HamSpam: Comment Spam Detector

Tasnim Ferdous Anan, Abdullah Ibne Masud Mahi, Tausif Khan Arnob, Tawratur Rashid Tanha
Department of Computer Science & Engineering
Islamic University of Technology(IUT)
Gazipur, Dhaka, Bangladesh
{tasnimferdous, ibnemasud, tausifkhan38, tawraturrashid}@iut-dhaka.edu

***Abstract*—Present days, people share what they feel other people might need or might be important to them by creating variety of media contents. Problem arises when some other malicious people start spam commenting on those contents. Having spam comments on those contents can bring great downfall to their career, even to their popularity. To solve this problem, we were trying to make an web and NLP based spam comment detection program. It is very difficult to keep track of every single comments of a content for spam detection as there could be thousands or millions of comments for a media content. So, keeping track of these billions and billions of media content for spam comments is really a strugling problem. For this reason, a Machine Learning based model could potentially be able to solve this problem. In this paper, we have implemented Bidirectional Long Short Term Memory (LSTM) model to solve the problem of spam commenting and to detect a comment into either spam or ham (non-spam) comment.**

***Index Terms*—spam, detection, bi-lstm, bert, naive bayes**

I. Introduction

Spam is any kind of unwanted, unsolicited digital communication that gets sent out in bulk [1]. And comment spam is when a spam bot or person leaves an irrelevant comment, sometimes also with a link to a spammy website [2]. There are also other types of spams such as- phishing emails, email spoofing, spam calls & texts etc. Comment spam has a detrimental influence on both the user experience and search engine rankings [2]. As time goes by, comment spammers have gotten better at making such comments that look genuine in order to trick website admins. In the age of social media, spam comments can be seen in almost any platforms. Sometimes those comments are so trivial that it is very difficult for other users to detect whether it is a spam comment or not. So it is very crucial to monitor comments.

Text classification is a machine learning techniques that classifies or provides tags to certain text. And those tagged data can be used to extract meaningful information by using Natural Language Processing(NLP) [3]. NLP is commonly used in text classification tasks for example detecting spam and analyzing sentiment and even classifying document [3]. This paper is about HamSpam, an NLP-based Spam Detection Web Application. In this sessional project of CSE4622: Machine Learning Lab, we used YouTube comment dataset of several popular music artists(Psy [4], Katy Perry [5], LMFAO [6], Eminem [7], Shakira [8]) to detect spam comments from their music videos' comments. We build a binary spam classification model to detect whether a text is spam or not, known as ham. We have used multiple model to train and test our data. The models we used Bidirectional-LSTM(Bi-LSTM) where LSTM stands for Long Short Term Memory, BERT(Bidirectional Encoder Representations from Transformers) & Naive Bayes Classifer. We got highest accuracy on validation set by using BERT architecture which is **89.98%**. And also we used Django framework and basic HTML & bootstrap to create our web application.

II. Literature Review

A. Spam Detection

Due to huge growth of social media as well as the spammers, spam comments can be seen almost every social media platform. We can see SMS spams, E-mail spams, common spams in social medias such as Facebook, Twitter ,YouTube. Even in LinkedIn, we can see spam comments. For this huge growth of spam comments, it is hindering users day-to-day experience as well as creating problems for owners of the product to get proper feedback from consumers.

Naive-Bayes is the most common and simple classifier for detecting spams [9]. The problem with naive bayes classifier is that it over-simplifies the problem which is not true in every cases. Still there are some literatures, that uses naive-bayes classifiers for their classification problem. For example- [10] uses Naive Bayes and other classifiers to detect SMS spam and they have shown that multinomial Naive Bayes provides 98.88% accuracy. Later on, [11] uses both CNN and Bi-lstm to detect spam reviews and they have found that in mixed-domain review, their model performs pretty good result which is 89.3% accuracy.

Then on [12], the researchers use Bi-LSTM and BERT model to classify data and compare their accuracy which is pretty similar to our project work. Later on we can see that BERT model is being used in lie detecting [13] and even in universal spam detector [14]. Those are very extensive works that can be achieved through using BERT. As time goes by, researches are inventing more BERT variations which are specialised purpose, it can be expected that there will be more work in text classification using BERT model in future.

To recap the whole review, it is clearly being seen that naive bayes classifier as being the simpliest and easy text classifier is used in literature for a certain period. Later on, as research proceeds and new models are emerging such as Bi-LSTM or BERT, those models caught researchers attention. Now transformer based model are extensively used in both

industries and academica, so transformer based model such as BERT or equivalent others are being most used in current time.

We used used three individual model to train our data. Those three models' brief description are provided below.

B. Naive-Bayes

Naive Bayes classifiers are considered as a collection of classification algorithms that is based on Bayes' Theorem [9]. The fundamental concept of Naive Bayes is that it has an assumption that each feature is independent and equal. This assumption might be not true in all aspects.

As Bayes' theorem is the main key player in this classifier, the formula of Bayes' thoerem can be written as-

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where A,B are events and $P(B) \neq 0$. P(B) is called the evidence, P(A) is the prior probability and we need to find the posterior probability, $P(B|A)$. Later on for $x_1, x_2, ...x_n$ feature vector elements the mathematical equation can be obtained as-

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

There are different types of naive bayes classifiers such as-

- Gaussian Naive Bayes Classifier: In this case, the continuous values associated with each feature is normally distributed.
- Multinomial Naive Bayes Classifier: Each feature is multinomially distributed.
- Bernoulli Naive Bayes Classifier: Each feature has bernoulli distribution.

Although naive bayes classifier has over-simplified assumptions, it worked quite well in document classification and spam filtering.

C. Bi-LSTM

Neural networks are the combination of interconnected nodes where each node does some simple calculations [15]. Now combinations of those calculation results in some numeric values that is further used in classification or detection. There are several types of neural networks- one of them is RNN or Recurrent Neural Networks. RNN uses feedback loops to remember the sequence of the data and use the data patterns to provide prediction. LSTM(Long Short Term Memory) is a special kind of RNN. LSTM can solve the long-term dependency problem by removing or adding new information. There are two types of LSTM networks.

- LSTM or Unidirectional LSTM
- Bi-LSTM or Bidirectional LSTM

In case of unidirectional LSTM, the network only stores the forward information. On the contrary, bidirectional LSTM enables the network to have the sequence information in both directions- backward and forward. This kind of network used in text classifications, speech recognition and forecasting models [15]. Figure 1 depicts the process of Bi-LSTM. Figure 1 is collected from [15].

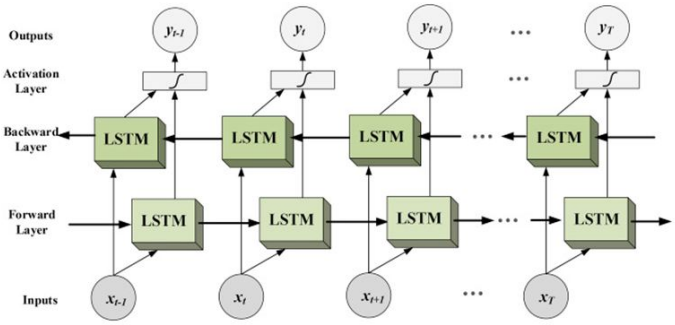

Fig. 1. Bi-LSTM

D. BERT

BERT(Bidirectional Encoder Representations from Transformers) is relatively new deep learning model introduced by Google [16] in 2018. BERT is based on Transformers which is a deep learning model [17] where every output element is connected to every input element and their weightings are dynamically calculated [18]. Other language models can only read in sequence- left to right or right to left. Meanwhile BERT can read in both directions [16].

BERT is pre-trained on two different NLP(Natural Language Processing) tasks: Masked Language Modeling(MLM) and Next Sentence Prediction(NSP). The significance of Masked Language Model (MLM) is to hide a word in a sentence and program predict what word has been hidden (masked) based on the context. The significance of Next Sentence Prediction(NSP) is to have the program predict whether two given sentences have a logical, sequential connection or whether their relationship is simply random [18]. BERT uses the process named transfer learning which is it can adapt to the growing text of searchable content and be fine-tuned.

BERT is now used at Google to optimize user search texts [19]. The functionalities of BERT can be classified as below [18]-

- Seq-to-seq tasks:
  - Question Answering
  - Text Summarization
  - Predicting Sentences
- NLU(Natural Language Understanding) tasks:
  - Sentiment Analysis
  - Word Sensing

As BERT is an open-source framework [16], there are some specialized usage of modified BERT architecture. Some of those are-

- patentBERT: This model [20] is fine-tuned to classify patents.

- docBERT: This BERT model [21] fine-tuned for classifying documents.
- bioBERT: This is pre-trained BERT model [22] for biomedical text mining.
- VideoBERT: This visual-linguistic model [23] is used for unsupervised learning of unlabeled YouTube data.
- SciBERT: It [24] is a pre-trained model for scientific text.
- TinyBERT: This model [25], proposed by Huawei, proposed a process where 'student' BERT model learns from 'teacher' BERT model.
- DistilBERT: It [26] is a smaller and faster version of BERT trained from BERT.

III. Methodology

A. Data Acquisition

The dataset used in this project is named as YouTube Spam Collection Data Set which is the collection of comments classified as spam or ham. Those comments were collected in 2015 from five popular music videos of distinct five international artists. The dataset is available in the UCI Machine Learning Repository [27]. This dataset was created and used in [28]. Number of instances of this dataset is 1956 and the number of attributes is 5.

The following table I is collected from [28] which shows the distribution of spam and non-spam samples of the dataset.

Datasets	YouTube ID	#Spam	#Ham	Total
Psy	9bZkp7q19f0	175	175	350
KatyPerry	CevxZvSJLk8	175	175	350
LMFAO	KQ6zr6kCPj8	236	202	438
Eminem	uelHwf8o7_U	245	203	448
Shakira	pRpeEdMmmQ0	174	196	370

TABLE I
DATASET USED IN THIS PROJECT

The distribution of samples is shown in figure 2.

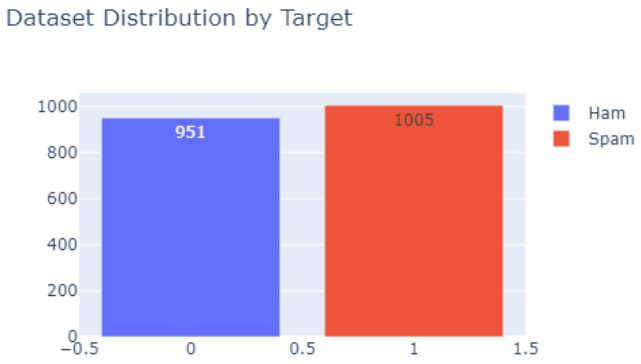

Fig. 2. Dataset Distribution By target

B. Pre-processing

As mentioned before. the dataset has 5 attributes- Comment ID, Author, Date, Content and Classified Class where 0 is represented as non-spam or ham and 1 is represented as spam. We first concatenated those five separate dataset into a single dataset and drop the attributes- comment id, author and date. Then the text are cleaned by removing punctuation marks and stopwords using nltk.corpus module. A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. The stopwords in nltk are the most common words in data. They are words that you do not want to use to describe the topic of your content. Later on, stemming is used to preprocess the data and source was nltk.SnowballStemmer(English) module. example of stemming for root word "like" include: "likes", "liked", "likely", "liking".

C. Proposed Model

We have used multiple models to train on the dataset and check their accuracy accordingly. The short descriptions of how those models are used in our project are descirbed below-

1) Naive-Bayes:

- After preprocessing the data, we vectorized the data with CountVector from sklearn.feature_extraction.text.
- Then we used multinomial Naive Bayes classifier to train the data both on training and validation set.

2) Bi-LSTM:

- Preprocessed data are being converted into tokenizeed words and source is tensorflow.keras.preprocessing.text module.
- Then the data is padded. Padded word sequence source was tensorflow.keras.preprocessing.sequence module.
- Then Bi-LSTM model is used which is a kera sequential model. The architecture is shown below.

3) BERT:

- Pre-processed data is being encoded with pre-trained bert tokenizer.
- The pre-trained tokenizer is 'bert-large-uncased' and the source is transformers.
- Then we use pretrained BERT model and the source of this model is TFBertModel module from transformers.
- The model summary is shown below-

IV. Experimental Setup

The number of instances of the dataset was 1956. And the split ratio between train set and validation set was- 75% & 25% respectfully.

We used 10 epochs and the batch size was 32. The optimizer used in this project was Adam Optimizer with a learning rate(α) $1e-4$. And the loss was calculated by binary crossentropy function.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 120, 128)	256000
bidirectional (Bidirectional)	(None, 120, 240)	239040
global_max_pooling1d (GlobalMaxPooling1D)	(None, 240)	0
batch_normalization (BatchNormalization)	(None, 240)	960
dropout (Dropout)	(None, 240)	0
dense (Dense)	(None, 120)	28920
dropout_1 (Dropout)	(None, 120)	0
dense_1 (Dense)	(None, 120)	14520
dropout_2 (Dropout)	(None, 120)	0
dense_2 (Dense)	(None, 1)	121

Total params: 539,561
Trainable params: 539,081
Non-trainable params: 480

Fig. 3. LSTM Architecture

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 60)]	0	[]
input_2 (InputLayer)	[(None, 60)]	0	[]
tf_bert_model (TFBertModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 60, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	109482240	['input_1[0][0]', 'input_2[0][0]']
dense_3 (Dense)	(None, 32)	24608	['tf_bert_model[0][1]']
dropout_40 (Dropout)	(None, 32)	0	['dense_3[0][0]']
dense_4 (Dense)	(None, 1)	33	['dropout_40[0][0]']

Total params: 109,506,881
Trainable params: 109,506,881
Non-trainable params: 0

Fig. 4. BERT Architecture

V. Result Analysis

The results that we got from three models are discussed below individually.

A. Naive Bayes

The accuracy on validation set that we get from Naive Bayes Classifier is **88.14%** which is quite good. Naive Bayes classifier achieved this sort of accuracy because of the short length of the dataset. The confusion matrix is provided in figure 5 for reference.

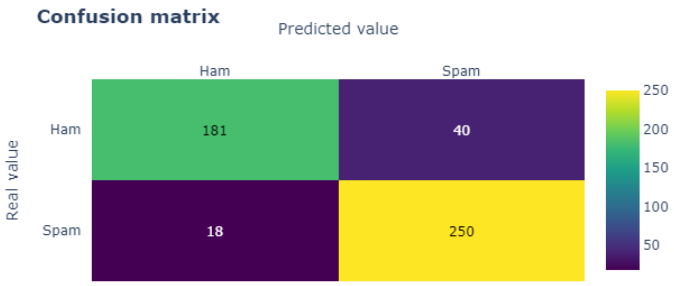

Fig. 5. Confusion Matrix generated from Naive Bayes Classifier

B. Bi-LSTM

Bi-LSTM performs better than Naive Bayes Classifier in accuracy on validation set. Bi-LSTM provided an accuracy of **89.57%** which is quite higher and satisfactory according to our expectation. The accuracy is higher because it uses bidirectional feedback loops to gain information. The confusion matrix and accuracy graph is shown in figure 6, 7 respectively for reference.

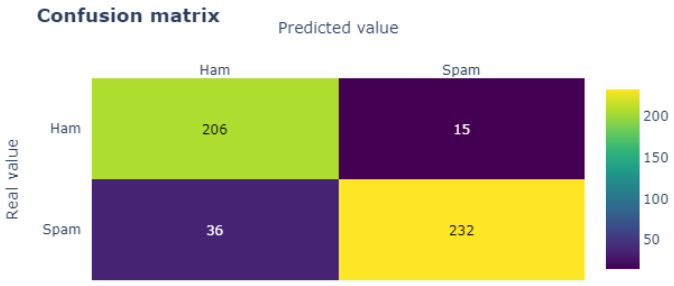

Fig. 6. Confusion Matrix generated from Bi-LSTM

C. BERT

BERT model shows the highest accuracy on validation set which is **89.98%**. BERT performs slightly better than Bi-LSTM because of the uniqueness of its architecture. BERT

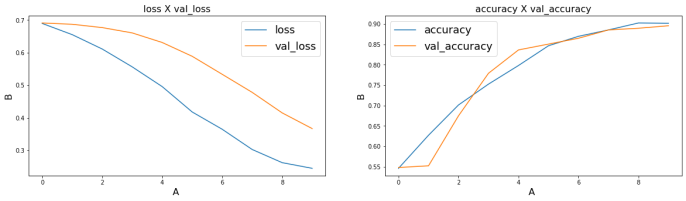

Fig. 7. Accuracy and loss graph generated from Bi-LSTM

uses transformers whereas Bi-LSTM uses bidirectional feedback loops. So it can be drawn as a conclusion that BERT performs better than Bi-LSTM [29] which also proved from our result. The accuracy and loss graph are shown in figure 8 for reference.

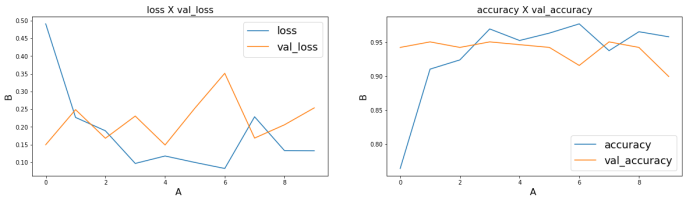

Fig. 8. Accuracy and loss graph generated from Bi-LSTM

After training those three popular models- Naive Bayes, Bidirectional LSTM & BERT, we get the following result which is presented in table II. From table II, it is clearly seen

Model	Train	Validation
Naive-Bayes	96.23%	88.14%
Bi-LSTM	90.18%	89.57%
BERT	**95.84%**	**89.98%**

TABLE II
RELATIVE ACCURACY ACHIEVED BY USED MODELS

that BERT outperforms other models. BERT performs slightly better than Bidirectional LSTM in this dataset.

VI. Conclusion & Future Work

We have used multiple models to train this dataset [27]. From the results based on the accuracy on validation set, we see that BERT model provides the highest accuracy of **89.98%**.

Due to shortage of time and huge academic pressure, we couldn't been able to work on a latest dataset which contains more samples. And [27] is the only available dataset that works on YouTube comment spam. Other datasets contains E-mail spams or Twitter spams or SMS spams. We didn't have that much enough time to create a new dataset and annotate accordingly. The only drawbacks of this project work is that the dataset we used here is too small. The other future works can be listed as-

- We will try our models on bigger datasets later on.
- The dataset we used here is short and biased on music video spam data. So our project work can be further used in that kind of datasets where all types of spam comments are available.
- We have saved our model into .h5 file. This .h5 file can be integrated into any android app that can classify spam comments.
- We can also use pre-trained models those are variations of BERT to see which variation provides more accuracy.

References

[1] What is spam? [Online]. Available: https://www.malwarebytes.com/spam

[2] What is comment spam? [Online]. Available: https://loganix.com/what-is-comment-spam/

[3] Nlp: Spam detection in sms (text) data using deep learning. [Online]. Available: https://towardsdatascience.com/nlp-spam-detection-in-sms-text-data-using-deep-learning-b8632db85cc8

[4] Psy. [Online]. Available: https://en.wikipedia.org/wiki/Psy

[5] Katy perry. [Online]. Available: https://en.wikipedia.org/wiki/Katy_Perry

[6] Lmfao. [Online]. Available: https://en.wikipedia.org/wiki/LMFAO

[7] Eminem. [Online]. Available: https://en.wikipedia.org/wiki/Eminem

[8] Shakira. [Online]. Available: https://en.wikipedia.org/wiki/Shakira

[9] Naive bayes classifiers. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/

[10] H. Shirani-Mehr, "Sms spam detection using machine learning approach," *unpublished) http://cs229. stanford. edu/proj2013/Shir aniMehr-SMSSpamDetectionUsingMachineLearningApproach. pdf*, 2013.

[11] Y. Liu, L. Wang, T. Shi, and J. Li, "Detection of spam reviews through a hierarchical attention architecture with n-gram cnn and bi-lstm," *Information Systems*, vol. 103, p. 101865, 2022.

[12] G. Xu, D. Zhou, and J. Liu, "Social network spam detection based on albert and combination of bi-lstm with self-attention," *Security and Communication Networks*, vol. 2021, 2021.

[13] D. Barsever, S. Singh, and E. Neftci, "Building a better lie detector with bert: The difference between truth and lies," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[14] V. S. Tida and S. Hsu, "Universal spam detection using transfer learning of bert model," *arXiv preprint arXiv:2202.03480*, 2022.

[15] Complete guide to bidirectional lstm (with python codes). [Online]. Available: https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] Bert language model. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

[19] Understanding searches better than ever before. [Online]. Available: https://blog.google/products/search/search-language-understanding-bert/

[20] J.-S. Lee and J. Hsiang, "Patentbert: Patent classification with fine-tuning a pre-trained bert model," *arXiv preprint arXiv:1906.02124*, 2019.

[21] A. Adhikari, A. Ram, R. Tang, and J. Lin, "Docbert: Bert for document classification," *arXiv preprint arXiv:1904.08398*, 2019.

[22] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[23] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7464–7473.

[24] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[25] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[26] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[27] Youtube spam collection data set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection

[28] T. C. Alberto, J. V. Lochter, and T. A. Almeida, "Tubespam: Comment spam filtering on youtube," in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, 2015, pp. 138–143.

[29] Building state-of-the-art language models with bert. [Online]. Available: https://medium.com/saarthi-ai/bert-how-to-build-state-of-the-art-language-models-59dddfa9ac5d