Open-Set Deep Learning for Text Classification

Sridhama Prakhya
Email: sridhama@sridhama.com

Vinodini Venkataram
Email: vvenkata@uccs.edu

Jugal Kalita
Email: jkalita@uccs.edu

***Abstract*—Most research in text classification has been done under a *closed world* assumption. That is, the classifier is tested with unseen examples of the same classes that it was trained with. However, in most real world scenarios, we come across novel data that do not belong to any of the known classes, and hence should not ideally be categorized correctly by the classifier. The goal of *open world* classifiers is to anticipate and be ready to handle test examples of classes unseen during training. The classifier can simply declare that a test example belongs to an unknown class, or alternatively, incorporate it into its knowledge as an example of a new class it has learned. Although substantial research has been done in open world image classifiers, its applications in text classification is yet to be explored thoroughly.**

***Keywords*—open-set classification, text classification, convolutional neural networks, deep learning, outlier ensembles, isolation forest, weibull distribution**

I. Introduction

With increasing amounts of textual data from various online sources like social networks, text classifiers are essential for the analysis and organization of data. Text classification usually consists of a corpus being assigned one or more classes according to its content. Some popular text classification applications include: spam filtering, sentiment analysis, movie genre classification and document tagging. Traditional text classifiers assume a *closed world* approach. The classifier is expected to be tested with the same classes that it was initially trained with. Such classifiers fail to identify and mitigate when examples of new classes are presented during testing. In real world scenarios, classifiers must be able to recognize unknown classes and accordingly adapt their learning model. This is known as the *open world* approach. A popular example of an open world text classification scenario is authorship attribution. An open world text classifier must recognize the author of a document and subsequently label it appropriately. The classifier must also recognize whether the writing style matches a known author, or is something unknown.

In this paper, we elaborate the methodology that we followed in developing our CNN-based open-set text classifier.

II. Related Work

A majority of existing open-set learning techniques deal with image classification rather than text classification.

The basis of most open-set classifiers is the Nearest Class Mean Classifier (NCM) [16]. This classifier represents classes by the mean feature vector of its elements. An unseen example is assigned a class with the closest mean. This is calculated by taking the distance *(Euclidean)* between the test vector and the computed class mean feature vectors.

Mensink et al. [4] proposed the nearest class mean metric learning (NCMML) approach extending the NCM technique by replacing the Euclidean distance with a learned low-rank Mahalanobis distance. This showed better results than the former as the algorithm was able to learn features inherent in the training data. The Nearest Non-Outlier (NNO) algorithm [3] adapts NCM for open world recognition based on a metric known as *open space risk*. This concept, introduced by Scheirer et al [11], minimizes an error function combining empirical risk over training data with the risk model for the open space. The NNO algorithm proved to perform better at image classification than the NCMML technique.

Regarding closed-set text classifiers, Fei and Liu [1] piloted an approach that they call *CBS learning*. Doan and Kalita [2] built upon the NCM, designing a set of closest neighbors of centroid class rather than the class mean for each class member.

Most ANN-based closed-world text classifiers use recurrent neural network based architectures e.g., Long short-term memory (LSTM) models. The state-of-the-art classifier uses a convolutional neural network model that is 29 layers deep [7]. Conneau et al. were able to show that the performance of their model increased with depth of the network.

III. Method

A. Datasets

For an 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 plan on using two freely available data sets:

- 20 Newsgroups [14] - Consists of 18828 documents partitioned (nearly) evenly across 20 mutually exclusive classes.
- Amazon Product Reviews [13] - Consists of 50 classes of products or domains, each with 1000 review documents.

B. Evaluation Procedure

Traditional evaluation (closed-set) is when the classifier is assessed with data similar to what was learned during training. The number of classes presented during testing is equal to that the model was trained on. In open-set evaluation, the classifier has incomplete knowledge during the training phase. Unknown classes can be submitted to the classifier during the testing phase. During the training phase, we will train the classifiers on a limited number of classes. While testing, we then present the model with the 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. [9] [11], is a measure to estimate the open-world range of a classifier. This measure is only concerned with the number of classes used rather than the open space itself.

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 set. A True Positive is when a known class is correctly classified and a True Negative is when 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. We use the F-score as a primary metric compared to accuracy, as it takes the incorrectly classified examples (FP and FN) into consideration.

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

where C_R = number of classes to be recognized,

C_T = number of classes used in training, and

C_E = number of classes used during evaluation (testing)

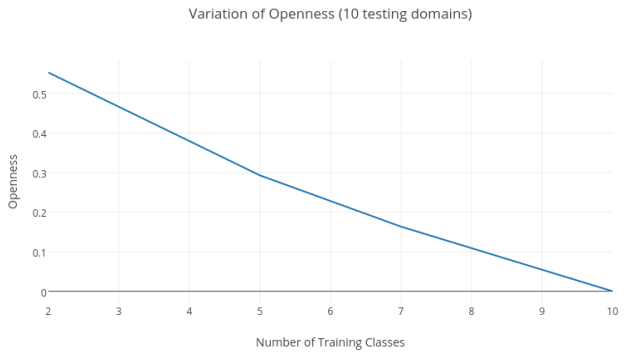

Fig. 1: Variation of openness with number of training classes

IV. Experiments

We initially experimented with distances from mean document vectors to see if they followed a Weibull distribution. We calculated document vectors by taking the mean of all word embeddings in each document. The cosine similarity between each training example and its respective mean document vector was calculated. All vectors were normalized (using Euclidean norm) to improve computation time, as vector magnitude does not affect the angle between two vectors (vector similarity). Table I shows the 5 closest cosine similarities (averaged) between 20 examples from the "comp.graphics" class to other mean document vectors. According to the data, examples from the "comp.graphics" class are more similar to "comp.windows.x", rather than the class itself. 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.

We decided to follow a CNN-based approach due to their ability of extracting useful features. For all experiments, the

TABLE I: Cosine similarities between examples of "comp.graphics" to other mean document vectors

Class	Cosine similarity
comp.windows.x	0.23269
comp.graphics	0.24248
comp.os.ms-windows.misc	0.24905
comp.sys.ibm.pc.hardware	0.25001
comp.sys.mac.hardware	0.28630

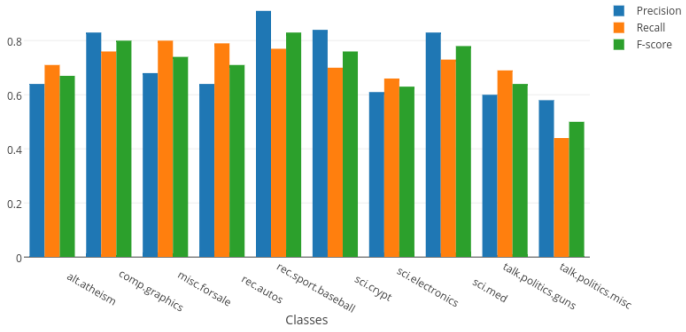

Fig. 2: l_2 constraint = 0.0
Model Accuracy: 0.710309

CNN-static architecture proposed by Kim [12] was used. We used pre-trained *word2vec* [10] vectors for our word embeddings. These embeddings are kept static while other parameters of the model are learned. According to the experiments of Zhang and Wallace [17], imposing an l_2 norm constraint on the weight vectors generally does not improve performance drastically. Figures 2, 3, 4 show the accuracies achieved on the 20 Newsgroups dataset while varying the l_2 norm constraint. The configuration details of the CNN used in all our experiments are shown in Table II. Figure 5 shows the CNN architecture we followed. In our case, we used a single static channel instead of multiple channels.

A. Ensemble Approach

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

1) Isolation Forest: An Isolation forest is a combination of a set of isolation trees. Isolation trees consist of data being recursively partitioned at random partition points with randomly chosen features. Doing so isolates instances into nodes containing one instance. The heights of branches containing outliers are comparatively less than other data points. The

TABLE II: CNN baseline configuration

Description	Values
input word vectors	Google word2vec (300 dimensional)
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

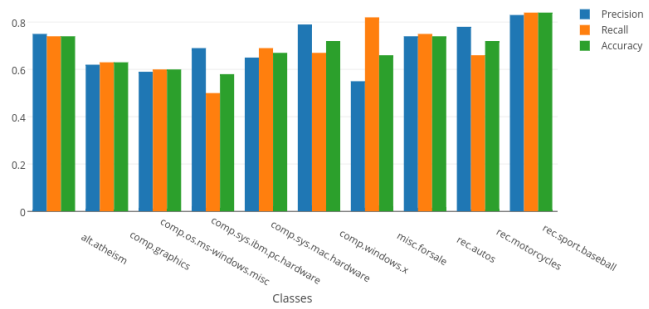

Fig. 3: l_2 constraint = 2.0
Model Accuracy: 0.688253

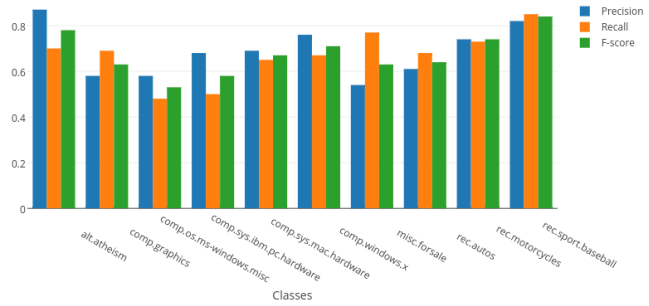

Fig. 4: l_2 constraint = 3.0
Model Accuracy: 0.672197

height of the branch is used as the outlier score. The scores obtained from the isolation forest are min-max normalized. Scores are calculated for every trained 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.

2) Probabilistic Approach: In closed-set classification, the Softmax layer essentially chooses the output class with the highest probability with respect to all output labels. This idea was extended to open-set image classification by Bendale and Boult [8]. They proposed the OpenMax, which is a new model layer that estimates the probability of an input belonging to an unknown class. OpenMax is based on the concept of Meta-Recognition [15]. 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 medoids of every trained class. For every class, the distances between the class activation vectors and the respective k class medoids are calculated. For every activation vector, we take the average of the k calculated distances. As the number of classes in our dataset is far less than those used in image classification, the k medoids of a class are used represent a class more accurately than a single mean activation vector.

In our outlier ensemble, we have used two distance metrics - Mahalanobis distance and Euclidean-cosine (Eucos) distance [8].

Ideally, we want a distance metric that can tell how much an example deviated from the class mean. The Mahalanobis distance does this by giving us a multi-dimensional generalization about the number of standard deviations a point is from the distribution's mean. The closer an example is to the distribution mean, the lesser the Mahalanobis distance. The Mahalanobis distance between point x and point y is given by:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})' C^{-1} (\vec{x} - \vec{y})} \quad (1)$$

Here, C is the covariance matrix that is prior calculated among the feature variables.

The Euclidean-cosine distance is a weighted combination of Euclidean and cosine distances. While using this metric, we do not normalize the activation vectors. Doing so decreases the vector magnitude, thereby affecting the overall distance.

The distances obtained are used to generate a Weibull model for every training class. We use the libMR [15] FitHigh method to fit these distances to a Weibull model that returns a probability of inclusion of the respective class. Figure 6 shows the probabilities of inclusion obtained from the Weibull distribution for a training class from the 20 Newsgroups dataset. As an example deviates more from the mean (k-medoids), the probability of inclusion decreases.

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

$$\text{open set probability} = 1 - \text{total closed set probability} \quad (2)$$

We used a voting scheme to combine the three approaches (Mahalanobis Weibull, Eucos Weibull and Isolation Forest). 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.

V. Results and Discussion

Open-set performance largely depends on the "unknown" classes used during evaluation. This is true especially when classes are not completely exclusive. The activation vectors of similar classes usually overlap in their vector space. Similar to [1], [2], 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

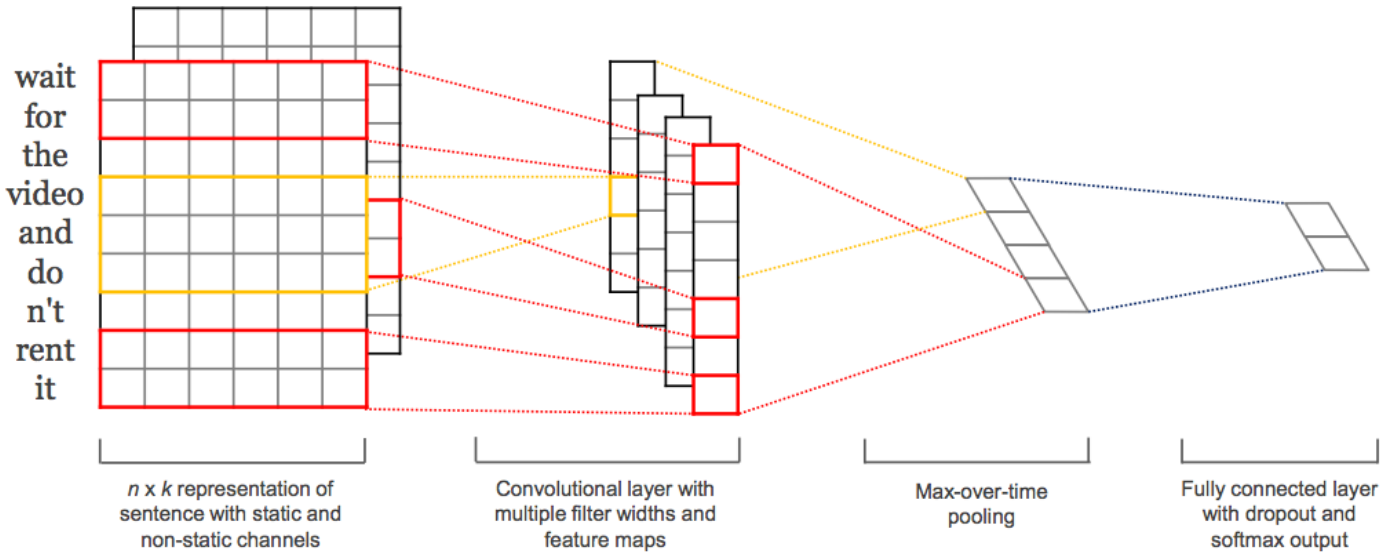

Fig. 5: Model architecture with two channels for an example sentence (image taken from [12] without permission)

TABLE III: Experiments on Amazon Product Reviews dataset and 20 Newsgroups 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*	0.45	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	0.7	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	0.59	0.662	0.731
P_i-osvm-rbf*	0.061	0.142	0.137	0.148
P_i-svm-linear*	0.6	0.695	0.701	0.705
P_i-svm-rbf*	0.245	0.59	0.718	0.774

Amazon Product Reviews	**20 Domains**			
	25%	**50%**	**75%**	**100%**
our model	**0.648**	0.603	0.663	-
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

20 Newsgroups	**10 Domains**			
	25%	**50%**	**75%**	**100%**
our model	**.719**	.747	.738	.864
NCC*	652	**.781**	**.818**	**.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	-
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

evaluated on the same number of test classes (10, 20). We also evaluate our model on smaller domains, shown in Table IV. 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 for the 20 Newsgroups and Amazon Product Reviews dataset are shown in Table III. We report only the F-scores due to space constraints. 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 the class activation vectors. Due to such distinct clusters, we believe our model performs better than other SVM based approaches in smaller domains.

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.

VI. Future Work

We are currently working on adapting our open-set classification techniques to multi-layered CNNs. This involves changing the longitudinal kernal (height x 300) to a lateral

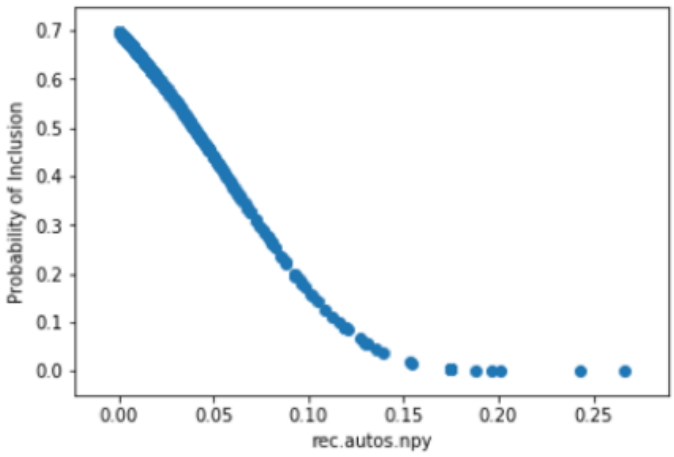

Fig. 6: Weibull distribution for the *rec.autos* class

TABLE IV: 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	.763
4	-	-	0.797

kernal (height x 1). This allows us to extract activation vectors from the antepenultimate layer which may represent the input data more accurately.

VII. 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* [1] and *nearest centroid class classification* [2]. This research can prove beneficial when classifying novel data, applications of which can be used to tackle tough text classification problems like authorship attribution and sentiment analysis.

VIII. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-1359275 and IIS-1659788. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Fei, G. and Liu, B., 2016. Breaking the Closed World Assumption in Text Classification. In HLT-NAACL (pp. 506-514).

[2] Doan, T. and Kalita, J., 2017, January. Overcoming the challenge for text classification in the open world. In Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual (pp. 1-7). IEEE.

[3] Bendale, A. and Boult, T., 2015. Towards open world recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1893-1902).

[4] Mensink, T., Verbeek, J., Perronnin, F. and Csurka, G., 2013. Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE transactions on pattern analysis and machine intelligence, 35(11), pp.2624-2637.

[5] Jnior, P.R.M., de Souza, R.M., Werneck, R.D.O., Stein, B.V., Pazinato, D.V., de Almeida, W.R., Penatti, O.A., Torres, R.D.S. and Rocha, A., 2017. Nearest neighbors distance ratio open-set classifier. Machine Learning, 106(3), pp.359-386.

[6] Gogoi, P., Bhattacharyya, D.K., Borah, B. and Kalita, J.K., 2011. A survey of outlier detection methods in network anomaly identification. The Computer Journal, 54(4), pp.570-588.

[7] Conneau, A., Schwenk, H., Barrault, L. and Lecun, Y., 2016. Very deep convolutional networks for text classification. arXiv preprint arXiv:1606.01781.

[8] Bendale, Abhijit, and Terrance E. Boult. "Towards open set deep networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[9] Scheirer, W.J., de Rezende Rocha, A., Sapkota, A. and Boult, T.E., 2013. Toward open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(7), pp.1757-1772.

[10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[11] Scheirer, Walter J., Lalit P. Jain, and Terrance E. Boult. "Probability models for open set recognition." IEEE transactions on pattern analysis and machine intelligence 36.11 (2014): 2317-2324.

[12] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[13] Jindal, Nitin, and Bing Liu. "Opinion spam and analysis." Proceedings of the 2008 International Conference on Web Search and Data Mining. ACM, 2008.

[14] Rennie, J. 20-newsgroup dataset. 2008

[15] Scheirer, W.J., Rocha, A., Micheals, R.J. and Boult, T.E., 2011. Meta-recognition: The theory and practice of recognition score analysis. IEEE transactions on pattern analysis and machine intelligence, 33(8), pp.1689-1695.

[16] Rocchio, J.J., 1971. Relevance feedback in information retrieval. The Smart retrieval system-experiments in automatic document processing.

[17] Zhang, Y. and Wallace, B., 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

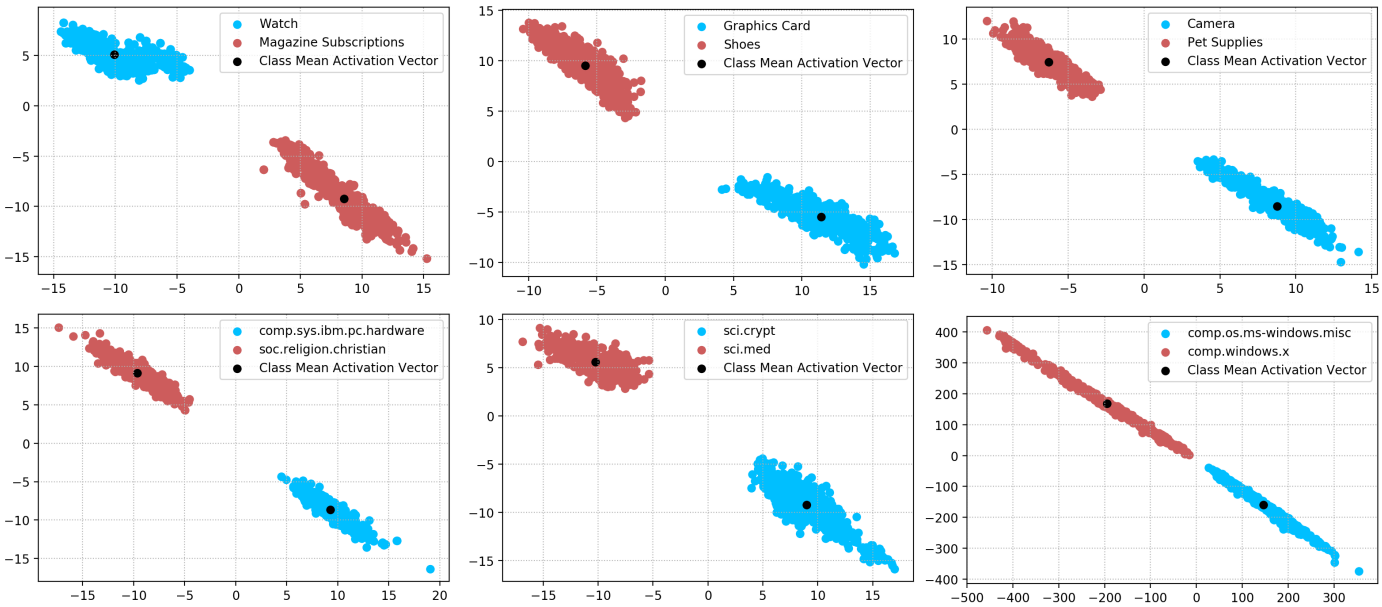

Fig. 7: Activation vectors obtained from models trained on 2 randomized classes.