Deep Belief Networks with Feature Selection for Sentiment Classification

Patrawut Ruangkanokmas\textsuperscript{1}, Tirane Achalakul\textsuperscript{2}, and Khajonpong Akkarajitsakul\textsuperscript{3}

\textsuperscript{1,2}Department of Computer Engineering
\textsuperscript{3}Department of Mathematics
King’s Mongkut’s University of Technology Thonburi (KMUTT), Bangkok, Thailand
patrawut.ru@gmail.com, \textsuperscript{2}tirane@cpe.kmutt.ac.th, \textsuperscript{3}khajonpong.akk@mail.kmutt.ac.th

Abstract—Due to the complexity of human languages, most of sentiment classification algorithms are suffered from a huge-scale dimension of vocabularies which are mostly noisy and redundant. Deep Belief Networks (DBN) tackle this problem by learning useful information in input corpus with their several hidden layers. Unfortunately, DBN is a time-consuming and computationally expensive process for large-scale applications. In this paper, a semi-supervised learning algorithm, called Deep Belief Networks with Feature Selection (DBNFS) is developed. Using our chi-squared based feature selection, the complexity of the vocabulary input is decreased since some irrelevant features are filtered which makes the learning phase of DBN more efficient. The experimental results of our proposed DBNFS shows that the proposed DBNFS can achieve higher classification accuracy and can speed up training time compared with others well-known semi-supervised learning algorithms.

Keywords—Chi-squared Feature Selection; Deep Belief Networks; Deep Learning; Feature Selection; Restricted Boltzmann Machine; Semi-supervised Learning; Sentiment Classification;

I. INTRODUCTION

Nowadays, amounts of social media data on the web sites tend to grow dramatically. Individuals and organizations try to extract useful information from these large datasets in order to make better judgments and enhance customer satisfaction. For example, before making a decision whether to purchase a product or service, a customer looks through product reviews expressed by others as recommendations. In the same way, the manufacturer of the product also uses this information for improving the quality of their products or services [1]. However, due to vast amount of data available online, it is an expensive task for people to utilize the data manually. As a result, sentiment classification, which aims at determining whether a sentiment expressed in a document is positive, neutral, or negative, will be helpful and be beneficial in business intelligence applications, recommender systems, and message filtering applications [2].

To construct an accurate sentiment classifier, in the past few years, many researchers have tried to integrate the concept of deep learning with machine learning [3]-[6]. With its power to handle millions of parameters, deep learning can drastically improve a model prediction power. One of the great examples is Recursive Neural Tensor Network trained on Sentiment Treebank proposed by Richard Socher [4]. It can accurately predict sentiment orientation with over 85% correctness. Unfortunately, a large amount of labeled training data is needed in supervised training methods and it is often difficult and time consuming to label the data manually.

Not too long before, a new approach called semi-supervised learning has been proposed. It aims at utilizing the advantage of a huge amount of unlabeled data together with labeled data to construct sentiment classifiers [7]. Many research papers in [5], [6], and [8] claim that semi-supervised deep learning models can avoid the aforementioned issues while they still get competitive performance. Unfortunately, the current deep learning algorithms are computationally expensive for large-scale applications.

Moreover, most of the classification algorithms use a fixed size of numerical feature vectors as inputs rather than use raw variable-length text documents. Thus, it is necessary to convert a corpus of documents into a matrix with one row per document and one column per token (i.e., word) occurring in the corpus. Because of the complexity of human languages, there can be more than ten-thousand dimensions of feature terms while most of them are noisy or redundant. This can lead to the increase in the number of classification errors and the computation time.

To overcome the aforementioned problems, an effective feature selection, which aims at filtering inessential terms occurring in the training set and selecting only meaningful terms, is a must in order to make the learning phase more efficient and more accurate [9]. Forman [10] has proposed an empirical comparison among many feature selection methods. The results show that with the feature selection methods, the performance of the classification algorithms in most situations can be improved since they can reduce the number of dimensions of input data by eliminating noisy features. Thus, we can train a classification model faster, reduce memory consumption, and also get the better result accuracy.

The remainder of this paper is organized as follows. Section II presents the theoretical background of feature selection and semi-supervised deep learning used in our work. The design workflow of our proposed framework is described in Section III and our experiment results and discussion are available in Section IV. Lastly, in Section V, the conclusion of the proposed work and our future direction are presented.
II. LITERATURE REVIEW

In this section, the theoretical background of feature selection and semi-supervised deep learning related to our work is presented.

A. Feature Selection

Feature selection is a process, which aims to simplify the model construction, by selecting a subset of relevant features. It serves two major roles. The first role is to improve the training process of a classifier more efficient by reducing the size of vocabulary input. The second role is to increase the prediction accuracy by filtering inessential terms or noisy features. As a result, a shorter period of training time and also a better model representation can be achieved.

1) Feature Selection Techniques

Basically, feature selection techniques can be organized into three categories [9]: filter, wrapper and embedded techniques. The filter-based technique is used as a pre-processing step prior to a learning algorithm. Features are ranked by some criteria and then selected if their scores are above an appropriately pre-defined threshold. Next, the wrapper technique utilizes a learning algorithm to select and evaluate a subset of features among all features. Finally, the embedded technique performs feature selection as a part of the training process.

Among the three categories, the filter-based technique is the most suitable one since it is simple, fast, and independent from classifiers. With its good scalability, it can efficiently be applied for large-scale applications. Examples of the filter-based technique are Information Gain, $\chi^2$ (Chi-squared), Mutual Information, t-test, F-measure, and etc. Forman [10] has studied among these methods in his comparative research and summarized the Chi-squared filter-based technique as the most effective way to perform the feature reduction in text analysis problems. As all reasons presented so far, $\chi^2$ is selected as our feature selection procedure in the proposed framework.

2) $\chi^2$ (Chi-squared) Feature Selection

A chi-squared test is a common statistical method used to test independence of two events. Particularly, it can be used as the feature selection while the two events are occurrence of the term and occurrence of the class. The higher score on $\chi^2$, the more dependent between the feature term and class, and therefore considered important. Consequently, the top-rank features with most $\chi^2$ scores are chosen as a set of features for the text classification [11].

B. Semi-supervised Deep Learning

Semi-supervised deep learning is a branch of machine learning making use of a small amount of labeled data with a large amount of unlabeled data. A well-known example of deep semi-supervised learning algorithms is Deep Belief Network (DBN). To construct a DBN, we can follow the below steps [3]:

2.1) A DBN is constructed by greedy layer-wise unsupervised learning using a stack of RBMs as building blocks. The learning algorithm makes the effective use of unlabeled data, which provides a significant amount of patterns in input data, to produce better initial weights than random ones. The objective of this phrase is to perform feature representation.

2.2) A DBN is trained according to an exponential loss function using gradient descent based supervised learning. The weights of the model are refined by labeled data. The objective of this phrase is to perform pattern classification.

Figure 1 shows the undirected graphical network of an RBM.

Figure 1. An illustration of an RBM network.

To get a better performance, a stack of restricted Boltzmann machines can be defined as a Deep Belief Network (DBN). To construct a DBN, we can follow the below steps [3]:

2.1) A DBN is constructed by greedy layer-wise unsupervised learning using a stack of RBMs as building blocks. The learning algorithm makes the effective use of unlabeled data, which provides a significant amount of patterns in input data, to produce better initial weights than random ones. The objective of this phrase is to perform feature representation.

2.2) A DBN is trained according to an exponential loss function using gradient descent based supervised learning. The weights of the model are refined by labeled data. The objective of this phrase is to perform pattern classification.

Figure 2 shows the greedy layer-wise unsupervised training process of a DBN with one input layer; i.e., $x$, and three hidden layers; i.e., $h_1$, $h_2$, and $h_3$ from left to right. Lighter-color layers represent already-trained layers while darker color layers are being-trained layers. After the greedy layer-wise unsupervised learning, $h_3(x)$ is the representation of $x$. Then, one output layer is added at the top and labeled reviews are used to refine the weights for better discriminative ability. Figure 3 gives an illustration of a complete DBN.
In text classification, a DBN takes a matrix represented in a bag-of-words model as an input. The first few layers are expected to extract low-level features from the input while the upper ones refine the previously learned feature patterns therefore producing more complex features. Finally, sentiment orientation of reviews is predicted in the output layer whether it is positive or negative.

III. METHODOLOGY

In this section, the main design workflow of our proposed Deep Belief Network with Feature Selection (DBNFS) is presented. Figure 4 is the framework diagram showing the sequence of tasks to perform the sentiment classification. Most of the tasks are similar with other classification approaches, including feature extraction, data partitioning, and model training and testing, respectively. However, in our work, two new tasks which are feature selection and reduction are integrated to the common tasks. The detail of each task is described below.

A. Feature Extraction

In text analysis applications, it is necessary to transform variable-length documents into fixed-size numerical feature vectors which are appropriate with a classification algorithm. The common technique to perform feature extraction of text-based input is the bag-of-words technique, which documents are described by word occurrences while ignoring the relative position information of words. There are two main tasks including feature defining and weighted score computing. In our proposed work, we apply unigrams of token as a “feature” and its term presence as a “weighted score”.

To extract features, the first step is tokenizing documents using white-space and punctuation as separators. Then, all tokens (i.e., words) are lowercased. Next, punctuations, numbers, and one-character words are removed. The unigrams of remaining tokens are formed as a “vector of features”. The binary weighted value of each token in the vector is 1 if it is presented in that document; otherwise, the value is 0. Furthermore, the top 1.5% vocabularies sorted by the number of occurrences of each vocabulary in a descending order are removed since most of them are stop words (e.g., articles) or they may be domain-specific or general-purpose words (e.g., “hotel” in the hotel reviews). Theoretically, stop words can appear in either a positive training dataset or a negative training dataset without any sentiment information. This can increase the number of classification errors because of their sentiment ambiguity.

After this process, a corpus of documents is formed as a matrix of binary values with one row per document and one column per feature or token occurring in the corpus.

B. Data Partitioning

To produce comparable results, we perform experiments using the five widely-used sentiment classification datasets same as the other approaches [5-8], [14-15]. The first dataset is movie reviews (MOV) [2]. The other four datasets are multi-domain sentiment classification datasets from Amazon.com, including books (BOO), DVDs (DVD), electronics (ELE), and kitchen appliances (KIT) [13]. There are 2,000 labeled reviews (i.e., 1,000 positives and 1,000 negatives) in each dataset.

Since the proposed classifier is a semi-supervised learning algorithm which utilizes both unlabeled and labeled data to construct a classifier. Thus, we have to partition a dataset into three sets including unlabeled training set, labeled training set, and labeled testing set. We begin with
dividing each dataset of 2,000 reviews into ten equal-sized folds randomly while we still maintain balanced class distributions in each fold for cross-validation purposes. In each round, we select one fold as a labeled dataset, and then randomly select a half of reviews in that fold as a labeled training dataset and another half as a labeled testing dataset. The remaining nine folds are utilized as an unlabeled dataset.

C. Feature Selection and Reduction

In order to improve the prediction accuracy, we aim to eliminate noisy features that can cause classification errors by performing the feature selection and reduction. In our framework, we apply the chi-squared feature selection to determine which features are most correlated with the sentiment class in order to obtain the highest predictive power.

First, we calculate Chi-squared scores of all the features and then rank them by their scores in a descending order. Then, only top \( n \)-percent features are selected for building our classification model while others are not used for the analysis. The best \( n \) percentage value of each dataset is presented in Table I. Importantly note that we perform the feature selection by using labeled training datasets only because the algorithm is based on a supervised learning method and to avoid an overfitting problem from testing sets.

D. Model Training and Testing

The sentiment classification model used in our framework is based on Deep Belief Networks. First, the learning algorithm performs greedy layer-wise unsupervised learning using unlabeled training reviews. Then, the weights of the model are refined by labeled training reviews with an exponential loss function using gradient descent based supervised learning. After the prediction model is completely constructed, the labeled testing reviews are utilized to perform the model testing. The final classification results are averaged using ten-fold cross validation in terms of accuracy. The learning parameters and the structure of DBN used will be mentioned in the next section.

IV. EXPERIMENTS AND DISCUSSION

In this section, we demonstrate the performance of the proposed DBNFS and also compare the performance in term of accuracy and training time with the other recent algorithms [3], [6], [8], [14] and [15].

A. Experimental Setup

The goal of our experiments is to evaluate performance of our proposed framework against five semi-supervised learning sentiment classifiers in term of accuracy and training time, i.e., semi-supervised spectral learning [14], Transductive SVM (TSVM) [15], Personal/Impersonal Views (PIV) [8], Deep Belief Networks (DBN) [3], and Hybrid Deep Belief Networks (HDBN) [6]. The overview of the classifiers is as follows:

Spectral learning enhances a spectral algorithm which uses information contained in the eigen-vectors of a data affinity (i.e., item-item similarity) matrix to detect structure to perform semi-supervised clustering and classification.

TSVM improves the generalization accuracy of SVMs by using unlabeled data. Similar to SVMs, it learns a large margin hyper-plane classifier using labeled training data but simultaneously force this hyper-plane to be far away from the unlabeled data. Spectral learning and TSVM methods are two baseline methods for semi-supervised sentiment classification.

PIV adopts personal and impersonal views to build a semi-supervised classifier. Personal views consist of those sentences which directly express speakers’ feelings and preferences towards a target object while impersonal views focus on statements towards a target object for evaluation.

DBN is the classical deep learning method proposed recently as aforementioned in Section II.

HDBN is a hybrid of RBM and convolutional RBM (CRBM) deep architecture, the bottom layers are constructed by RBMs, which can reduce the dimension and abstract the information of the input quickly. Then, CRBM is used to abstract more complex information in the upper layers. Finally, the whole network is fine-tuned by to an exponential loss function using gradient descent based supervised learning.
Again, the five sentiment classification datasets used for evaluation are reviews of movies (MOV), books (BOO), DVDs (DVD), electronics (ELE), and kitchen appliances (KIT).

B. Parameter Setup

To fairly compare the results, the learning parameters used in our proposed model are the same as the previous works [5]-[6]. In the pre-training step, we perform greedy layer-wise unsupervised learning with the number of epochs equal to 30 for all the hidden layers and the output layer. For supervised learning step, the number of epochs is set to be 30 while the learning rate is 0.1 with decay rate of 0.9 in each epoch.

However, its architecture is different. The DBN structure used in [5] is 100-100-200-2 for the MOV dataset, which represents the number of neuron units in the three hidden layers are 100, 100, and 200 respectively, and the number of neuron units in the output layer is 2 (positive and negative). For the other four datasets, the DBN structure is 50-50-200-2. The number of units in the input layer is the same as the dimension of input features of each dataset. Instead of using the same mentioned structure, we change our proposed DBN structure to 300-2 which is the same for all the datasets.

C. Experimental Results

Before getting into the detail of the experimental results and their analysis in term of classification accuracy and training time, the effects of feature selection and reduction on the size of input dimension should be demonstrated first.

Table I provides information about the number of features before and after implementing the feature selection and reduction technique. It demonstrates the performance of our proposed dimensional reduction method which can reduce the number of features approximately 74% in average and at most 90% in the ELE dataset. In addition, the first row of Table I presents the value of \( n \) which is the percentage of each dataset used as a threshold to select the most relevant features giving the highest accuracy.

1) Accuracy Performance

The classification accuracy results from ten-fold cross validation of the five sentiment classification datasets with semi-supervised learning approaches are shown in Table II.

![Table II. Average of Classification Accuracy Percentage From 10-Fold Cross Validation With 100 Labeled Reviews](image)

The results of spectral learning and TSVM methods have been reported in [7]. The accuracy results of PIV method excluding the MOV dataset have been found in [8]. The accuracy results of DBN and HDBN techniques have been reported in [5] and [6], respectively. In the last row of the table, the best results of our proposed DBNFS from empirical study are presented.

From Table II, we can observe that DBNFS performs the best accuracy results in the three datasets marked as bold. However, using the KIT and DVD datasets, the accuracy results of DBNFS are slightly less than ones of PIV and HDBN but PIV gets significantly worse results in the other datasets. Furthermore, PIV has much more complex pre-processing steps than our work to deal with personal/impersonal sentences. On the other hand, DBN, HDBN, and DBNFS can perform very well in all the datasets. From the results, the powerful capabilities of deep architecture in sentiment classification are proved.

DBNFS is enhanced from DBN by using the Chi-squared feature selection technique to eliminate inessential features before training a model. By choosing the right features, the accuracy and efficiency of the classification model can be potentially improved. The experimental results show that DBNFS can improve the classification power comparing with the original DBN in all the datasets and better than HDBN for all the datasets but one.

2) Training Time Performance

We can compare training time (i.e., the time is measured from performing feature extraction until complete model is constructed) between DBN and our DBNFS as shown in Figure 5. To fairly compare, we run both of them in the same environment and the same parameter setting except the number of features to train and the network structure. Regarding to the number of features to train, for DBN, we utilize a full set of features without any feature selection and reduction but for DBNFS, we follow the proposed method as described in Section III-C.

In Figure 5, we can observe that DBNFS spends much less training time than DBN in all the datasets. DBNFS can speed up 3 times in average and can speed up almost 5 times when the MOV dataset is utilized. This is because the MOV

![Table I. Amount of Features Comparison Between Before and After Performing Feature Selection and Reduction](image)
dataset has the number of features more than the other four datasets. So using DBNFS, the more number of features are removed which results in the lower training time comparing with DBN while its good accuracy is still maintained.

Particularly, the main factor that significantly improves the DBNFS’s training time is its simpler deep structure by replacing several hidden layers but adding our proposed feature selection method.

According to the experimental results, we can demonstrate that our proposed DBNFS is both faster and more accurate than the other recent semi-supervised sentiment classification algorithms.

V. CONCLUSIONS

We have proposed an enhanced version of DBNs called DBNFS to address the sentiment classification problem. We have replaced several hidden layers in DBNs which are computationally expensive with the filter-based feature selection technique using chi-squared tests. Then, inessential features are filtered and only meaningful ones are selected. With the aid of the feature selection and reduction, the learning phase of DBNFS is more efficient as we have found from the experimental results. The classification accuracy of DBNFS outperforms the accuracy of the baseline semi-supervised learning algorithms such as spectral learning, transductive support vector machine (TSVM), and personal/impersonal views (PIV). Additionally, DBNFS can perform slightly better than the other deep learning algorithms, DBN, and Hybrid Deep Belief Networks (HDBN). Moreover, we have observed that DBNFS spends much less training time compared with the traditional DBN. In the future work, we plan to parallelize our algorithm and run on GPUs platform in order to accelerate its computation. We aim to tackle with large-scale problems in the real world with better scaling ability while good classification accuracy could still be maintained.

REFERENCES


Figure 5. Training time comparison between DBNFS and DBN.