A Deep Learning Approach to Classify State Cables

Karthik Anantha Padmanabhan
The University of Texas at Austin
akarthik@cs.utexas.edu

Vijay Talluru
The University of Texas at Austin
vijayt@cs.utexas.edu

Abstract

Old archived state diplomatic cables maintained by the National Archives and Records Administration undergo a review process where their original classification is reviewed. This is done with a view that some of the secret documents may no longer be considered sensitive and can be made available to the public. This is essentially a text classification task with two labels- "secret" and "unclassified". Using traditional text classification techniques is not suitable because of non-availability of sufficient labeled data and the sensitivity of information. Classifiers coupled with Deep Learning techniques have shown a lot of promise in a number of pattern recognition tasks due to the enhanced feature representations produced by Deep Learning techniques. In this paper we apply a Marginalized Stacked Denoising Autoencoder (mSDA) to solve the above mentioned problems in using a classifier for the "classification-review" of state cables. We hypothesize that because of the better feature representation obtained by mSDA, only fewer training samples are required when compared to the traditional Bag-of-Words approach. We confirm our hypothesis and also show that this property can be used to greatly reduce the human effort involved in the review process of state cables.

1. Introduction

The National Archives and Records Administration (NARA) [4] is an independent agency of the United States government that is responsible for preserving and documenting government and historical records. Far beyond its conservation role, NARA ensures that these records are increasingly available to the public. State cables are one among the various record collections that are maintained by NARA. State cables can be defined as text messages that are exchanged between a diplomatic mission like embassy or consulate with its parent embassy [3]. The state cables maintained by NARA should be released to the public whenever they are deemed fit to not contain sensitive information. So as a consequence, these old state cables undergo a review process where the original classification that was assigned to the cable at the time of its drafting, is reviewed so that they may be declassified and made available to the public. This process however involves manual inspection of a few hundred thousand documents. Essentially this involves manually classifying the documents into two major classes – *secret* and *unclassified*. The use of traditional text classification techniques for this problem faces two main challenges: (1) Need to label large number of documents to train the classifier (2) The cost involved in misclassifying a *secret* document as *unclassified* is high. The first issue arises because training a classifier requires sufficiently large amounts of labeled data and such data is not readily available. Already existing labeled documents cannot be used for training because their labels would be based on external conditions that existed during their "review period" and would not be relevant for the current review period. So, in order to generate a training set, a large number of documents have to be labeled from those documents that are currently under review. Secondly, as state cables can contain extremely sensitive information, the cost of misclassifying a *secret* document as *unclassified* would be catastrophic. Only very little classification error can be tolerated. So some amount of manual intervention is necessary. In fact a practical use-case of text classification techniques would be to filter out those documents that the classifier is sure of, and present to the archivist only a limited set of documents that have to be inspected. For this problem, apart from the overall accuracy of the classifier, the accuracy of the confident predictions is also important.

In this project we partially tackle the above two challenges of traditional text classification using Deep Learning techniques. Deep Learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features [2]. It has been shown that these higher level features produce better representations of data [2]. The performance of machine learning algorithms used in classifiers relies heavily on the features that are used to represent the data. So by using these better features from Deep Learning techniques we can build better classifiers. Deep Learning techniques coupled with traditional

classifiers have shown state of the art results in most pattern recognition tasks.

In this project we use a deep architecture to learn better document representations. We hypothesize that using these better representations, we can train a classifier with fewer training examples and it would perform as well as a classifier that is trained with a full training set using the Bag-of-Words model. This would result in fewer documents that have to be labeled to generate the training data. To verify our hypothesis we use a Marginalized Stacked Denoising Autoencoder (mSDA) [1] - a deep architecture that has shown promising results in another text classification task - Domain Adaptation in Sentiment Analysis. We also hypothesize that when using better features, the documents in different classes become more distinguishable. This would result in a classifier (Logistic Regression) being more confident about its correct predictions and as a result more documents may be filtered out.

We verify both our hypotheses and show that using Deep Learning techniques can reduce the manual effort involved in the classification-review process of state cables.

The outline of the remainder of the paper is as follows. Section 2 describes the related work that has been done in text classification using Deep Learning techniques. Section 3 describes the Marginalized Stacked Denoising Autoencoder and the state cables data set. Finally, in section 4 an analysis of the results is presented followed by conclusion in section 5 and future work in section 6.

2. Background and Related Work

2.1. Classification-review of state cables

As mentioned in section 1, state cables are text messages exchanged between an embassy post and its parent country. When these cables are exchanged a classification level is assigned to these labels which can take on the following: (1) *top secret* (2) *secret*, (3) *classified*, (4) *unclassified* [4]. *Top secret*, *secret* and *classified* cables contain information that was considered sensitive when it was drafted. The state cables that have been sent over the years are maintained by NARA. In order to increase the amount of information available to the public, all records maintained by NARA undergo a review process so that records which may no longer be considered sensitive can be released to the public. Old state cables also undergo a similar review process. For example a cable that was considered *secret* in 1974 may undergo a re-evaluation in 1980, when it may be considered that it no longer has any sensitive information. The number of documents that have to be reviewed is very large and this task rightly fits the text classification framework. As mentioned in section 1, there are some challenges in applying traditional text classification techniques. To handle the problem of sensitivity of information, the predictions of the classifier may not be used as such. Instead it can be used to filter out documents based on how confident its predictions are. Only the doubtful predictions of the classifier are then presented to the archivist. By doing this we can greatly reduce the manual effort involved.

2.2. Deep Learning in Text Classification

Text classification is the process of learning the characteristics of a predefined set of categories from a training corpus and categorizing new documents to the most relevant one based on its contents. Performance of text classification systems greatly depend on the document representations [5]. Generally, documents are represented as 'Bag-of-Words' which is a numeric vector representation of a document in which each element corresponds to the weight assigned to an n-gram. The drawbacks of the Bag-of-Words model are: (1) It does not capture the meaning of the word.(2) Similar documents may have little overlap if they use different vocabulary (3) Does not handle rare features that never appear in training data.

To overcome some of the limitations in the Bag-of-Words model a number of authors have proposed the usage of deep architectures to get better document representations. Liu et al. [6] explore the use of Deep Belief Networks as a feature extraction tool to get higher level representations of documents that model the semantic correlation among words in a document. These features are then used along with the document's label to train an SVM which is then used as a classifier to classify text. Ranzato et al. [7] used a Stacked Denoising Autoencoder (SDA) to learn document representations for text classification in a semi-supervised manner from partially labeled corpora. They show that making use of labels to learn the features gives much better performance than unsupervised feature learning. Glorot et al. [8] also used a Stacked Denoising Autoencoder for domain adaptation in Sentiment Analysis. Domain adaptation considers the setting in which the training and testing are sampled from different distributions. The authors train the SDA on the union of source and target distributions to reconstruct the input $\boldsymbol{x}$. A classifier trained on this new representation performs much better than when trained on the source distribution alone. But one of the drawbacks in using SDA is the computational complexity involved in learning an SDA. Chen et al [1] address this issue by using a linear denoiser as the basic building block of their Stacked Denoising Autoencoder. The authors also demonstrate that, on domain adaptation tasks, their mSDA performs as well as SDAs, but with a training time far lesser than that required to train an SDA.

Larochelle et al. [8] propose the Discriminative Restricted Boltzmann Machines that can be used directly as a classification framework instead of just a feature extraction framework. They evaluate the model on document classification tasks and show that its performance is much better when compared to shallow architectures like SVM's.

In this project, our task is most similar to the one that Liu et al. [6] have tackled. But instead of choosing the Deep Belief Network, we have opted to use the Marginalized Stacked Denoising Autoencoder [1] as our Deep Learning framework. We have opted for this approach as it is easily implementable and computationally less demanding when compared to other techniques. The following section gives a brief description of the Autoencoder and the Marginalized Stacked Denoising Autoencoder.

2.3. Deep Learning Networks

Deep Learning architectures have a hierarchy of layers, with the layers at higher levels taking as an input the output of the lower levels. Various Deep Learning algorithms have been developed in the Deep Learning literature [2]. In this section we will build up towards explaining the Marginalized Stacked Denoising Autoencoder by explaining the Stacked Denoising Autoencoder.

Autoencoders

The Autoencoders are the building blocks of a stacked Autoencoder and its variants. So it is important to understand this first.

An Autoencoder takes as an input $\boldsymbol{x} \in [\mathbf{0}, \mathbf{1}]^{\boldsymbol{d}}$ and uses an encoder to map it to $\boldsymbol{y} \in [\mathbf{0}, \mathbf{1}]^{\boldsymbol{d'}}$ which is a hidden representation. This is done using some deterministic mapping. Then, the hidden representation $\boldsymbol{y}$ is mapped back to $\boldsymbol{x}$ using a transformation similar to the first one. This is called reconstruction. The form of the mapping function looks like this: $\boldsymbol{y} = \mathbf{s}(\mathbf{W}\mathbf{x} + \mathbf{b})$**.** It is similar for the reconstruction. $\mathbf{W}$ is the weight matrix which needs to be constructed in a way that reduces reconstruction error. $\mathbf{s}$ is a non-linearity like a sigmoid. $\mathbf{y}$, as in a PCA, is a distributed representation that captures the coordinates along the main factors of variation in the data. As the number of training examples increase, this reconstruction error will decrease for test examples which belong to the same distribution as the training examples.

Denoising Autoencoders

As can be seen from the discussion above, while reconstructing the original input, there is a possibility that the Autoencoder might end up learning the identity function and this should be avoided. This is achieved by using the stochastic version of the Autoencoders which is called Denoising Autoencoder. The idea is to introduce a corruption in the input before feeding it to the encoder, and then let the decoder construct the original input. The corruption can be done randomly. As much as half the input vector can be corrupted. This way it can learn more robust features instead of just learning the identity.

Stacked Autoencoders and Denoising Auto-Encoders

Stacked Autoencoder (SA) stacks several Autoencoders whereas a Stacked Denoising Autoencoder (SDA) stacks several Denoising Autoencoders. Each layer is trained greedily. The input from the k^{th} layer of the Autoencoder/ Denoising Autoencoder is fed to the $(k+1)^{th}$ layer. It has been shown that SDAs extract the hidden features at the higher levels by grouping related hidden factors [2].

After the SA/DSA has been trained, the different hidden layers represent multiple levels of representation for the input. They can then be combined with the original input, and then given as a new input to a classifier. This latter usage improves the performance of the classification. This usage will be explored more in the later sections.

3. Marginalized Stacked Denoising Autoencoders (mSDA)

In the previous section we looked at various Autoencoders. Chen et al. [1], have explored a variant of SDA. The authors claim and demonstrate that linear denoisers could be used as a building block for the Stacked Denoising Autoencoder. It employs a greedy layer-by-layer training. In this setup, the random feature corruption can be marginalized out and this is equivalent to training the model with infinitely large number of corrupted data.

The linearity significantly simplifies the parameter estimation as they can be found by solving closed-form equations.

3.1. Linear Denoiser

As mentioned above, linear denoisers are used as the building blocks of the mSDA. $\mathbf{x_i}$ is the input vector and it is corrupted by random removal of the features. Each feature is set to 0 with a probability of p. If this corrupted version is $\tilde{\mathbf{x}}_\mathbf{i}$ the linear denoiser tries to reconstruct $\mathbf{x_i}$ from $\tilde{\mathbf{x}}_\mathbf{i}$. It differs from a Denoising Encoder in that it uses a linear mapping $\mathbf{W}: \mathcal{R}^d \rightarrow \mathcal{R}^d$ that reduces the squared construction loss.

$$\mathcal{L}(\mathbf{W}) = \frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x_i} - \mathbf{W}\,\tilde{\mathbf{x}}_\mathbf{i}\|^2 \qquad 1$$

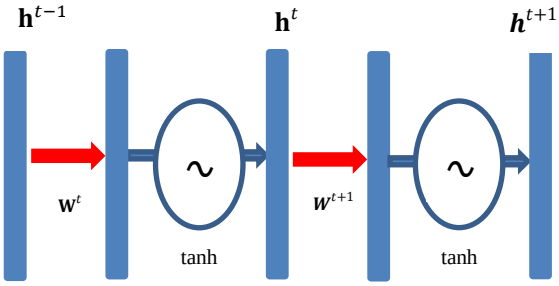

Figure 1: Stacking layers of mDA to form the mSDA

A constant feature is present within $\mathbf{x}_i$ which is never corrupted and a corresponding bias is incorporated within the mapping: $[\mathbf{W}, \mathbf{b}]$. With m samples of input vectors, $[\mathbf{x}_1, \ldots, \mathbf{x}_m]$ (with possible repitions), the data matrix is $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$ and the corrupted version is $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m\}$. The solution of (1) can be expressed as

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1} \qquad 2$$

where $\mathbf{Q} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathbf{T}}$ and $\mathbf{P} = \mathbf{X}\tilde{\mathbf{X}}^{\mathbf{T}}$.

3.2. Marginalization

Ideally, each sample input should be corrupted with all possible corruptions while computing $\mathbf{W}$. By the weak law of large numbers, matrices $\mathbf{P}$ and $\mathbf{Q}$ converge to their expected values E[$\mathbf{P}$] and E[$\mathbf{Q}$] with the increasing value of m. In the limit, i.e. as $m \to \infty$, the expectations are used to express the mapping for W in closed –form as:

$$\mathbf{W} = \mathrm{E}[\mathbf{P}]E[\boldsymbol{Q}]^{-1} \qquad 3$$

$$E[\boldsymbol{Q}]_{\alpha,\beta} = \begin{cases} \mathbf{S}_{\alpha\beta}\boldsymbol{q}_{\alpha}\boldsymbol{q}_{\beta} & \text{if } \alpha \neq \beta \\ \mathbf{S}_{\alpha\beta}\boldsymbol{q}_{\alpha} & \text{if } \alpha = \beta \end{cases},$$

$$E[\mathbf{P}]_{\alpha,\beta} = \mathbf{S}_{\alpha\beta}\boldsymbol{q}_{\beta},$$

$$\boldsymbol{q} = [1-p, \ldots, 1-p, 1]^{\mathrm{T}} \in \mathcal{R}^{d+1}$$

$\mathbf{S} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$ denotes the covariance matrix of the uncorrupted data.

This closed-form Denoising layer is Marginalized Denoising Autoencoder (mDA).

3.3. mSDA

Like in SDAs, where multiple layers of Denoising Autoencoders are stacked, the mDAs can be stacked to create a Marginalized Stacked Denoising Autoencoder. The output of the t^{th} layer is fed to $(t+1)^{th}$ layer as input. Each transformation $\mathbf{W}^t$ is learned to reconstruct the

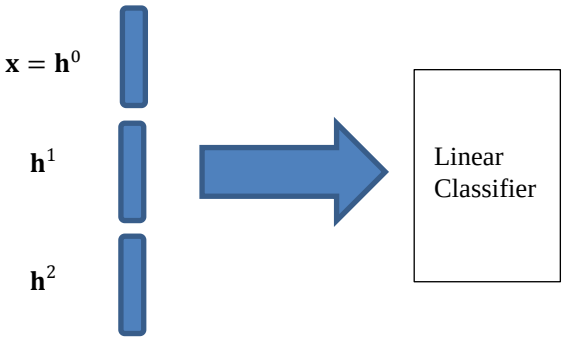

Figure 2: Forming the mSDA representation for documents

previous mDA output $\mathbf{h}^{t-1}$ from its corrupted equivalent. A non-linear squashing function is applied between layers to extend the mapping to a non-linear transformation. Each layer's representation is obtained from the preceding layer through $\mathbf{h}^t = \tanh(\mathbf{W}^t\mathbf{h}^{t-1})$. $\mathbf{h}^0$ denotes $\mathbf{x}$, the input.

3.4. Extension to high dimensional data

From Equation 3 it can be seen that calculating the weights involves finding the inverse of a matrix. When the dimensionality is very high finding the inverse of Q would become computationally expensive.

To work with high-dimensional data, instead of reconstructing all the corrupted features at once, only a subset of features is reconstructed. These features called pivot features [11] correspond to the most frequent terms in the documents. All the input features are divided into K manageable subsets, and the pivot features are reconstructed from these K different subsets. The K pivot reconstructions are summed. The subsequent layers don't require any special treatment. This approach can be used to scale-up the dimensionality of mSDAs as well as SDAs.

3.5. mSDA for classification-review of state cables

In text classification tasks, mSDA can be used to generate a new representation for a document from its Bag-of-Words representation. Using the mSDA we learn the features in an unsupervised fashion on the union of all the available documents (test and train). This is done by taking the representation at each layer, $\boldsymbol{h}^{\boldsymbol{t}}$, and combining it with the original Bag-of-Words representation. The new representation of the training documents, along with its labels is then used to train a Logistic Regression classifier. The trained classifier can then be used to classify the test documents. There are two meta-parameters that can be learned for the mSDA: (1) the corruption level p and the number of layers l. We have used a five-fold cross-validation technique on the training data to set these

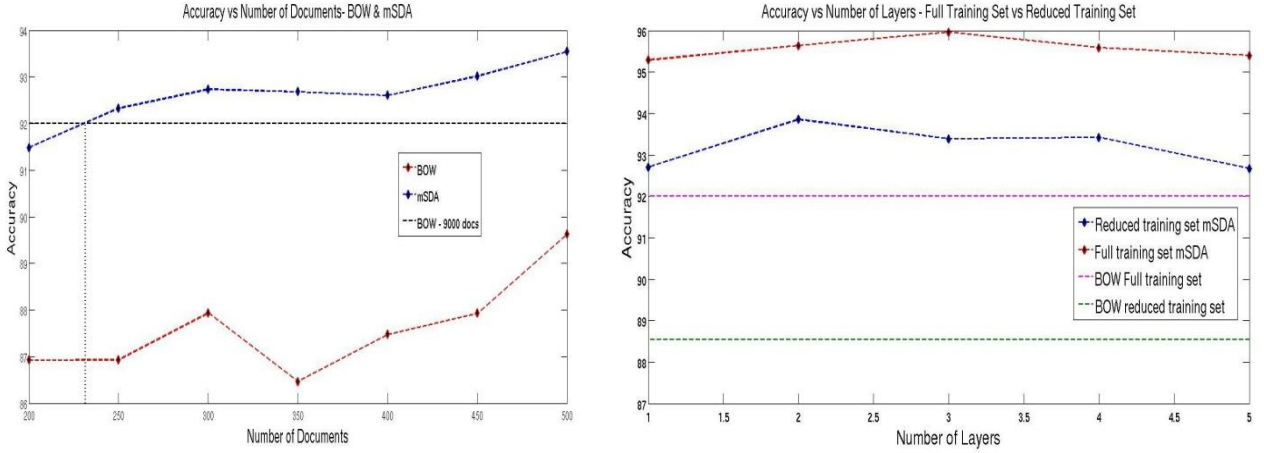

Figure 2(a) shows how the accuracy varies with the number of training documents. Figure 2(b) shows how the accuracy varies with the number of layers for the reduced training set and full training set

parameters. The following is a use case of how our framework can be used:

A whole bunch of state cables are presented to an archivist who has to review them. The archivist then takes the entire document collection and passes them through the mSDA to get the new representation for these documents. Out of the entire collection, the archivist then starts labeling a few documents. Now with these labels and the new document representation, a logistic regression classifier is trained. This trained classifier can then be used on the other documents. The classifier then outputs the labels and probability estimates for these documents. The archivist may then choose to select only those documents which have probability estimates (confidence) less than (say 0.9) and starts to manually label them.

3.6. The state cables dataset

The dataset that was used in this project consists of state cables that were exchanged in the years 1973 and 1974. Attached with each cable is metadata information associated with that cable. The fields that are most relevant to us were "Review Dates" and Review Classification". To form the dataset, we first chose those documents that have the same Review Date. Then from these documents we randomly selected 12,000 documents which have Review Classification = "*secret*" and 12,000 documents which have Review Classification = "*unclassified*". Finally from these state cables we extracted the subject and Message Body and append them together. We then chose 9000 documents to form the training set and 15000 documents to form the test set.

3.7. Experimental Setup

For the state cables dataset, preprocessing was done by removing stop words and stemming. Feature selection was done using Document Frequency technique and only terms that appeared at least in 3 documents were chosen. This reduced the feature dimension to 22398.Each document is then treated as a Bag-of-Words vector with each element in the vector corresponding to the number of times that word appeared in that documents.

As baseline we train a logistic regression classifier on the raw Bag-of-Words representation of the training set and use it to classify the test data. We chose logistic regression over other linear classifiers because Logistic regression outputs can be interpreted as the probability estimate of a document. The LIBLINEAR package supports logistic regression and linear support vector machines. We then evaluate the performance of the logistic regression classifier on the features learned by the Marginalized Stacked Denoising Autoencoder We carried out a number of tests to compare the Bag-of-Words representation with the new representation obtained by using mSDA, which we will call mSDA representation for brevity in this discussion. Also we will use the term "mSDA approach" when the classifier uses the mSDA representation and similarly for the "BOW approach". The following metrics were used to compare the baseline and mSDA approach:

3.8. Metrics

We defined few metrics to evaluate and compare the performance of the classifier on the documents represented with the mSDA representation as opposed to the Bag-of-Words representation.

First, we will define some notations that will be used henceforth. For each test document, $\boldsymbol{l}$ is the label given by the classifier. $\boldsymbol{l}$ is either $\boldsymbol{C}$ meaning *secret,* or $\boldsymbol{U}$ meaning *unclassified*. $\boldsymbol{T}$ is the actual label of this test document. $\boldsymbol{T}$ is, again, either $\boldsymbol{C}$ meaning *secret, or* $\boldsymbol{U}$ meaning *unclassified.* $\boldsymbol{PE}$ denotes the probability estimate (the confidence) with which the classifier has classified this

For *secret* documents $h(0.99) = N(\boldsymbol{C},\boldsymbol{U},\mathbf{0.99})/N(\boldsymbol{C},\boldsymbol{C},\mathbf{0.99})$		For *unclassified* documents $h(0.99) = N(\boldsymbol{C},\boldsymbol{U},\mathbf{0.99})/N(\boldsymbol{U},\boldsymbol{U},\mathbf{0.99})$	
mSDA	BOW	mSDA	BOW
0.0231 (1 layer)	0.0324	0.0041 (1 layer)	0.008
0.0171 (2 layers)		0.0035 (2 layers)	
0.0216(3 layers)		0.0051 (3 layers)	
0.0277(4 layers)		0.0067 (4 layers)	
0.0256(5 layers)		0.0079 (5 layers)	

Table (1) : High Confidence Error Ratio – $h(0.99)$ for mSDA vs BOW approaches with 1000 training documents

For *secret* documents $h(0.99) = N(\boldsymbol{U},\boldsymbol{C},\mathbf{0.99})/N(\boldsymbol{C},\boldsymbol{C},\mathbf{0.99})$		For *unclassified* documents $h(0.99) = N(\boldsymbol{C},\boldsymbol{U},\mathbf{0.99})/N(\boldsymbol{U},\boldsymbol{U},\mathbf{0.99})$	
mSDA	BOW	mSDA	BOW
0.0084 (1 layer)	0.029	0.0014 (1 layer)	0.007
0.0078 (2 layers)		0.0017 (2 layers)	
0.0057(3 layers)		0.0015 (3 layers)	
0.0077(4 layers)		0.0012 (4 layers)	
0.0107(5 layers)		0.0023 (5 layers)	

Table (2): High Confidence Error ratio $h(0.99)$ for mSDA vs BOW approach for 9000 training documents

test document. Now we can define the notation $N(\boldsymbol{l},\boldsymbol{T},\boldsymbol{PE}_t)$ to define the number of documents in the test set, which have a label $\boldsymbol{l}$ given by the classifier with a $\boldsymbol{PE}$ greater than $\boldsymbol{PE}_t$ and have an original label $\boldsymbol{T}$. For instance, $N(\boldsymbol{C},\boldsymbol{U},0.6)$ denotes the number of *unclassified* documents which have been classified as *secret* by the classifier with a confidence of more than 0.6. Note that $N(\boldsymbol{U},\boldsymbol{C},0.5)$ will give the number of *secret* documents which have been wrongly classified as *unclassified*.

With these above notations, we now define a metric called *high confidence error ratio h(z).* This is defined as the ratio $N(x,y,z)/N(y,y,z)$, where $x \neq y \,\&\, z \geq 0.9$. In words, for each class of documents, this is the ratio between the number of documents wrongly predicted with confidence more than z and the number of documents correctly predicted with confidence more than z.

h(z) is a very important metric for the following reason. The text classification step for this task is only a preliminary step. The documents will be filtered out only if the classifier has classified them with reasonable confidence. There will be a manual verification of the remaining documents which the classifier had classified with lesser confidence. As we want to filter out as many documents as possible and leave only few documents for the manual verification, it is imperative to consider the performance of the classifier when it has classified a document with a very high confidence. We want to increase the number of documents correctly classified with high confidence and at the same time reduce the number of documents wrongly predicted with high confidence. The metric *h* will capture this information. The lower this value the better the performance of the classifier in this aspect.

3.9. Results

For the experiments, we used a total of 24000 documents. The labels were available for all the documents. 15000 documents have been used for testing and the training set documents were picked from the remaining 9000 documents. Both the test set and the training set were evenly split between *secret* and *unclassified* documents.

We compared the accuracies of the classifier on the Bag-of-Words representation and the mSDA representation. The size of the training set was a varying parameter here. The graph in Figure 2(a) shows how the accuracies vary with the number of documents used for training. The number of layers in the mSDA has been fixed at 3 for this set of experiments.

The effect of the number of layers in the mSDA on the accuracy was also a point of interest. The graph in Figure 2(b) shows how the accuracy varies with the number of layers. This test was done for two sizes of the training sets: 1000 randomly chosen documents from the training set, and all 9000 documents. It also shows that for the Bag-of-Words approach, the accuracy was 92.0067% with all the 9000 documents used for training.

For the next set of experiments, we used the metric $h(z)$ as defined in the previous section. We fixed z at 0.99 and we compared the results of the classifier on the two representations. Table (1) and Table (2) show the comparisons of the *high confidence error ratio- h(0.99)* for the two representations for the set of *secret* documents as well as the *unclassified* documents. Table (1) shows the information for the training set size of 1000 documents (500 *secret* and 500 *unclassified*) whereas the Table (2) is for all the 9000 documents of the training set. Choosing the 1000 training documents out of 9000 was done randomly. For this reason, the results were each averaged over 10 runs of the experiment.

4. Discussion

It can be seen from the graph of Figure 2(b) that the accuracy of the mSDA representation is significantly better when compared to the Bag-of-Words approach even with just a single layer. As more number of layers are stacked, the accuracy increases. However after 3 layers the performance of mSDA degrades, but it still remains better than the accuracy with BOW approach (92.0067%). The same trend can be seen when the classifier is trained with only 1000 documents. The accuracy achieved with any number of layers, again, is still better than the accuracy achieved with the Bag-of-Words approach (88.5294%). It is also clear from this graph that the accuracy is much

better when more data is provided, which is expected. It is evident from the results that increasing the number of layers doesn't necessarily improve the performance of the mSDA approach. This could be because of the way mSDA actually constructs its feature representations. For every layer that is added to the mSDA, more word co-occurrence information is incorporated. For example, consider the document having just the words "President Obama Whitehouse". When Obama and President are removed due to corruption, we are learning to reconstruct "President Obama Whitehouse" from just "Whitehouse". So whenever we just see the words Whitehouse, we will add more contextual information by adding the words President and Obama. But as we increase the number of layers, we will add more word co-occurrence information which may not be contextually related. So increasing the number of layers beyond a point only adds more noise. So this could a reason why the accuracy decreased for Layers 4 & 5.

In Figure 2(a), it can be seen that the mSDA approach can achieve the performance of a BOW model trained on 9000 documents with just 440 documents. This amounts to roughly 20 fold reduction in the number of documents required for that level of accuracy. In fact, as the number of documents decreases, the difference in the performance of mSDA and BOW becomes more significant. mSDA's accuracy reduces by only 2.3 percentage points whereas, the BOW approach has a decrease in accuracy by 3.4 percentage points. This confirms our hypothesis that the mSDA approach, as a result of learning better features, can work with fewer training documents.

Table (1) and Table (2) show the value of the *high confidence error ratio h(0.99)* for the mSDA approach and for the BOW approach. The *high confidence error ratio* tells the extent to which the filtered documents were actually classified correctly. The smaller the value, the smaller is the error rate. From the two tables, it can be seen that the mSDA approach gives much better performance when compared to the BOW approach. Although the difference in the error ratio might be small, in terms of number of documents this is quite significant as even a single misclassification could have very bad consequences. Again as expected, with more training data, the results are better for both approaches. It is clear from the table that the value of $h(0.99)$ for the mSDA approach is lesser than that of the BOW approach. This is an indication that the mSDA approach, when it has classified a document with a very high confidence, is less likely to misclassify that document. It should also be noted that the value of $h(0.99)$ for the mSDA approach using 1000 training documents is still lesser when compared with $h(0.99)$ value for the BOW approach using all the 9000 documents. Thus it can be seen that even when the number of training documents are decreased the filtering obtained by the mSDA approach is either comparable or better than the filtering obtained by the BOW approach.

5. Conclusion

In this project, we aimed at reducing the manual work involved in the classification-review process of state cables. We solved the traditional problems of text classification using a Deep Learning technique-Marginalized Stacked Denoising Autoencoder. The results show that the accuracy obtained by the mSDA approach is better than that achieved by the BOW approach. Further, we also show that we can achieve a 20-fold reduction in the number of documents and still obtain the same accuracy as that of BOW approach. This verified our first hypothesis about having to use lesser number of training documents with Deep Learning techniques. We defined a metric $h(z)$ to measure the error made in the filtering step of this classification-review process. By comparing this metric for both BOW and the mSDA approaches, we were able to show that mSDA approach will give a filtering that is more accurate. This also verifies our second hypothesis that the mSDA representation makes the document classes more distinguishable.

The first hypothesis will enable us to use lesser number of documents to train the classifier. This reduces the manual effort in labeling the documents. The second hypothesis enables us to filter more number of documents accurately. This ensures that the documents that have been filtered out have indeed been correctly classified. So the traditional problems associated with the text classification can be solved to some extent by using Deep Learning techniques.

6. Future Work

In this paper, we fixed the corruption value at 0.9 through cross-validation for all our experiments. It will be interesting to look at the results for other corruption values. Also, our document representation only has unigrams. It is worth looking at the results when bigrams are used as well. In addition, instead of randomly corrupting the intermediate layer representations, we could use a feed-back loop to determine which words are more correlated and use that information to corrupt the input in a way that would give more robust performance. It will also be worthwhile to modify the mSDA to make use of the labels and learn the features in a semi-supervised manner.

References

[1] Chen, Minmin, et al. "Marginalized Stacked Denoising Autoencoders." *Learning Workshop*. 2012.

[2] Bengio, Yoshua. "Learning deep architectures for AI." *Foundations and Trends® in Machine Learning* 2.1 (2009): 1-127.
[3] http://en.wikipedia.org/wiki/Diplomatic_cables
[4] http://www.archives.gov/
[5] http://nlp.stanford.edu/IR-book/html/htmledition/features-for-text-1.html
[6] Liu, Tao. "A novel text classification approach based on deep belief network."*Neural Information Processing. Theory and Algorithms* (2010): 314-321.
[7] Ranzato, M., and Martin Szummer. "Semi-supervised learning of compact document representations with deep networks." *Internal Conference of Machine Learning*. 2008.
[8] Larochelle, Hugo, and Yoshua Bengio. "Classification using discriminative restricted Boltzmann machines." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.
[9] Chen, Minmin, et al. "Marginalized denoising autoencoders for domain adaptation." *arXiv preprint arXiv:1206.4683* (2012).
[10] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach." (2011).
[11] Blitzer, John, et al. "Learning bounds for domain adaptation." *Advances in neural information processing systems* 20 (2007): 129-136.