Computerization of Off-Topic Essay Detection: A possibility?

Areeba Shahzad¹ · Aamir Wali¹

Received: 19 October 2021 / Accepted: 12 December 2021 / Published online: 20 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Checking essays written by students is a very time consuming task. Besides spellings and grammar, they also need to be evaluated on their semantic content such as cohesion, coherence, etc. In this study we focus on one such aspect of semantic content which is the topic of the essay. Putting it formally, given a prompt or essay-statement and an essay, this study aims to address the problem of predicting whether the essay is off-topic or not using machine learning techniques. With an increase in online learning and evaluation platforms especially during the COVID-19 pandemic, the off-topic detection system can be very useful to check essays that are mainly submitted online. In this paper, we answer the question: given a prompt and an essay written in Pakistani English, can the process of detecting whether the essay is off-topic or not be reliably and completely autonomized with zero human intervention using currently available tools and techniques? For this purpose, we explore and implement various embedding techniques proposed in recent years to extract similarity or dissimilarity features between question and response, and compare the performance of these techniques using 10 benchmark data sets and 6 classifiers. With different classifiers and different datasets along with different embeddings, we conclude that combining word mover distance, average word embeddings and idf weighted word embeddings together and then using random forest as the classifier is the best combination for off-topic essay detection. The accuracy obtained is 93.5%.

Keywords Off-topic essay detection · Word2vec · Word-embedding-based features · Word mover's distance · Essay checking

✉ Aamir Wali
aamir.wali@nu.edu.pk

¹ FAST-NUCES, 852-B, Faisal Town, Lahore, Pakistan

1 Introduction

Traditionally, automated essay scoring tools such as Attali and Burstein (2006) have been used by testing systems such as ETS for scoring essays. The problem with such systems is that they are trained on essays limited to one specific topic and need to be re-trained for another topic on its respective sample of essays Attali and Burstein (2006). Essentially, such systems when trained, would take in an essay and determine if it is related to the essays on which it was trained.

Advances in natural language processing and machine learning have paved the way for intelligent systems that are dynamic and can predict whether or not a response is off-topic or not based on a prompt or question Tashu (2020) AlNoamany et al. (2015). What makes this problem interesting is that unlike other text prediction applications such as sentiment analysis that would take a piece of text and output a sentiment, off-topic text prediction takes in two inputs: question and response. In this case, a trained system needs to predict if the response matches the prompt.

Off-topic refers to something that has little or nothing to do with what is being discussed at present. If something is off-topic, it is not relevant to the topic or subject to be discussed. There are a variety of different cases in which you might come across this word, but it always means the same thing essentially.

Off-topic essay detection has application in education in terms of an automated checking and verification tool for teachers. In a broader context, it can also be used by students as an essay checking tool just like there are tools for checking spellings and grammar in text.

In the deep learning and machine learning paradigms, off-topic essay detection is a classification problem. In that, off-topic and on-topic essays are first human-scored and then assigned a class label 0 or 1. Then the essays, prompts, and class labels are used to train a classifier. For this process to be used practically, there is always a need for high precision from off-topic detection systems to prevent cases where off-topic essays are labeled as on-topic.

Off-topic text detection pipeline usually has the following steps. First, all text samples i.e., both the prompts and responses present in the dataset are split into words. Then they go through a process called stemming which can be crudely stated as reducing words to their stem form. For this purpose, part of speech (PoS) tagging can be used. Then each word is encoded into numeric form. For this purpose different techniques can be used such as bag of words Zhang et al. (2010), word2vec Mikolov et al. (2013), glove Pennington et al. (2014), etc. Then every corresponding prompt and response are combined using an embedding scheme such as word mover's distance Yoon et al. (2017) to extract word-embedding-based features. These features usually assess how similar or dissimilar the question and response are. Finally, these features are used to train a classifier.

There are various types of off-topic detection systems in use in the field of written text assessments and high-end stake exams such as TOEFL iBT test, IELTS test Wang et al. (2019). These are limited to one-topic learning. For wider use and dynamic environments, such as classrooms where there are ever-changing topics, there is a need for more scalable and dynamic system. Off-topic detection has many

applications in the area of classroom essay grading, fake news detection, spoken responses in a large scale, plagiarism detection and character level text classification.

In recent years, a very limited amount of work has been done on off-topic essay detection. For example, one study that used Siamese convolutional neural network (CNN) achieved a very good accuracy. However, the system was tested on just one specific data set. This raises a question which this paper attempts to answer: can the process of off-topic text detection be computerized such that it can work for *any* prompt and dataset rather than one specific prompt or dataset? Furthermore, the main issue in all related studies is that every study suggested a specific embedding for a specific classifier using a specific dataset. This makes comparison difficult. For this purpose, we explore and implement various embedding techniques proposed in recent years to extract similarity or dissimilarity features between question and response, and compare the performance of these techniques using 10 benchmark data sets and 6 classifiers.

The rest of the paper is organized as follows. Section 2 provides an overview of some method that is used in literature to detect whether an essay is off-topic or not. Section 3 presents the experimental setup that would eventually help us answer the research question. Section 4 gives the results and discussion followed by a conclusion in Section 5.

2 Literature Review

In this section we review all recent studies on on-topic and off-topic text detection since 2017. We begin with Tashu (2020) who proposed a Siamese architecture based on convolutional bidirectional gated recurrent unit (C-BGRU) for evaluation of essays automatically and reliably. The bidirectional gated recurrent unit is used as an alternative to LSTM and its main purpose is to connect two separate input sequences into the same output layer. Both the forward and backward input sequences join at each timestamp to produce the bidirectional model. The dataset consisted of essays on eight different topics submitted by students studying in grades 7, 8 and 10. For embedding, the word2vec Mikolov et al. (2013) encoding was used that is trained on 3 million words from Google news.

Furthermore, AlNoamany et al. (2015) evaluated whether a web archive has gone off-topic or not by using 5 similarity measures. This technique did not use any machine learning and relied on just the 5 measures. These measures are cosine similarity, Jaccard similarity coefficient, the intersection of the most frequent terms, web-based kernel function, and change in size. As some archived pages may contain a small amount of text, a web-based kernel function was used that focused on the semantic context of the documents rather than the lexicographical term matching.

In Yoon et al. (2017), an automated off-topic response detection system was developed for speech input. The techniques proposed for spoken responses are relevant for off-topic text detection because in all studies on spoken responses, the speech input is converted to text using an automatic speech recognition system (ASR). In Yoon et al. (2017) too, an ASR trained on non-native speech is used to

convert speech to text. After that features based on convolutional neural network (CNN) and various encodings are used to assess the similarity of the response to the question. These features are discussed next.

- Word Mover’s Distance (WMD): This feature calculates the sum of the minimum euclidean distances between words in the question and response text.
- Averaged word embeddings: This feature calculates the average of the vectors of all words in question and response separately and then calculates the cosine similarity between these two vectors.
- Idf weighted word embedding: This is an extension of averaged word embedding where the average of the word embeddings vector of the question is scaled by idf weight. The same is done for the embeddings vector of the response before calculating cosine similarity of these two scaled vectors.
- Siamese-CNN: This study also used Siamese-CNN to measure similarity between question and response. This network consists of three components. The absolute difference of the twin convolutional neural network is used as a feature to determine if the response is off-topic or not.
- Question based VSM: Another feature used in this study is vector space model (VSM). A term frequency (tf) VSM is trained using just the text of the questions.

Other studies such as Huang et al. (2018) and Lee et al. (2017) also used a subset of the features listed in Yoon et al. (2017) to access similarity between question and response for speech input and then used these features to train deep networks. Raina et al. also proposed a system to detect off-topic spoken response Raina et al. (2020) and used a similarity grid model (SGM) instead to access the question-response similarity. The similarity grid is a 2D matrix or image that represents the distance between every word of question to every word of response. For every prompt-response pair, an image or similarity grid is generated which is then used to train a CNN. A similar technique that uses similarity grid and deep networks was also proposed by Wang et al. (2019). Finally, Malini et al. (2017) proposed an off-topic spoken response detection using an attention-based bidirectional neural network (BiRNN) to embed the prompt-response pair. The embeddings are used to train a deep neural network.

3 Experimental Setup

3.1 Datasets

We used 10 publicly available datasets. The details of each dataset are given in Table 1. Essay in all of the datasets range from an average length of 150 to 750 words. Every dataset has an almost equal amount of off-topic and on-topic breakdowns and consisted of three primary columns.

Table 1 Data sets

Dataset No.	Title	Reference
Dataset 1	The Effects Computers have on People	The hewlett foundation (2019)
Dataset 2	Censorship in the Library	The Hewlett foundation (2019)
Dataset 3	Comparing lives of geese with the life of hibiscus flowers with snow	The Hewlett foundation (2019)
Dataset 4	How to be patient	The Hewlett foundation (2019)
Dataset 5	Laughter one of life's greatest gifts	The Hewlett foundation (2019)
Dataset 6	The Features of the Setting Affect the Cyclist	The Hewlett foundation (2019)
Dataset 7	Memoir is love	The Hewlett foundation (2019)
Dataset 8	The obstacles the builders of the Empire State Building faced in attempting to allow	The Hewlett foundation (2019)
Dataset 9	Bitcoin Tweets	Alex (2019)
Dataset 10	Covid19 Vaccine- People discussing Covid19 vaccine	FullMoonDataScience (2021)

- Essay Title: Titles of essay
- Essay: The ASCII texts of a response
- Labels: Labels are in binary coding of 0/1 as on-topic and off-topic.

For word vectorization we used the word2vec encoding scheme.

3.2 Classification Models

For training and classifying text as on-topic or off-topic based on prompt, we experimented with 6 classifiers. These are logistic regression (LR), CNN, random forest (RF), decision trees (DT), naive Bayes (NB) and SVM. By choosing this set of classifiers, we could experiment with an array of classifiers that include parametric, non-parametric, ensemble-based and deep learning classifiers. For

Table 2 Hyper-parameters for Random Forest

bootstrap	True
class_weight	None
criterion	'entropy'
max_depth	40
max_features	'auto'
max_leaf_nodes	None
min_samples_leaf	1
min_samples_split	2
n_estimators	100
n_jobs	2
oob_score	False
random_state	1

Table 3 Hyper-parameters for SVM

cache_size	200
class_weight	None
decision_function_shape	'ovr'
degree	3
gamma	10
kernel	'rbf'
shrinking	True
tol	0.001

CNN we used the textCNN architecture Kim (2014) which has been designed specifically for sentence classification tasks. The hyper-parameters for random forest and SVM are given in Tables 2 and 3 below. In the case of decision trees, we used the CART algorithm

3.3 Question-Response Embeddings or Features

We applied 5 embeddings on every question-response pair to create word-embedding-based features. These embeddings were word mover distance (WMD), average word embeddings (AW), idf weighted word embeddings (IW), similarity grid model (SGM) and Siamese CNN (SCNN).

4 Results and Discussion

We conducted 4 experiments. In each experiment, we used all 10 datasets and 6 classifiers. Only the embeddings or feature set were changed. In experiments 1 and 2, we only used the features obtained from WMD and SGM respectively. In experiment 3, we used 3 features namely, WMD, AW, IW. For experiment 4, we only used the SCNN feature. We split the testing and training dataset by the ratio of 30% to 70%, respectively. The performance measure is accuracy.

4.1 Experiment 1: WMD

Table 4 shows the accuracy on different classifiers and datasets when WMD was applied on the question-response pair to create a feature. The best results for each dataset are shown in bold. Naive Bayes performs better than all other classifiers in most cases and has the overall best average accuracy.

Table 4 Comparison of different classifiers using only WMD word-embedding-based feature

Data sets	Classifiers					
	LR	CNN	RF	DT	NB	SVM
1	0.834	0.808	0.829	0.743	0.837	0.836
2	0.835	0.806	0.826	0.780	0.834	0.832
3	0.853	0.850	0.865	0.826	0.874	0.868
4	0.688	0.612	0.684	0.582	0.688	0.688
5	0.631	0.597	0.637	0.597	0.643	0.624
6	0.992	0.993	0.991	0.990	0.993	0.992
7	0.999	0.999	0.998	0.998	0.999	0.999
8	0.991	0.990	0.989	0.984	0.990	0.991
9	0.918	0.910	0.916	0.860	0.918	0.918
10	0.926	0.911	0.918	0.878	0.923	0.925
Average	0.86	0.85	0.86	0.82	0.87	0.86

Table 5 Comparison of different classifiers using only SGM word-embedding-based feature

Data sets	Classifiers					
	LR	CNN	RF	DT	NB	SVM
1	0.854	0.838	0.851	0.771	0.853	0.853
2	0.937	0.933	0.936	0.902	0.938	0.938
3	0.904	0.911	0.913	0.873	0.918	0.922
4	0.688	0.694	0.735	0.631	0.662	0.729
5	0.613	0.623	0.661	0.554	0.664	0.668
6	0.995	0.994	0.992	0.992	0.995	0.995
7	0.899	0.808	0.910	0.784	0.919	0.919
8	0.995	0.994	0.993	0.990	0.993	0.995
9	0.918	0.911	0.920	0.862	0.103	0.918
10	0.917	0.907	0.916	0.853	0.917	0.917
Average	0.87	0.87	0.882	0.83	0.8	0.89

4.2 Experiment 2: SGM

Table 5 reports the accuracies when SGM is applied on the question-response pair to create a feature. The best results for each dataset is shown in bold. Naive Bayes performs better than all other classifiers in most cases and has the overall best average accuracy. The best performance obtained using only SGM in literature is 89.1% which is reported by Wang et al. (2019) using CNN. From Table 5 we can see that most of the values fluctuate around this value. Having said that, SVM performs very well than the benchmark 89.1% in 7 out of 10 cases and SVM also outperforms CNN (used in Wang et al. (2019)) in almost all cases.

Table 6 Comparison of different classifiers using WMD + AW + IW word-embedding-based feature

Data sets	Classifiers					
	LR	CNN	RF	DT	NB	SVM
1	0.891	0.895	0.898	0.846	0.890	0.867
2	0.907	0.947	0.958	0.941	0.949	0.868
3	0.906	0.952	0.966	0.960	0.941	0.904
4	0.695	0.751	0.767	0.709	0.698	0.688
5	0.812	0.910	0.889	0.885	0.832	0.802
6	0.993	0.996	0.995	0.914	0.992	0.993
7	0.912	0.891	0.929	0.962	0.931	0.900
8	0.991	0.995	0.996	0.994	0.991	0.991
9	0.918	0.914	0.923	0.881	0.931	0.918
10	0.926	0.911	0.925	0.919	0.921	0.917
Average	0.904	0.927	0.935	0.908	0.920	0.898

4.3 Experiment 3: WMD + AW + IW

Table 6 shows the accuracies obtained when WMD, AW and IW word-embedding-based features are used. As was the case in experiment 1 and 2, one of the classifiers performs better than the rest. In this case it is random forest. The average accuracies are slightly higher than the first two experiments. This is expected since three features are used to train the classifiers instead of 1. The average accuracy of RF is 0.935 which is the highest average for any classifier in all experiments. In Yoon et al. (2017) an accuracy of 0.869 was reported when all of these embeddings were used. The classifier was decision trees. Although Yoon et al. (2017) reported results using a different dataset, we provide a fair comparison, and based on the results RF seems to be a better classifier when using the WMD, AW and IW embeddings.

Table 7 Comparison of different classifiers using SCNN word-embedding-based feature

Data sets	Classifiers					
	LR	CNN	RF	DT	NB	SVM
1	0.998	0.974	0.992	0.915	0.997	0.995
2	0.967	0.967	0.96	0.91	0.974	0.973
3	0.954	0.97	0.915	0.944	0.982	0.986
4	0.783	0.768	0.812	0.697	0.771	0.809
5	0.904	0.951	0.96	0.908	0.973	0.969
6	0.991	0.981	0.991	0.944	0.99	0.994
7	0.998	0.998	0.997	0.998	0.998	0.998
8	0.927	0.919	0.923	0.886	0.927	0.928
9	0.918	0.911	0.914	0.865	0.917	0.998
10	0.917	0.906	0.907	0.865	0.918	0.996
Average	0.932	0.930	0.942	0.886	0.940	0.970

4.4 Experiment 4: SCNN

Finally, Table 7 shows the accuracy on different classifiers and datasets when SCNN is applied on the question-response pair to create a feature. The best results for each dataset is shown in bold. SVM shows best performance on 5 datasets and also has the highest overall average. Using Siamese CNN for embedding question-response pair has produced excellent average accuracies for all classifiers. In Lee et al. (2017) where SCNN was also used as an embedding technique the reported accuracy is 0.97. However, this was achieved for just one dataset. Based on our comprehensive experimentation, we achieved a highest accuracy of 0.998 for dataset 7. Also, we achieved an overall average accuracy of 0.970 for SVM.

4.5 Discussion

All the classifiers have their pros and cons. Some perform well under certain conditions, others do not. Interestingly, this also applies to datasets. For example, in all four experiments, all classifiers have performed poorly for dataset 4. Whereas all classifiers have performed very well with dataset 7 in all experiments. Dataset 4, “How to be patient” consists of 2293 essays where 1571 are labeled as on-topic and 722 are off-topic. The dataset has 5 unique prompts. On the other hand, dataset 7 has a total of 3605 essays, where 1807 are on-topic and 1798 are off-topic with 8 unique prompts for just the off-topic essays.

From the results, we conclude that using WMD, AW and IW embeddings together with random forest as the classifier is the best combination for off-topic essay detection. The accuracy obtained is 0.935. However, if computational resources are not an issue, then the best combination is to use Siamese-CNN as an embedding technique with SVM as a classifier. Coming to the research question: can the process of off-topic text detection be computerized? The answer is definitely yes but still more needs to be done. The accuracy of 0.97 indicates an error of 3 in a class of 100, but on a large scale such an error can only be acceptable if student feedback option is available.

Furthermore, if we consider the possibility of extending the reported performance beyond the specific test cases that is on big data, the classifiers used in this study should be adequate. However, a full-fledged study is warranted.

5 Conclusion

In this paper, we answer the question: can the process of off-topic text detection be computerized using currently available tools and techniques? For this purpose, we explore and implement various embedding techniques proposed in recent years to extract similarity or dissimilarity features between question and response, and compare the performance of these techniques using 10 benchmark data sets and 6 classifiers. For different classifiers, different datasets, and different embeddings, we are able to find that the best appropriate method combination that produces best results.

Based on the results, using Siamese-CNN as an embedding technique along with SVM is a good combination for off-topic essay detection. Alternately, WMD, AW and IW embeddings together with random forest as the classifier is another combination that can be used when computation cost is a factor. In all 4 experiments that we performed using different embedding schemes, the highest average accuracy achieved is 0.970 for SVM using the SCNN embedding. This certainly calls for computerization of off-topic text detection at least at a small scale, but for a large scale, computerization is possible if student feedback options are available.

5.1 Future Research Directions

For future directions, it would be interesting to experiment with various word to vector encoding schemes such as glove, word2vec, etc. BERT has also recently gained a lot of attention in NLP. For future work, BERT encoding scheme can also be used to encode text. Finally, to really understand the context and semantics of both the prompts and essays, transformers and semantic analysis tools should be used.

Declaration

Compliance with Ethical Standards This statement is to certify that the author list is correct. The Authors also confirm that this research has not been published previously and that it is not under consideration for publication elsewhere. On behalf of all Co-Authors, the Corresponding Author shall bear full responsibility for the submission. There is no conflict of interest. This research did not involve any human participants and/or animals

Conflicts of interest The authors declare that they have no relevant financial or non-financial interests to disclose. There is no personal relationship that could influence the work reported in this paper. No funding was received for conducting this study. The authors have no conflicts of interest to declare that are relevant to the content of this article

References

- Alex. (2019). Bitcoin tweets - 16m tweets. Retrieved March 15, 2021, from <https://www.kaggle.com/alaix14/bitcoin-tweets-20160101-to-20190329>
- AlNoamany Y., Weigle M.C., Nelson M.L. (2015). Detecting off-topic pages in web archives. In: *International Conference on Theory and Practice of Digital Libraries*, Springer, pp 225–237
- Attali Y., Burstein J. (2006). Automated essay scoring with e-rater [r] v. 2. *Journal of Technology, Learning, and Assessment* 4(3) Available from <http://www.jtla.org>
- FullMoonDataScience. (2021). Covid-19 vaccine tweets with sentiment annotation. Retrieved March 15, 2021, from <https://www.kaggle.com/datasciencetool/covid19-vaccine-tweets-with-sentiment-annotation>
- Huang G., Liu J., Fan C., Pan T. (2018) Off-topic english essay detection model based on hybrid semantic space for automated english essay scoring system. In: *MATEC Web of Conferences, EDP Sciences*, vol 232, p 01035
- Kim Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882)
- Lee C.M., Yoon S.Y., Wang X., Mulholland M., Choi I., Evanini K. (2017) Off-topic spoken response detection using siamese convolutional neural networks. In: *INTERSPEECH*, pp 1427–1431

- Malinin A., Knill K., Ragni A., Wang Y., Gales M.J. (2017) An attention based model for off-topic spontaneous spoken response detection: An initial study. In: *7th ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, ISCA, pp 144–149
- Mikolov T., Chen K., Corrado G., Dean J. (2013) Efficient estimation of word representations in vector space. arXiv preprint [arXiv:13013781](https://arxiv.org/abs/1301.3781)
- Pennington J., Socher R., Manning C.D. (2014) Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp 1532–1543
- Raina V., Gales M., Knill K. (2020) Complementary systems for off-topic spoken response detection. In: *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp 41–51
- The Hewlett foundation. (2019). Automated essay scoring. Retrieved March 15, 2021, from <https://www.kaggle.com/c/asap-aes>
- Tashu T.M. (2020) Off-topic essay detection using c-bgru siamese. In: *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, IEEE, pp 221–225
- Wang X., Yoon S.Y., Evanini K., Zechner K., Qian Y. (2019) Automatic detection of off-topic spoken responses using very deep convolutional neural networks. In: *INTERSPEECH*, pp 4200–4204
- Yoon S.Y., Lee C.M., Choi I., Wang X., Mulholland M., Evanini K. (2017) Off-topic spoken response detection with word embeddings. In: *INTERSPEECH*, pp 2754–2758
- Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1–4), 43–52.