Experiments with One-Class SVM and Word Embeddings for Document Classification

Jingnan Bi[1][0000−0001−6191−3151], Hyoungah Kim[2][0000−0002−0744−5884], and Vito D'Orazio[1][0000−0003−4249−0768]

[1] University of Texas at Dallas
School of Economic, Political, and Policy Sciences
[2] Chung-Ang University
Institute of Public Policy and Administration

Abstract. Researchers often possess a document set that describes some concept of interest, but which has not been systematically collected. This means that the researcher's set of known, relevant documents, is of little use in machine learning applications where the training data is ideally sampled from the same set as the testing data. Here, we propose and test several methods designed to help solve this problem. The central idea is to combine a one-class classifier, here we use the OCC-SVM, with feature engineering methods that we expect to make the input data more generalizable. Specifically, we use two word embeddings approaches, and combine each with a topic model approach. Our experiments show that Word2Vec with Vector Averaging produces the best model. Furthermore, this model is able to maintain high levels of recall at moderate levels of precision. This is valuable for researchers who place a high cost on false negatives.

Keywords: document classification · content analysis · one-class algorithms · word embeddings.

1 Introduction

Text classification is widely used in the computational social sciences [5, 10, 12, 25]. Typically, the initial corpus is not labeled, but researchers wish to classify it to identify relevant information on some concept of interest. In an ideal setting, and to use standard supervised machine learning, a researcher would randomly sample from the corpus, label the sampled documents, and use that as training data. To estimate performance on the out-of-sample set, the researcher may then sample from the out-of-sample set and label those documents for assessment purposes.

However, this ideal setting is often not what researchers face in practice. Rather, it is common for researchers to possess a document set that describes some concept of interest, but which has not been systematically collected for a supervised learning problem. Several issues arise. One, this set only contains relevant documents, making this a one-class classification problem. Two, this

set has not been systematically collected, but rather has been accumulated over time because the researcher has found information of interest from a variety of sources. Together, this means this particular document set is of little use for standard, supervised machine learning applications. Yet at the same time, systematically collecting documents, randomly sampling and labeling, and then classifying is a very time-consuming and costly effort.

In this paper, we propose and test a method to solve this problem. That is, we want to provide a method for researchers to use their arbitrarily collected, relevant document sets, to classify other documents that have been drawn from distributions that are not identical to that from which the initial set has been constructed. The central idea is to combine a one-class classifier with feature engineering methods that enrich the raw input data in ways we expect to be generalizable. Specifically, we test one topic model approach and two word embeddings approaches against a baseline of unigram tokens. The word embeddings methods that have been trained using external corpora, and we expect this to enhance the generalizability of the model to new sets drawn from unknown distributions.

To tie this more closely to the objective of computational social scientists, we tune and compare the four models using the area under the precision recall curve (AUPRC) and explore the PR curves specifically. These are relevant performance measures in this case because researchers have low cost for false positives and a high cost for false negatives, and the PR curve describes this tradeoff. That is, we want to use the known documents to identify a set of potentially relevant documents that contains as few irrelevant documents as possible. In this way, researchers can sort through the positively classified documents in search for additional relevant ones as simply as possible. The emphasis on reducing false negatives helps researchers make the case that the measures they ultimately extract from the text are not biased due to missing instances.

We test this approach using a document set on multinational military exercises, a concept of interest in the field of Political Science. Three findings emerge. First, both word embeddings approaches perform considerably better than the topic model and univariate token approaches. Second, the addition of topic features to the word embeddings models does not improve performance. Third, the Word2vec with Vector Averaging model performs better than the Word2vec with K-Means model. This model is able to maintain high levels of recall at moderate precision, and is thus promising for social scientists who place high costs on false negatives.

2 Multinational Military Exercises

Although the method is not particular to multinational military exercises (MMEs), it is important to describe the concept precisely because the method is intended to generalize to similar social concepts.

MMEs are cooperative, non-combative actions undertaken by the militaries of multiple states (countries) intended to improve future military coordination.

As with many social concepts, MMEs have evolved over time. During the Cold War, the United States and the Soviet Union would regularly conduct MMEs with partner states intended to improve warfighting capabilities and to deter one another from acts of aggression. For example, NATO's Reforger Exercise, which began in 1969, aimed at improving NATO's ability to deter a Soviet invasion [3, 4]. Since the end of the Cold War, MMEs have generally reduced in size but have proliferated in number. For example, the United States exercises with South Korea are much smaller than they were during the Cold War [6]. In addition to South Korea and NATO partners, the United States participates in MMEs with states such as Morocco, Sri Lanka, Thailand, India and Israel. The nature of the exercises has also changed over time. As exercises proliferate, they are now conducted for purposes such as counter-terrorism, coordination during natural disasters, and simulations of humanitarian crises such as food shortages and pandemics.

Although we do not specifically experiment with multi-modal sources, information about exercises comes from text, video, and audio sources. Twitter feeds, news sites with embedded video content, and official government pages describing military activity are all potential sources of information on MMEs. Generalizable algorithms, trained without excessive manual labeling, are needed for computational social scientists to sort through such diverse source materials in search of relevant information.

3 Problem Description

Our core problem is similar to that of [14], although their focus is on developing an algorithm to improve keyword searches. For convenience, we use their notation and description of task. We begin with a reference set, R, which is the set of relevant documents. In our case, these are the relevant documents that the researcher has accumulated over time. We form a search set, S, which is a new set of unlabeled documents that may or may not be drawn from the same distribution as R but which contains both relevant and irrelevant documents. The goal is to identify the target documents, T, in S. Furthermore, we have the same general criteria as [14] in that we have a high cost for false negatives.

3.1 Reference Set

Our reference set contains 1,373 news articles on MMEs that took place during 1970 to 2000 [7]. These specific articles have been twice filtered from a larger set of stories, first with an automated classification method and then manually. The initial, larger set of stories came from select news sources made available through LexisNexis, including the New York Times, BBC, and Associated Press.

3.2 Search Set

To build our search set, we queried Factiva, a news aggregator, for documents that contain specific keywords such as milit- (- is the wildcard). We also excluded

those with words such as sport in their headline and/or lead paragraph. This led us to a search set of 15,428 news articles published between 2011 to 2015. So, S was constructed from a different time period using a different news aggregator, although there was overlap in specific sources included.

To this set of 15,428, we applied to two initial cuts. First, we deleted 229 extremely long documents, a step which simplified our document processing. Second, and much more consequential, we dropped all stories that did not contain mention of at least two countries. Since our goal is to collect data on *multinational* military exercises, and story with just one country mentioned is not relevant. This heuristic process removed 5,840 articles, returning a new search set of 9,349 documents. Finally, a random sample of 606 documents in the new search set was manually labeled, which we used for validation and assessment.

3.3 Experimental Comparison

The one-class SVM is a common algorithm that has been used to identify conflict zones from event data [13], change points in time series [11], and radicalized statements on Twitter [2]. The general idea underlying the classification process is that the OCC-SVM constructs a hyper-plane with maximum margin of separation to the origin so that every data point could be classified by its signed distance to the hyper-plane [24]. To implement the OCC with Python, OCC-SVM classifier, built in Scikit-learn toolkit of Python, is used [22]. As is common with classifiers, OCC-SVM needs to be trained with sample documents and then fit on the ones in search set. For comparison purposes and to evaluate different algorithms' performance, we will show the applications of OCC with same reference and search set but different document representation methods: (1) Baseline Count-Vector OCC Model; (2) Word2Vec with Vector Averaging OCC Model; (3) Word2Vec with K-Means OCC model; (4) Topic Model with OCC.

3.4 Baseline Count-Vector OCC Model—Model1

Count-Vector OCC model serves as a baseline model. Assuming that words under similar contexts have similar meanings, word co-occurrence could be used to represent text numerically [15]. With Python machine learning library, Scikit-learn, the model could be built and estimated as follows [22]. First, it would convert documents in reference set into a vector that has a dimensionality equal to the number of unique words in all reference documents. In this study, we had a reference matrix of size 1373×8546: for each of 1373 documents in reference set, if it featured the tokens in corpus, we had a number of tokens' frequency in that dimension, leaving 0s elsewhere. Then, document-term matrix would be generated for search set, using the same vocabulary dimensionality derived from reference set. That being sad we had a matrix of size 606×8546 for our search set in this experiment. Finally, the reference matrix would be used to train the OCC classifier first and predict on search matrix.

3.5 Word2Vec with Vector Averaging OCC Model—Model2

Word2Vec is another way to represent words with numeric vectors while preserving their syntactic and semantic relationships with other words in original documents [19]. Unlike the Count-Vector model which processes at document level, Word2Vec model breaks document down to sentences as lists of words so that every word in the document set could have a vector which preserves its semantic meaning and compare its similarity with other words in the corpus. To build up this model in Python, we used an open source Python library — Gensim [23]. It allows us to set parameters which would affect model quality and computation efficiency. Specifically, we set vector dimensionality as 300, which was identified to be good enough to assure syntactic and semantic accuracy, and the minimum count threshold as 20, which cut down number of words in corpus to 2579 [18]. For example, 'exercise' as a frequently used word in our experiment, was represented with a vector of size 300, and its distance with its synonym ,'dill', could be calculated as cosine similarity, which was roughly 0.86 in this case.

Once word embeddings are generated from Word2Vec model, different methods to aggregate word embeddings to map a document to a feature class have been experimented in this study, and this is what differentiate Model 2, Model 3, and Model 4. In this model, we simply take the average of word vectors in a given document to create a single vector as a numeric representation for that document. This is intuitive and has been applied in various text classification experiments [1, 16, 21]. Take our reference set as an example. Mean vector of size 300 would be first calculated for each reference document and then mean vectors stream one by one, comprising a 1373×300 matrix as a representation for the whole reference set. Similarly, the matrix for our search set was 606×300, number of column equal to number of documents in our search set and number of rows equal to the dimensionality we set in Word2Vec model.

3.6 Word2Vec with K-Means OCC model—Model3

With word embeddings derived from the same Word2Vec model, this model takes a different method to transform the word vectors to a feature set for document, which is referred as K-means clustering method. It is an unsupervised method which assumes that semantically similar words are more commonly clustered together and by finding out the cluster centroids, the document could be represented as a bag-of-centroids [17]. Python library, Scikit-learn, also provides a viable way to achieve this. First, the optimal number of clusters, K, should be determined. In this study, we estimated the number of clusters using Elbow method, which suggested $k = 30$. Once the number of clusters was chosen, documents in both search and reference sets could be mapped in terms of their perspective distance to 30 cluster centroids. A matrix of size 1373×30 for reference set and one for search set of size 1373×30 were generated to fit the OCC model.

3.7 Topic Model with OCC—Model4

Topic models are used to find links between documents with a method of Latent Dirichlet Allocation (LDA) [9]. The LDA assumes that a document in a corpus consists of a certain number of topics. Each topic has a probability of generating several words, which can be found in the documents in the corpus. Based on the likelihood of word co-occurrence, therefore, the hidden topics are revealed. The revealed topics and their probabilities provide a thematic representation of text collections with a structure [8]. In the studies of political science, this model was used to analyze congressional speeches (text as data) and predict a conflict from the topics from news reports [20]. In our experiment, we tested for different topic numbers, ranging from 3 to 15, and chose 8 in the end. Then each document could be converted to a topic vector describes topic occurrence in that document. For example, a document in reference set can consist of 41.3% being topic 1, 57.4% being topic 7, and approximate 0% being the rest topics. Combining topic vectors together, we would have a matrix of size 1373×8 for reference set and a matrix of size 606×8 for search set. In combined models, these 8 features were appended to the word embeddings features to construct Models 5 and 6.

3.8 Training and Classification Process

A summary of the overall training and classification process is as follows:

1. Text Preprocessing: Removed stop-words and applied word stemming;
2. Document Representation: Documents are represented as matrices of real numbers using aforementioned methods. This step produces the inputs to fit the OCC-SVM model;
3. Fit OCC-SVM: Train the model using the data in R, using the 606 labeled documents in S for validation:
   - We established a grid of commonly used tuning parameter values and estimated each model for each unique set of values in that grid
   - In total, 300 models were fit for each of our 6 feature sets, from which we selected the one that performed best using the area under the precision-recall curve
4. Evaluate model performance: precision-recall curves and AUPRC were used.
5. Rerun step 2-step 4 with revised R and S: we removed all country names from the documents to prevent the models from overfitting a military exercise to a country name

4 Results

Table 1 shows the AUPRC along with the tuning parameters that were selected for each model. The findings show several different combinations of optimal tuning parameters, at least in the grid that we searched. Although we do not find major differences with and without country names, the models without countries

do consistently perform as good or better. Ultimately, Model 2, Word2vec with Vector Averaging, scores best. The performance of this model does not improve when the topic features are added (Model 5), and actually when countries names are in the sample the performance slightly decreases.

Table 1. Hyperparameter Information for Each Model

	With Countries					**Without Countries**				
	β_0	γ	**kernel**	ν	**AUC**	β_0	γ	**kernel**	ν	**AUC**
Model 1	0.1	1	sigmoid	0.1	0.64	0.1	1	sigmoid	0.1	0.64
Model 2	1	auto	rbf	0.1	0.74	1	10	rbf	0.1	0.76
Model 3	1	0.001	rbf	0.1	0.65	1	0.001	rbf	0.1	0.66
Model 4	0.1	0.1	poly	0.1	0.60	1	0.01	sigmoid	0.5	0.61
Model 5	0.1	0.1	poly	0.1	0.73	1	1	sigmoid	0.1	0.76
Model 6	1	0.01	rbf	0.1	0.60	1	0.001	rbf	0.1	0.66

Fig. 4 shows the precision recall curves for the six models. It clearly shows that the baseline and the topic model approaches, Models 1 and 4 respectively, perform much worse than the word embeddings approaches. The addition of the topic model features to the feature space, as shown in Models 5 and 6, hardly change the shape of the curve. Thus, the stand-alone Word2Vec representation is preferable as it is more parsimonious. Finally, the Vector Averaging method performs better than the K-Means methods. Importantly, this is particularly true at the high levels of recall, as the precision is considerably higher.

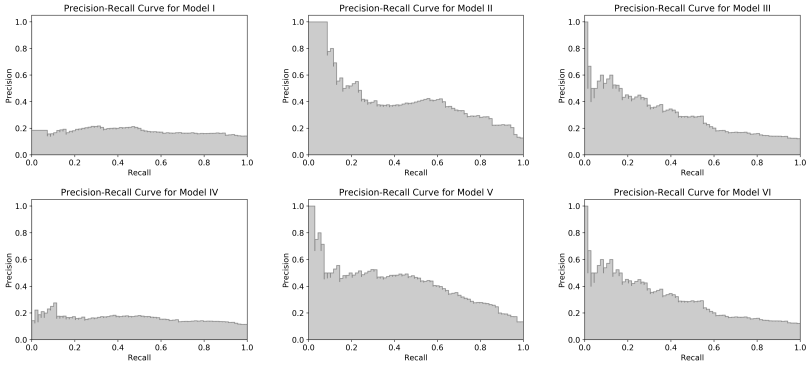

Fig. 1. Precision and Recall Curves, Models 1-6, Without Country Names

We move forward with Model 2 and show four confusion matrices in Table 2. The most interesting finding here is the table in the lower right. When we set a lower threshold value, we are able to filter 303 stories from the set of 606, and only 4 of those stories are false negatives. For researchers with the exact problem

described above, and who care about maintaining high levels of recall in their classification, this result is very promising.

Table 2. Confusion Matrices for Model II

	With Countries			Without Countries		
Default Threshold						
		Predicted: -1	Predicted: 1		Predicted: -1	Predicted: 1
	True: -1	477	60	True: -1	478	59
	True: 1	28	41	True: 1	26	43
Lower Threshold						
		Predicted: -1	Predicted: 1		Predicted: -1	Predicted: 1
	True: -1	297	240	True: -1	299	238
	True: 1	6	63	True: 1	4	65

5 Conclusions

In this paper we proposed and tested several methods to classify documents using a sample of relevant documents that did not come from the same set we wanted to classify. The central idea is to combine a one-class classifier, here we used the OCC-SVM, with feature engineering methods that we expect to make the input data more generalizable. Specifically, we used two word embeddings approaches, and combine each with a topic model approach. Our experiments show that Word2Vec with Vector Averaging produces the model. Furthermore, our results show that this model is able to maintain high levels of recall at moderate levels of precision. We believe this finding is very promising for future research into document classification methods for computational social scientists.

To develop insights for future work, we explored the model's false positives. Several stories in the false positive set were cases of single country military exercises, but where multiple countries were named. For example, a Russian military exercise where Georgia is mentioned. Future work could explore methods to identify the participants in the exercise.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No. SBE-SES-1528624. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

1. Abdelwahab, O., Elmaghraby, A.: Uofl at semeval-2016 task 4: Multi domain word2vec for twitter sentiment classification. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). pp. 164–170 (2016)
2. Agarwal, S., Sureka, A.: Using knn and svm based one-class classifier for detecting online radicalization on twitter. In: International Conference on Distributed Computing and Internet Technology. pp. 431–442. Springer (2015)
3. Blackwill, R.D., Legro, J.W.: Constraining ground force exercises of nato and the warsaw pact. International Security **14**(3), 68–98 (1989)
4. Caravelli, J.M.: Soviet and joint warsaw pact exercises: Function and utility. Armed Forces and Society **9**(3), 393–426 (1983)
5. Carley, K.M., Cervone, G., Agarwal, N., Liu, H.: Social cyber-security. In: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. pp. 389–394. Springer (2018)
6. D'Orazio, V.: War games: North korea's reaction to us and south korean military exercises. Journal of East Asian Studies **12**(2), 275–294 (2012)
7. D'Orazio, V.: Joint military exercises: 1970-2010 (2016), https://www.vitodorazio.com/data.html
8. Grimmer, J., Stewart, B.M.: Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political analysis **21**(3), 267–297 (2013)
9. Günther, E., Quandt, T.: Word counts and topic models: Automated text analysis methods for digital journalism research. Digital Journalism **4**(1), 75–88 (2016)
10. Hussain, M.N., Ghouri Mohammad, S., Agarwal, N.: Blog data analytics using blogtrackers. In: International conference on social computing, behavioral–cultural modeling and prediction and behavior representation in modeling and simulation (SBP-BRiMS 2017) (2017)
11. Jin, B., Chen, Y., Li, D., Poolla, K., Sangiovanni-Vincentelli, A.: A one-class support vector machine calibration method for time series change point detection. In: 2019 IEEE International Conference on Prognostics and Health Management (ICPHM). pp. 1–5. IEEE (2019)
12. Jin, Z., Cao, J., Guo, H., Zhang, Y., Wang, Y., Luo, J.: Detection and analysis of 2016 us presidential election related rumors on twitter. In: International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation. pp. 14–24. Springer (2017)
13. Kikuta, K.: A new geography of civil war: A machine learning approach to measuring zones of armed conflicts. Political Science Research and Methods **forthcoming** (2020)
14. King, G., Lam, P., Roberts, M.E.: Computer-assisted keyword and document set discovery from unstructured text. American Journal of Political Science **61**(4), 971–988 (2017)
15. Lebret, R., Collobert, R.: Rehabilitation of count-based models for word vector representations. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 417–429. Springer (2015)
16. Liu, H.: Sentiment analysis of citations using word2vec. arXiv preprint arXiv:1704.00177 (2017)
17. Ma, L., Zhang, Y.: Using word2vec to process big text data. In: 2015 IEEE International Conference on Big Data (Big Data). pp. 2895–2897. IEEE (2015)

18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
20. Mueller, H., Rauh, C.: Reading between the lines: Prediction of political violence using newspaper text. American Political Science Review **112**(2), 358–375 (2018)
21. Nii, M., Tuchida, Y., Iwamoto, T., Uchinuno, A., Sakashita, R.: Nursing-care text evaluation using word vector representations realized by word2vec. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 2165–2169. IEEE (2016)
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
23. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer (2010)
24. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation **13**(7), 1443–1471 (2001)
25. Zhang, F., Stromer-Galley, J., Tanupabrungsun, S., Hegde, Y., McCracken, N., Hemsley, J.: Understanding discourse acts: Political campaign messages classification on facebook and twitter. In: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. pp. 242–247. Springer (2017)