-processing

In the first step, we divide the 18846 article sets into 80% training and 20% testing data set. In order to have unlabeled data, we mask different ratios of labeled data to attain unlabeled data set. We use 5653 number of labeled data and 9423 number of unlabeled data for determining the training accuracy. Later we also vary the number of unlabeled data to observe how the accuracy changes with the change in unlabeled data.

After having the training data set, we remove the headers, footers, quotes and all the metadata. Subsequently, we tokenize the sentences into words and apply stemming technique using the *Porter*

Stemming algorithm. This algorithm removes the common morphological and inflexional endings from the words and normalizes it. We take a bucket list of stop words and remove any unwanted words from the tokenized list. In the end of this process, we apply lemmatizing method to group together the different inflected forms of a word so that all the items are analyzed as a single item.

3.3 Supervised Methods

Apart from the baseline algorithm, we apply couple of other supervised learning techniques such as SVM, multilayer perceptron (MLP), KNN to have a comparison of its accuracy in predicting labels/categories of news groups.

3.4 Semi-supervised Methods

Next, we apply various semi-supervised methods such as Self-learning, EM-NB, Graph-based and S3VM.

3.4.1 Self-learning

Self learning is a fast, intuitive and a straight-forward technique.In this technique, we initially train a chosen machine learning classifier as a base model using the available labeled data. The trained model is then used to label the unlabeled data. Then, the confidently labeled pseudo labeled data points are used along with the labeled dataset to re-train the model. They are selected by setting a confidence threshold. This is done iteratively until the model converges after a given number of iterations. One disadvantage of this technique is that any error in prediction of the labels for the unlabeled data can reinforce itself through the many iterations during training.

We use a multinomial naive bayes as our base model even though SVM results in higher accuracy, as the time required for self learning along with SVM is very high. Our self learning model has a confidence threshold of 0.8 and converges after 200 iterations.

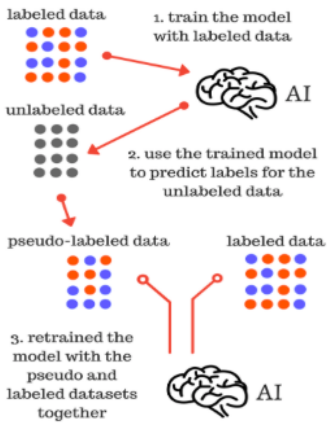

Figure 3: Self-learning

3.4.2 EM-NB

Given a set of parameters, $\theta^0, \theta^1, \theta^2, \ldots, \theta^T$, the initial parameter, θ^0, is chosen in random. In each iteration, new parameter values, θ^t, is calculated as a function of training set and previous parameter value. For unlabeled data, y being the possible label is determined by conditional probability as given by, $\delta(y|i) = p(y|x^i, \theta^{t-1})$. If the algorithm is applied only to labeled data set, then the possible label is determined by, $\delta(y|i) = 1$ if $y_i = y$ else 0. As the EM algorithm utilises NB classifier, by using unlabeled data, it iteratively assigns label based on conditional probability and after estimating, maximises the parameter values.

3.4.3 Graph-based

In this method, the node represents an article and a weighted edge represents the connection between each node, as shown in Figure 4. Before the model builds a graph, there are few labelled and

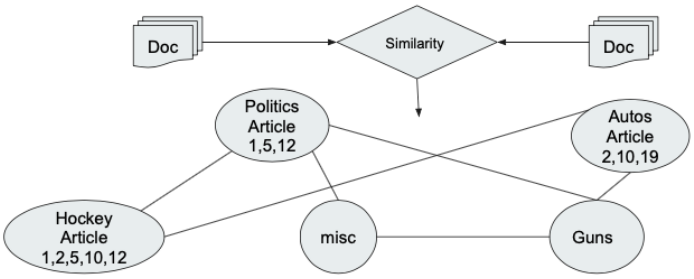

Figure 4: Graph Construction

unlabeled nodes in the initial round. As a first step, we make use of K Nearest neighbors (KNN), an unsupervised method for those training data which had labels. For KNN, we depended on Euclidean distance metrics to form small clusters based on the similarity between each article.

In the next step, for a particular value of K, we used Label propagation technique to iteratively label the unlabeled articles. In this method, we compute a diagonal matrix with the degree of similarity between each article. Finally, until the algorithm converges we propagate the values from the labelled article to the unlabelled ones. In the third step, we use Principal Component Analysis (PCA) to minimize the dimension of the features. This procedure helped to understand the variance and co-variance between the variables and we only use the first seven features which had the highest variance.

At the end of this method, different values of K such as K = 3 and K = 65 are taken to perceive what improvements we get for accuracy. We also use different sizes training data set on different K values and calculate the accuracy. For each value of K, we construct a graph and additionally perform PCA to observe whether we get better accuracy or not. In our final observation, we found that when K = 65 (with PCA) gives us the best results.

3.4.4 S3VM

After doing pre-processing we used the next S3VM to calculate accuracy. In this method we used a simple scikit-learn compatible wrapper for the QN-S3VM code developed by Fabian Gieseke. We passed in different size of testing and training data set. We observed in the end that it has high computational complexity while running on this data set.

4 Results

Based on different techniques of supervised and unsupervised algorithm being applied to our data set, Table 1 shows the training/validation accuracy of each supervised method. Table 2 shows the

Supervised Methods	Validation Accuracy (%)
Multinomial NB	88.05
SVM	88.67
MLP	82.11
KNN	71.07

Table 1: Validation Accuracy with #Labeled data = 5653, #Unlabeled data = 9423

training/validation accuracy of each semi-supervised method. Table 3 shows the testing accuracy of each semi-supervised method when tested on 20% of the data.

Semi-supervised Methods	Validation Accuracy (%)
Multinomial NB	88.05
Self-Learning	91.48
EM-NB	88.08
Graph-based	55.65

Table 2: Validation Accuracy with #Labeled data = 5653, #Unlabeled data = 9423

Semi-supervised Methods	Testing Accuracy (%)
Multinomial NB	77.13
Self-Learning	86.84
EM-NB	77.2
Graph-based	55.65

Table 3: Testing Accuracy

The results indicate that for semi-supervised method, Self-learning gave the highest accuracy (for both validation and testing) and for supervised method, SVM gave the highest validation accuracy. The accuracy of EM-NB is very close to multinomial NB and we believe the reason being as the EM with NB is based on NB, the prediction of label is very similar and hence the accuracy value is also close to each other.

For *Graph based*, we got the highest accuracy of 55.65% for Labeled data = 5653 and the unlabeled data = 9423, using PCA and K=65. And for labelled data = 1413 and unlabeled data = 2357 with K=65 and using PCA we got the highest accuracy of 43.22%. The algorithm took a long time to process it and finally converged, that was because of the confusion as to which node it should belong to while classifying itself.

With respect to *S3VM*, we were not able to get any results for S3VM because before it converged, the algorithm was getting out of memory error. Since we know it is a NP-hard problem, out of memory indicates that it increases the complexity while classifying itself and therefore in this data set we cannot use S3VM.

We also varied the number of unlabeled data to see how the accuracy changes for Self-learning and EM-NB semi-supervised method along with the baseline method (Multinomial NB). Figure 5 shows that as the number of unlabeled data is less, the accuracy is higher, which is expected as the number of labeled data is higher in such a scenario and vice-versa.

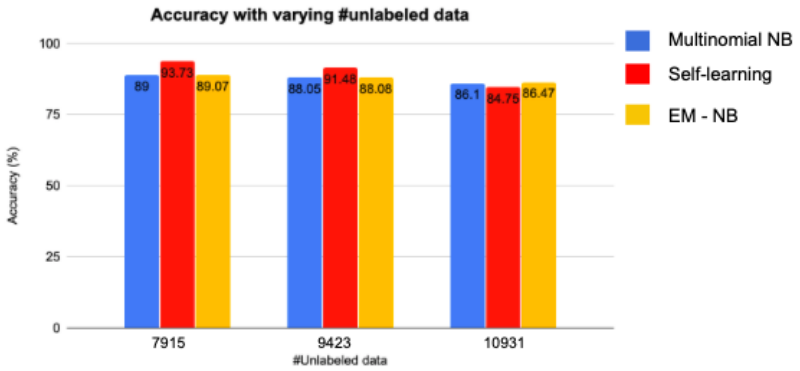

Figure 5: Validation Accuracy with varying number of unlabeled data

5 Conclusion & Future Work

We started with 3 questions on applying semi-supervised learning techniques to text classification. From our results we see that suitable semi-supervised techniques like self learning and EM can be applied to text classification of newsgroups and we see that these techniques perform better than supervised techniques.

We observe that there might be a possibility of fine tuning the hyper-parameters of the supervised techniques to obtain high accuracy, but it does not use unlabeled data to augment performance or adapt to the variation in the newly generated data points like in real-life scenarios. However, semi-supervised techniques use the newly generated unlabeled data to augment model performance and hence can be used to establish a pipeline where the semi-supervised algorithm uses the newly generated unlabeled data points to retrain the model and label them as they become available.

Through this project, we compare supervised and semi-supervised techniques and observe that for a data set with few labeled and a large number of unlabeled data points, semi-supervised techniques can be used to obtain a high classification accuracy.

We also attempted co-training but did not achieve good results and leave it for future work. Since there is no clear distinction of independent sets of features, we considered content and metadata as the set of independent features. Since metadata is not a strong enough feature, we considered extracting features from the content of the article to be used for co-training as our future work[1].

References

[1] Kristin P Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information processing systems*, pages 368–374, 1999.

[2] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[3] Tom Kalt and WB Croft. A new probabilistic model of text classification and retrieval. Technical report, Technical Report IR-78, University of Massachusetts Center for Intelligent . . . , 1996.

[4] David D Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–50. ACM, 1992.

[5] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[6] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[7] Kamal Nigam, Andrew McCallum, Sebastian Thrun, Tom Mitchell, et al. Learning to classify text from labeled and unlabeled documents. *AAAI/IAAI*, 792:6, 1998.

[8] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[9] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[10] Amar Subramanya and Partha Pratim Talukdar. Graph-based semi-supervised learning algorithms for nlp. In *Tutorial Abstracts of ACL 2012*, pages 6–6. Association for Computational Linguistics, 2012.

[1] The code can be found in the following Github link: `https://github.ncsu.edu/smallic/CSC591-ALL12-TextClassification`