Open Set Text Classification using Convolutional Neural Networks

Sridhama Prakhya[†], Vinodini Venkataram[‡] and Jugal Kalita[‡]

[†]School of Engineering & Technology, BML Munjal University, Gurugram, India
[‡]Department of Computer Science, University of Colorado Colorado Springs, USA
[†]sridhama@sridhama.com
[‡]{vvenkata,jkalita}@uccs.edu

Abstract

In a *closed world* setting, classifiers are trained on examples from a number of classes and tested with unseen examples belonging to the same set of classes. However, in most real-world scenarios, a trained classifier is likely to come across novel examples that do not belong to any of the known classes. Such examples should ideally be categorized as belonging to an unknown class. The goal of an *open set* classifier is to anticipate and be ready to handle test examples of classes unseen during training. The classifier should be able to declare that a test example belongs to a class it does not know, and possibly, incorporate it into its knowledge as an example of a new class it has encountered. There is some published research in open world image classification, but open set text classification remains mostly unexplored. In this paper, we investigate the suitability of *Convolutional Neural Networks* (CNNs) for open set text classification. We find that CNNs are good feature extractors and hence perform better than existing state-of-the-art open set classifiers in smaller domains, although their open set classification abilities in general still need to be investigated.

1 Introduction

With increasing amounts of textual data being generated by various online sources like social networks, text classifiers are essential for the analysis and organization of data. Text classification usually consists of training a classifier on a labeled text corpus where individual examples belong to one or more classes based on their content, and then using the trained classifier to place unseen examples in one of these classes. Popular text classification applications include spam filtering, sentiment analysis, movie genre classification, and document classification. Traditional text classifiers assume a *closed world* approach. In other words, the classifier is implicitly expected to be tested with examples from the same classes with which it was initially trained. However, such classifiers fail to identify and adapt when examples of previously unseen classes are presented during testing. In real-world scenarios, a robust trained classifier should be able to recognize examples of unknown classes and accordingly update its learned model. This is known as the *open world* approach to classification. Most research in open set classification has been in the computer vision domain, primarily in handwriting recognition (Jain et al., 2014), face recognition (Li and Wechsler, 2005; Scheirer et al., 2013), object classification (Bendale and Boult, 2015; Bendale and Boult, 2016) and computer forensics (Rattani et al., 2015). Open set classification is important in computer vision since the number of classes to which a seen object can belong to is almost limitless and datasets are available with training samples belonging to thousands of classes. Nevertheless, open set classification is important in natural language processing as well. An example of an open world text classification scenario is authorship attribution, where each author happens to be a class. An open set text classifier must recognize the author of a document to be one of the known ones when appropriate. Importantly, the classifier should also explicitly recognize when it fails to classify an unseen document as written by one of the known authors. Whether it is for historical or fictional works from the past, or emails, social media posts or leaked political documents, open set classification may be immensely helpful.

S Bandyopadhyay, D S Sharma and R Sangal. Proc. of the 14th Intl. Conference on Natural Language Processing, pages 466–475, Kolkata, India. December 2017.

In the recent past, many-layered Artificial Neural Networks (ANN) or deep learning techniques (Goodfellow et al., 2016) have become popular in Computer Vision and Natural Language Processing. This is mainly attributed to the increase in performance compared to standard machine learning techniques. As discussed later, current open set text classifiers do not rely on deep learning models. They employ either a clustering-based approach (Doan and Kalita, 2017) or a modified Support Vector Machine (SVM) (Fei and Liu, 2016). To this end, we explore the possibility of using a CNN for open set text classification and compare it to existing techniques.

2 Related Work

To allow for the possibility that the set of classes is open or expandable during deployment, the classification algorithms need to be adaptive. (Scheirer et al., 2013) combine empirical risk and open space risk due to the existence of a space in which classification probabilities are not currently known. Empirical risk comes from actual examples being misclassified by a trained classifier, and the open space risk recognizes the fact that the presence of unknown classes is likely to introduce errors into classification decisions. Their model reduces the risk by introducing parallel hyperplanes, one near the class boundary and another far from it to introduce slabs of subspaces for the classes, and then develops a greedy optimization algorithm that modifies a linear SVM and moves the planes incrementally. This work was extended to multi-class open set classification by introducing what (Scheirer et al., 2014) call a Compact Abating Probability (CAP) model. They build a classifier called W-SVM using properties of Extreme Value Theory for calibration of scores produced by 1-class and binary SVMs. Extreme Value Theory (EVT) (Smith, 1990; De Haan and Ferreira, 2007; Castillo, 2012) is usually used to deal with and predict rare events or values that occur at the tails of distributions. The unnormalized probability of inclusion for each class is estimated by fitting a Weibull distribution (Sharif and Islam, 1980) over the positive class scores from SVM classifiers. The assumption here is when a trained classifier cannot classify an example as belonging to any of the known classes, it is a case of "failure" of the classifier and is deemed unusual. (Jain et al., 2014) also use EVT to formulate the open set classification problem as one of modeling positive training data at the decision boundary. They introduce a new algorithm called the P_i-SVM for estimating the unnormalized posterior probability of class inclusion. Their approach is different from the one introduced by (Platt and others, 1999) of taking SVM outputs and converting them to probabilities by fitting a sigmoid function to the SVM scores.

(Bendale and Boult, 2015) present an approach to minimize the weighted sum of empirical risk and open set risk using thresholding sums of monotonically decreasing recognition functions, and use their approach to extend the Nearest Centroid Classifier (NCM) (Rocchio, 1971). This classifier represents classes by the mean feature vector of its elements. An unseen example is assigned a class with the closest mean. The Nearest Non-Outlier (NNO) algorithm (Bendale and Boult, 2015) adapts NCM for open set classification, taking into account open space risk and metric learning. The nearest class mean metric learning (NCMML) (Mensink et al., 2013) approach extends the NCM technique by replacing the Euclidean distance with a learned low-rank Mahalanobis distance. This gives better results than the former as the algorithm is able to learn features inherent in the training data.

All the work mentioned so far have been in the context of computer vision. Work in open set classification for textual data is limited. (Fei and Liu, 2016) use CBS learning (Fei and Liu, 2015) where a document is represented as a vector of similarities from centers of spheres that correspond to individual classes. Around the sphere that represents positive examples of a class, they draw a slightly bigger sphere to provide additional space for a class to accommodate unseen examples. They also use SVM hyperplanes to bound the bigger spheres. The unbounded regions correspond to unknown classes.

The Nearest Centroid Class (NCC) algorithm (Doan and Kalita, 2017) builds upon the NCM, but uses a density-based method following the approach of the clustering algorithm called DBSCAN (Ester et al., 1996). They represent a class not by a sphere but a set of density-connected regions and also consider the centroids of these regions and not the means.

In the context of deep learning, (Bendale and Boult, 2016) adapt a CNN (Krizhevsky et al.,

2012) to perform open set classification in the vision domain. In closed set classification, the final softmax layer of the CNN essentially chooses the output class with the highest probability with respect to all other output labels. Bendale and Boult propose OpenMax, which is a new model layer that estimates the probability of an input belonging to an unknown class instead of softmax. (Ge et al., 2017) adapt OpenMax to generative adversarial networks (GANs) for open set vision problems. There have been no such attempts in the text processing domain.

3 Method

Along the lines of existing open set techniques, our work was also motivated by the Rocchio method (Rocchio, 1971). We wanted to use pretrained word vectors (Mikolov et al., 2013) for open set determination. This led us to perform experiments to see whether simple cosine computation can be used for open set classification. We used a naive approach to construct document vectors by averaging all word vectors (Le and Mikolov, 2014) in a document. We calculated the cosine similarities between the mean of all document vectors and a test example. Due to the similarities being too close (sometimes overlapping), we concluded that calculating cosine similarity at the document level was not suitable for open set classification.

Prior open set text classification models (CBS learning and NCC) do not use artificial neural networks. We decided to pursue a novel approach to open set text classification that relied on a deep learning model, viz. CNNs due to their ability of extracting useful features. Since (Bendale and Boult, 2016) explored the use of CNNs in open set image classification, we started with their approach as the basis and extend the work as necessary. The work of (Kim, 2014) in CNNs for sentence classification helped us arrive at an efficient neural network architecture. Thus, we perform experiments with a single-layer CNN, using the Weibull-modified final layer instead of softmax. We also examine if increasing the number of CNN layers changes performance of open set text classification. We develop a novel ensemble approach to deal with the activations of the penultimate layer of the CNN. The penultimate layer is the focus because this is the layer that contains the real activations for nodes corresponding to the various classes for the problem at hand. Since these are raw activations, in a standard CNN, they are converted into probability-like values by performing the softmax operation.

$$softmax\left(x\right)_i \;=\; \frac{e^{x_i}}{\sum_j e^{x_j}} \qquad (1)$$

However, in our case, there is an unknown class to be considered as well and we do not know the activations or probabilities associated with such an unknown class. Therefore, this softmax layer needs to be modified. (Bendale and Boult, 2016) replace the layer that computes softmax with the so-called OpenMax layer, which uses a learned distance metric taking into account the open set risk.

Our new model uses an ensemble approach to make a decision with the activations in the penultimate layer. Our model is also incremental in nature. This means, the model does not have to be retrained after the introduction of a new unknown class. This is because open set determination happens after training, rather than during or before.

In our experiments discussed here, we compare the performance of our ensemble-based open set text classifier with other open set classifiers that have been previously used for image classification and the methods of (Fei and Liu, 2016) and (Doan and Kalita, 2017), which were used for open set text classification.

3.1 Datasets

For efficacious open world evaluation, we must choose a dataset with a large number of classes. This allows us to hide classes during training. These hidden classes can later be used during testing to gauge the open world accuracy. We use the following two freely available datasets.

- **20 Newsgroups** (McCallum et al., 1998; Slonim and Tishby, 2000) - Consists of 18,828 documents partitioned (nearly) evenly across 20 mutually exclusive classes.
- **Amazon Product Reviews** (Jindal and Liu, 2008) - Consists of 50 classes of products or domains, each with 1,000 review documents.

3.2 Evaluation Procedure

Traditional evaluation (closed set) occurs when the classifier is assessed with data similar to what was learned during training. The number of classes presented during testing is equal to the number

the model was trained on. In open set evaluation, the classifier has incomplete knowledge during the training phase. Examples of unknown classes can be submitted to the classifier during the testing phase. During the training phase, we train the classifiers on a limited number of classes. While testing, we then present the model with additional classes that were not learned during training. We evaluate the performance of the classifier based on how well it identifies these new classes. *"Openness"*, proposed by (Scheirer et al., 2013; Scheirer et al., 2014), is a measure to estimate the open world range of a classifier. This measure is only concerned with the number of classes rather than the open space itself.

$$openness = 1 - \sqrt{(2 \times C_T)/(C_R + C_E)} \quad (2)$$

where:

C_T = number of classes used for training
C_R = number of classes to be recognized
C_E = number of classes used during evaluation/testing

As a special case, when $C_T = C_R = C_E$, the value of *openness* is 0, i.e., it is the case of traditional classification when the numbers of classes trained on, tested on, and recognized are the same.

Accuracy, precision, recall, and F-score are used to measure the closed set performance of our model. These metrics are expanded to the open set scenario by grouping all unknown classes into the same subset. A True Positive is when an example of a known class is correctly classified and a True Negative is when an example of an unknown class is correctly predicted as unknown. False Positives (an unknown class predicted as known) and False Negatives (a known class predicted as unknown) are the two types of incorrect class assignment. Figure 1 shows how *openness* varies with the number of training classes when there are 10 testing classes.

4 Experiments

For all experiments, the CNN-static architecture proposed by (Kim, 2014) is used. We use pre-trained `word2vec`[1] (Mikolov et al., 2013) vectors as our word embeddings. These embeddings are kept static while other parameters of the model are learned. According to the experiments of (Zhang and Wallace, 2015), imposing an L^2 norm constraint on the weight vectors generally does not improve performance drastically. Figures 3 and 4 show the accuracies achieved on the 20 Newsgroups dataset while varying the L^2 norm constraint. Increasing the L^2 norm constraint proved detrimental to the model accuracy. The configuration details of the CNN used in all our experiments are shown in Table 1. Figure 2 shows a depiction of the CNN architecture we implemented. In our case, we use a single static channel instead of multiple channels.

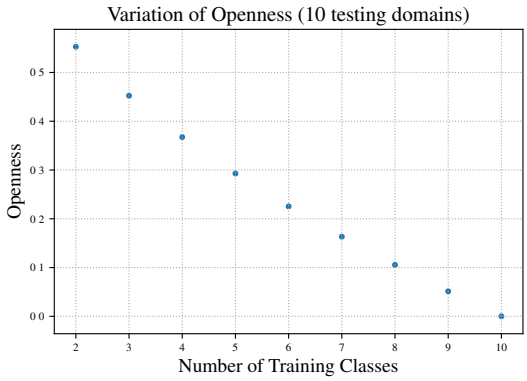

Figure 1: Variation of openness with number of training classes

Table 1: CNN baseline configuration

Description	Values
word embedding	`word2vec`
filter sizes	(3,4,5)
feature maps	100
activation function	ReLU
pooling	1-max pooling
dropout rate	0.5
L^2 norm constraint	0.0

4.1 Multi-layer CNN

In addition to Kim's architecture, we have also experimented with multi-layer CNNs. We used 2 convolutional layers, the initial layer used a kernel of size 3×1, while the second layer used a kernel of size 3×300. The first layer convolves the *same* feature across multiple words of the document. The second layer convolves *all* features (obtained from the previous convolution) across multiple (3 in our case) rows. The motive behind this approach was to extract activation vec-

[1] `https://code.google.com/p/word2vec/`

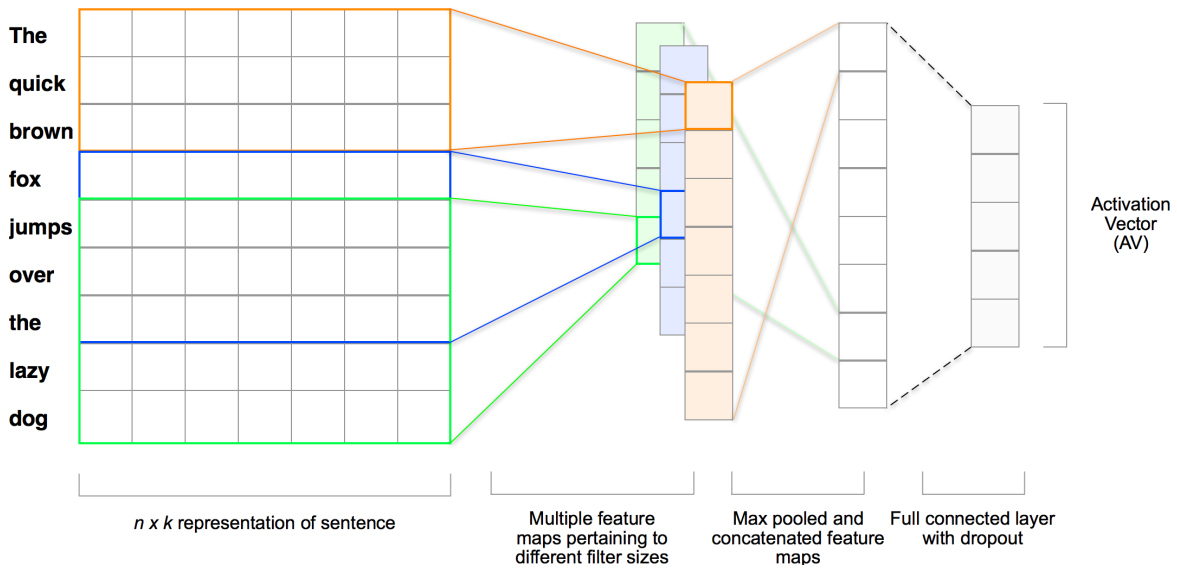

Figure 2: Model architecture with multiple filter sizes (3, 4, 5) for an example sentence

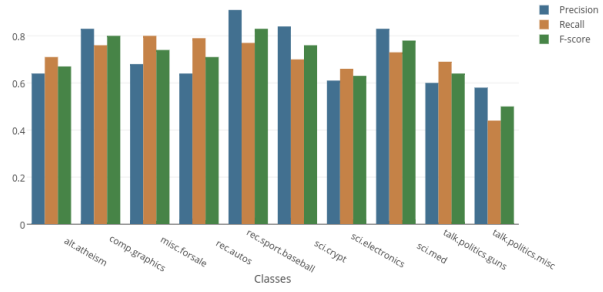

Figure 3: L^2 constraint = 0.0, Model Accuracy: 0.710

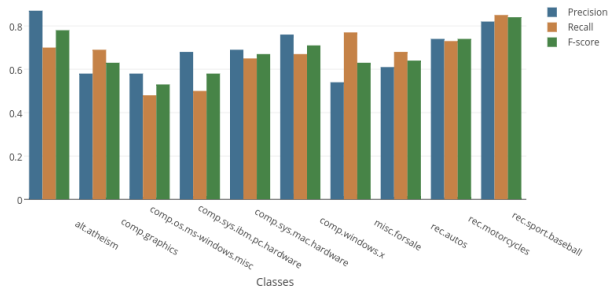

Figure 4: L^2 constraint = 3.0, Model Accuracy: 0.672

tors from the antepenultimate layer, which may represent the document more accurately. Unfortunately, the closed set (trained on 3 classes) accuracy of the muli-layer CNN was around 75%. The accuracy decreased significantly as we increased the number of training classes. A high closed set accuracy is necessary to achieve respectable open set results. Intuitively, the model must have a comprehensive understanding of what it knows. Only then can it be competent enough to classify unknown inputs correctly.

4.2 Ensemble Approach

In our open set classifier, we use an ensemble of approaches to determine whether a test example is from a known class or not. This ensemble includes probabilistic and high dimensional outlier detectors.

4.2.1 Isolation Forest

The isolation forest algorithm (Liu et al., 2008) detects outliers using combinations of a set of isolation trees. Isolation trees recursively partition the data at random partition points with randomly chosen features. Doing so isolates instances into nodes containing only one instance. The heights of branches containing outliers are comparatively less than other data points. The height of the branch is used as the outlier score. The scores obtained from the isolation forest are min-max normalized and calculated for every training class. Examples with scores below a predefined threshold are labelled as unknown. In case of multiple scores above the threshold, the example is assigned to the class with the highest score.

4.2.2 Probabilistic Approach

OpenMax (Bendale and Boult, 2016) is a new model layer based on the concept of Meta-Recognition (Scheirer et al., 2011). For all positive examples of every trained class, we collect the scores in the penultimate layer of our neural

network. We call these scores activation vectors (AV). We deviate from the original OpenMax by finding the k-nearest examples to the centroid of every training class. We refer to these examples as k-Class Activation Vectors (k-CAV). For every example in a training class, we calculate the distances between the respective AV and the k-CAVs. Doing so, results in k distances per AV. We then take the average of these k calculated distances. As the number of classes in our dataset is far less than those used in image classification, the k-CAVs of a class are used represent a class more accurately than a single mean activation vector. This also mitigates the effect of outlier AVs in a class. We observed that when k is around 10, the trade-off between performance and computation time is optimized. Therefore, for all experiments, we fix the value of k = 10.

In our outlier ensemble, we use two distance metrics – Mahalanobis distance and Euclidean-cosine (Eucos) distance (Bendale and Boult, 2016). Ideally, we want a distance metric that can tell us how much an example deviates from the class mean. The Mahalanobis distance precisely does this by giving us a multi-dimensional generalization of the number of standard deviations a point is from the distribution's mean. The closer an example is to the distribution mean, the lower is the Mahalanobis distance. The Mahalanobis distance between point x and point y is given by:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T C^{-1} (\vec{x} - \vec{y})} \qquad (3)$$

where C is the covariance matrix, among the feature variables calculated a priori. The Euclidean-cosine distance is a weighted combination of Euclidean and cosine distances.

The distances obtained are used to generate a Weibull model for every training class. We use the libMR[2] (Scheirer et al., 2011) *FitHigh* method to fit these distances to a Weibull model that returns a probability of inclusion of the respective class. Figure 5 shows the probabilities of inclusion obtained from the generated Weibull model for 2 training classes belonging to the 20 Newsgroups dataset. As an example deviates more from the class center (k-CAVs), the probability of inclusion decreases.

The sum of all inclusion probabilities is taken as the total closed set probability. Open set probability (OSP) is computed by subtracting the total closed set probability from 1.

$$OSP = 1 - total\ closed\ set\ probability \qquad (4)$$

We then compare the maximum closed set probability and total open set probability. If the total open set probability is greater than the former, we label the example as unknown, otherwise, the example is assigned the class with the highest closed set probability. Parameters like threshold and distribution tail-size can be be adjusted to decrease the open-space risk.

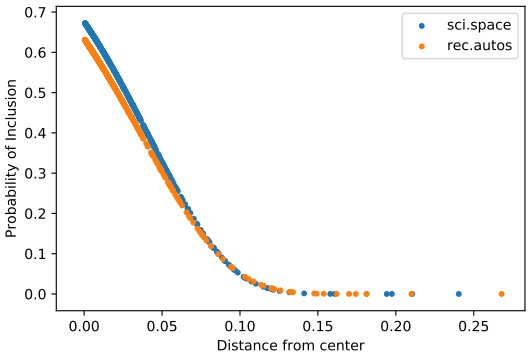

Figure 5: Weibull distribution generated using libMR for two classes belonging to the 20 Newsgroups dataset

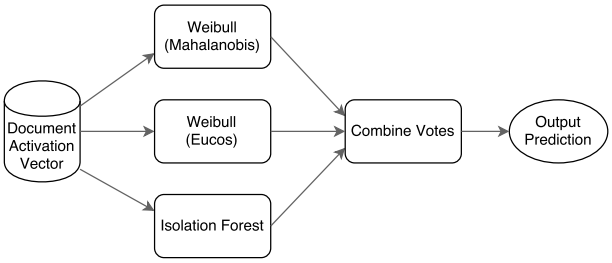

Figure 6: Our ensemble model

We use a voting scheme to combine the three approaches (Mahalanobis Weibull, Eucos Weibull and Isolation Forest), see Figure 6. It has been observed that Mahalanobis and Eucos perform nearly the same. Predictions from the Isolation Forest are usually used as a tie-breaker in case of differing predictions. When all 3 predictions differ, we give the Eucos Weibull the highest priority.

5 Results and Discussion

Open set performance largely depends on the "unknown" classes used during evaluation. This is especially true when classes are not completely exclusive. The activation vectors of similar classes

[2] https://github.com/Vastlab/libMR

Table 2: Experiments on Amazon Product Reviews dataset (10, 20 domains)

Amazon Product Reviews	**10 Domains**			
	25%	**50%**	**75%**	**100%**
our model	**0.797**	**0.753**	0.727	0.821
NCC §	0.61	0.714	**0.781**	0.854
cbsSVM*	0.450	0.715	0.775	**0.873**
1-vs-rest-SVM*	0.219	0.658	0.715	0.817
ExploratoryEM*	0.386	0.647	0.704	0.854
1-vs-set-linear*	0.592	0.698	0.700	0.697
wsvm-linear*	0.603	0.694	0.698	0.702
wsvm-rbf*	0.246	0.587	0.701	0.792
P_i-osvm-linear*	0.207	0.590	0.662	0.731
P_i-osvm-rbf*	0.061	0.142	0.137	0.148
P_i-svm-linear*	0.600	0.695	0.701	0.705
P_i-svm-rbf*	0.245	0.590	0.718	0.774

Amazon Product Reviews	**20 Domains**			
	25%	**50%**	**75%**	**100%**
our model	**0.648**	0.603	0.663	**0.793**
NCC §	0.606	0.657	**0.702**	0.78
cbsSVM*	0.566	**0.695**	0.695	0.760
1-vs-rest-SVM*	0.466	0.610	0.616	0.688
ExploratoryEM*	0.571	0.561	0.573	0.691
1-vs-set-linear*	0.506	0.560	0.589	0.620
wsvm-linear*	0.553	0.618	0.625	0.641
wsvm-rbf*	0.397	0.502	0.574	0.701
P_i-osvm-linear*	0.453	0.531	0.589	0.629
P_i-osvm-rbf*	0.143	0.079	0.058	0.050
P_i-svm-linear*	0.547	0.620	0.628	0.644
P_i-svm-rbf*	0.396	0.546	0.675	0.714

usually overlap in vector space. Similar to (Fei and Liu, 2016; Doan and Kalita, 2017), we conduct our experiments by introducing "unseen" classes during testing. In reality, as the train-test partition can be random, we arbitrarily specify the number of testing domains. For every domain, we report our results using 5 random train-test partitions for each dataset. Both datasets are evaluated on the same number of test classes (10, 20). We also evaluate our model on smaller domains, shown in Table 4. The number of testing classes used during training is varied in quarter-step increments (25%, 50%, 75% and 100%). We take the floor value in case of fractional percentages. Using 100% of the testing classes during training corresponds to closed set classification.

Results of the Amazon Product Reviews and 20 Newsgroups datasets are shown in Tables 2 and 3 respectively. We report only the F-scores due to space constraints. Classifiers used as baselines for comparison are described below.

- **1-vs-rest-SVM** - Standard 1-vs-rest multi-class SVM with Platt Probability Estimation (Platt and others, 1999)
- **1-vs-set-linear** - 1-vs-set machine model proposed by (Scheirer et al., 2013)
- **W-SVM** - Weibull-calibrated SVM (Scheirer et al., 2014)
- P_i**-SVM** - SVM model that estimates the unnormalized posterior probability of class inclusion (Jain et al., 2014)
- **ExploratoryEM** - "Exploratory" version of Expectation-Maximization algorithm (EM) (Dalvi et al., 2013)
- **cbsSVM** - Center-Based Similarity Space SVM (Fei and Liu, 2016)

Table 3: Experiments on 20 Newsgroups dataset (10, 20 domains)

20 Newsgroups	**10 Domains**			
	25%	**50%**	**75%**	**100%**
our model	**0.719**	0.747	0.738	0.864
NCC §	0.652	**0.781**	**0.818**	**0.878**
cbsSVM*	0.417	0.769	0.796	0.855
1-vs-rest-SVM*	0.246	0.722	0.784	0.828
ExploratoryEM*	0.648	0.706	0.733	0.852
1-vs-set-linear*	0.678	0.671	0.659	0.567
wsvm-linear*	0.666	0.666	0.665	0.679
wsvm-rbf*	0.320	0.523	0.675	0.766
P_i-osvm-linear*	0.300	0.571	0.668	0.770
P_i-osvm-rbf*	0.059	0.074	0.032	0.026
P_i-svm-linear*	0.666	0.667	0.667	0.680
P_i-svm-rbf*	0.320	0.540	0.705	0.749

20 Newsgroups	**20 Domains**			
	25%	**50%**	**75%**	**100%**
our model	**0.668**	0.686	0.685	0.787
NCC §	0.635	**0.723**	**0.735**	**0.884**
cbsSVM*	0.593	0.701	0.720	0.852
1-vs-rest-SVM*	0.552	0.683	0.682	0.807
ExploratoryEM*	0.555	0.633	0.713	0.864
1-vs-set-linear*	0.497	0.557	0.550	0.577
wsvm-linear*	0.563	0.597	0.602	0.677
wsvm-rbf*	0.365	0.469	0.607	0.773
P_i-osvm-linear*	0.438	0.534	0.640	0.757
P_i-osvm-rbf*	0.143	0.029	0.022	0.009
P_i-svm-linear*	0.563	0.599	0.603	0.678
P_i-svm-rbf*	0.370	0.494	0.680	0.767

- **NCC** - Nearest Centroid Class model (Doan and Kalita, 2017)

F-score performances of 1-vs-rest-SVM, 1-vs-set SVM, W-SVM, P_i-SVM, and cbsSVM are from study (Fei and Liu, 2016), marked as *. Results pertaining to the Nearest Centroid Class model (NCC) are from study (Doan and Kalita, 2017), marked as §. Our model performs better than cbsSVM and NCC classifiers in smaller domains. Figure 7 shows the activation vectors obtained from models trained on 2 classes plotted in 2-dimensional space. The plots show distinct clusters of activation vectors. We believe the CNN approach effectively isolates documents in smaller domains compared to other SVM-based approaches.

Unlike cbsSVM, our model is an incremental model i.e., we do not have to retrain the model when new unknown classes are introduced. Such models are more viable in real world scenarios.

Table 4: Open set results of Amazon Product Reviews Dataset in smaller domains (3, 4, 5)

Classes Trained on	**Classes Tested on**		
	3	**4**	**5**
2	0.802	0.824	0.808
3	-	0.725	0.763
4	-	-	0.797

6 Conclusion

Our incremental open set approach handles text documents of unseen classes in smaller domains more consistently than existing text classification models, namely CBS learning and the NCC model. This research can prove beneficial when

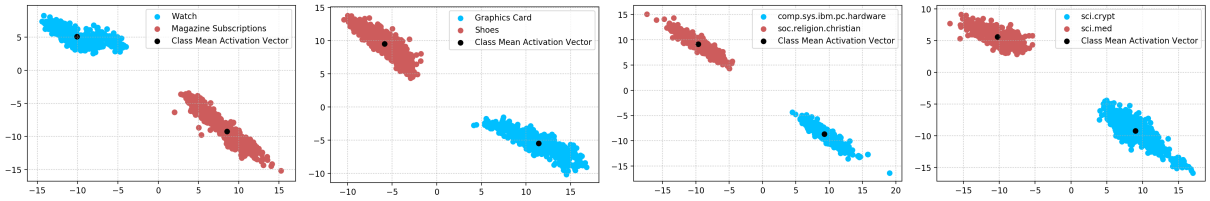

Figure 7: Activation vectors obtained from models trained on 2 randomized classes.

classifying novel data, applications of which can be used to tackle tough text classification problems in domains like forensic linguistics.

Our future work will involve improving the number and diversity of classifiers used in the ensemble. In addition, we plan to consider different neural network architectures that learn sequential information from text, namely variants of *recurrent neural networks* like *Long Short-Term Memory* networks with *attention* mechanism.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-1359275 and IIS-1659788. We are thankful for the support of BML Munjal University, particularly Prof. Sudip Sanyal and Dr. Satyendr Singh. We also thank Diptodip Deb and Kyle Yee for their insightful discussions and constant encouragement.

References

Abhijit Bendale and Terrance Boult. 2015. Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1893–1902.

Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572.

Enrique Castillo. 2012. *Extreme value theory in engineering*. Elsevier.

Bhavana Dalvi, William W Cohen, and Jamie Callan. 2013. Exploratory learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 128–143. Springer.

Laurens De Haan and Ana Ferreira. 2007. *Extreme value theory: an introduction*. Springer Science & Business Media.

Tri Doan and Jugal Kalita. 2017. Overcoming the challenge for text classification in the open world. In *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*, pages 1–7. IEEE.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Geli Fei and Bing Liu. 2015. Social media text classification under negative covariate shift. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2347–2356.

Geli Fei and Bing Liu. 2016. Breaking the closed world assumption in text classification. In *HLT-NAACL*, pages 506–514.

ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. 2017. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Lalit P Jain, Walter J Scheirer, and Terrance E Boult. 2014. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision*, pages 393–409. Springer.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230. ACM.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Fayin Li and Harry Wechsler. 2005. Open set face recognition using transduction. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1686–1697.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE.

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

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. 2013. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Ajita Rattani, Walter J Scheirer, and Arun Ross. 2015. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, 10(11):2447–2460.

Joseph John Rocchio. 1971. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*.

Walter J Scheirer, Anderson Rocha, Ross J Micheals, and Terrance E Boult. 2011. Meta-recognition: The theory and practice of recognition score analysis. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1689–1695.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2013. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772.

Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. 2014. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 36, November.

M Nawaz Sharif and M Nazrul Islam. 1980. The weibull distribution as a general model for forecasting technological change. *Technological Forecasting and Social Change*, 18(3):247–256.

Noam Slonim and Naftali Tishby. 2000. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–215. ACM.

Richard L Smith. 1990. Extreme value theory. *Handbook of applicable mathematics*, 7:437–471.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.