Analysis Of Written Comments On Social Media Posts For Content Similarity

Mohit Awachar - 191IT231
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
mohit.191it231@nitk.edu.in

Anshul Patel - 191IT208
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
anshulpatel.191it208@nitk.edu.in

Rakshit Kulkarni - 191IT245
Information Technology
National Institute of Technology Karnataka
Surathkal, India 575025
rakshitkulkarni2002@gmail.com

***Abstract*—Every comment on social media posts are an essential statistic for gauging the reaction of followers to the substance of the posts. However, the large number of irrelevant comments that follow each post can have an influence on many aspects of people's involvement, as well as the visibility of the post itself. Readers are more likely to pay attention to related comments to a post's topic since they give greater insight into the post's substance.In this research, we offer a framework for determining the content similarity of given comments to a post and distinguishing between relevant and unrelated written comments to the real post. Each post's relevant and unrelated comments are labelled using a machine learning-based categorization technique.So for the framework to be get work,semantic and syntactic based similarities has been used.The proposed framework has been evaluated for two languages i.e. Hindi and English, major priority being given to Hindi so to add more contribution to linguistic language based work.To test the working of framework, it has been first evaluated on English language then with required modifications it has been tested against Hindi language. For Hindi language, 3380 comments has been considered as a dataset under 92 posts from BBC Hindi news agency social media post.Model being in early phase provided accuracy of 62%. As a case study, we used the learnt classifier to a larger dataset of 6454 comments made beneath 180 posts, and we discovered that 306 of the written comments are unrelated to the content of the posts. When examining the content of both related and unrelated comments in relation to the topics of their respective posts, it becomes clear that most related comments are objective and discussion surround around the post only, whereas unrelated posts are mostly subjective without any relation about the post.**

Keywords— Facebook Scrapper, Cosine Similarity, Word-Embedding, Indic-Bert, RandomForestRegressor.

I. Introduction

People are more interested in following breaking news and stories from this platform than from the main news agencies' websites. News agencies are disseminating the news through social media such as Facebook to a large community of people. The audience nowadays is more interested in following breaking news and stories from social media platform rather than from the main news agencies websites. Following the posts of news agency pages on Facebook, users' comments are one of the most important sources of information. These comments can be factual and linked to the substance of the post, or they can be entirely or partially false and unrelated. Some famous news organisations' Facebook pages, such as the BBC News, have millions of daily visitors, and posting irrelevant comments by users can have a detrimental impact on their traffic and reader happiness.[1] and [2]. Because readers value comments as a source of additional information, they prefer to see comments that are more significant and address the issues of an article rather than unrelated notions like personal opinions, adverts, bot-generated material, and so on. As a result, social media content analysis [3] has a significant barrier in recognising such irrelevant comments after a publication.

The framework used for this project can be expected to full fill various other applications related to day to day life instances like labelling of spam or ham emails, fake news detection, similarity among YouTube comments and many more.This all can be achieved with a slight modification in existing framework. The proposed framework majorly concentrates on a linguistic language i.e. Hindi, instead of other languages so that more contribution and development can be done in this respective field.

Research have used text contents or temporal and geographical user activity in social media to detect irrelevant comments that are made to propagate spontaneous spam, influence public opinion, market products and events, and so on [3], [4], [5]. It may not have access to a post's whole story or to certain external corpuses such as Wikipedia or Google web pages connected to the post content to enrich existing short texts in some content analysis systems dealing with posts and following comments as short texts in social media. Using these types of external corpora in real-time content analysis,

TABLE I: Summary of Literature Survey

Authors	Methodology	Merits	Limitations
Xie et al. [3]	The context-aware semantics of a comment are determined by the semantic environment in which it is placed	To help in the early detection of irrelevant comments, also created a corpus	Not work for Hindi language.
Wang et al [8]	Use the LDA topic modelling technique	To group similar sentences in the same subjects and done for political spams	No major contribution in Indian linguistic languages.
Bhattarai et al [9]	Using a semi-supervised and supervised learning approach, extracting features and categorising them.	Features of spam comments were examined	Limited to only English language.
Mishne et al [10]	Construction of a statistical model for text generation using a language-based method	Identify spam comments in blogs	Not used word embedding, Limited to only English language.

on the other hand, might be time-consuming and hence have a detrimental impact on the efficiency of a real-time application.

To overcome these challenges, following works suggests it using a word embedding technique to distinguish related and unrelated comments following the posts of a news agency page on social media without referring to the complete article.It can be improve semantical characteristics to detect related and unrelated contents by using word embedding and extracting abstract semantic ideas in numerical, vector form from both pre-trained corpora and our current dataset. The major contributions of our work are:

- This is the first time written posts and comments related/unrelated analysis is done for hindi language.
- Without having access to the whole story of a post or external corpuses connected to each post's content, we apply a word embedding strategy inside semantical aspects to improve similarity identification.
- The project work employ a word embedding method inside semantical elements to increase similarity detection without having access to the entire story of a post or other corpuses associated to each post's content.
- In terms of accuracy, precision, recall, and F-measure, our experiment findings reveal that employing a mix of syntactical and word embedding-based features, our model outperforms techniques that just utilise syntactical modelling methods to detect related/unrelated contents.

The content of the paper is briefed as follows. Section II of this paper describes the problem statement and the objectives. Section III of this paper is devoted to a survey of research on the content analysis of the social media. Section IV is Methodology section which describes the framework in detail, the dataset used and all types of similarities used. Section V talks about the results and their analysis and a case study on how many comments are related or unrelated on the BBC News Hindi Page. Section VI is the Conclusion which summarize the findings of the paper and the Future Work.

II. Problem Statement

Analysis of Written Comments on Social Media Posts for Content Similarity

A. *Objectives*

- To classify the comments into related and unrelated comments for the given post
- Feature extraction of the comments based on the different similarity measures
- Compare three methods of text similarity cosine, word-to-vec and IndicBert for comment and post similarity To analyze social media post and comments based on the model built.

III. Literature Survey

Large body of research has focused on analysing user-generated material (e.g., posts, comments, and reviews) on social media, taking into account textual content as well as temporal and geographical user activity[6], [7]. Spam content is a notion that may be found in emails, websites, blog posts, and comments. Short text spam, such as spam comments left on blogs and social media sites, has gotten a lot of attention recently[8], [9].

Bhattarai et al[10] looked at the characteristics of spam comments in the blogosphere based on their content, with the goal of extracting features and classifying them using a semi-supervised and supervised learning technique.To identify spam comments in blogs, Mishne et al[11] used a language-based approach to construct a statistical model for text production. Diversionary comments, as defined by Wang et al[9], are comments on political blog postings that are intended to intentionally distract readers' attention to another issue. To replace pronouns with appropriate nouns, they employed a mix of co-reference resolution and Wikipedia embedding, as well as the topic modelling approach LDA to group similar phrases in the same subjects.

In, Xie et al.[3] suggested a context-aware technique for detecting irrelevant comments after posts. Their method believed that the semantic environment in which a comment is positioned determines its context-aware semantics. They also worked on developing a corpus of the most comparable past comments to the current posts on the same topic to aid in the early detection of irrelevant comments.

The work of the this project helps us to conclude that the model has gone above the state of the art in leveraging a mix of syntactic and semantic-based characteristics to uncover similarity between brief texts, based on all of the prior research in identifying related/unrelated comments after a post. The project helps us to do the same thing which has been done in previous research work i.e. labeling of comments similar or dissimilar for the linguistic language Hindi. In contrast to earlier research, following model does not rely on the complete story of a post or material from external web pages relevant to the post, and here it has use a word embedding strategy to enrich the short text corpus.

IV. Dataset

The dataset for this project is collected from BBC Hindi News Facebook Page. The BBC Hindi News page is used because it is the world's largest broadcast news organization. The agency has an audience from all over the world. The facebook scraper library was used to scrape comments from the News Agency Page. The scrapper is implemented in Python. All the posts and comments are stored in a JSON file for further use case. Each Caption, Comments and Replies is represented as a dictionary with the following fields.

- The caption for the post is given by the key *content*.
- The comments and replies for the post is given by the key *comments_and_replies*.
- The comment for the post is given by the key *comment*.
- The commenter name for the comment is given by the key *commenter_name*.
- The replies for the comment is a list which is given by the key *replies*.
- The replies list consists of key-value pair where key is reply and value is label (related/unrelated).
- The label for the comment is given by the key *related*.

	Training	**Testing**	**Total**
No of Comments and Replies	2379	794	3173
Percentage of split	75%	25%	-

TABLE II: BBC Hindi News Dataset

A total of **92** posts were scrapped and they had **3173** comments in them. These posts and comments are used for the training and testing purpose of the model.Hence there are a total of 3173 comments and replies with a 75-25 split for training and testing. Data description is shown in Table II.

V. Methodology

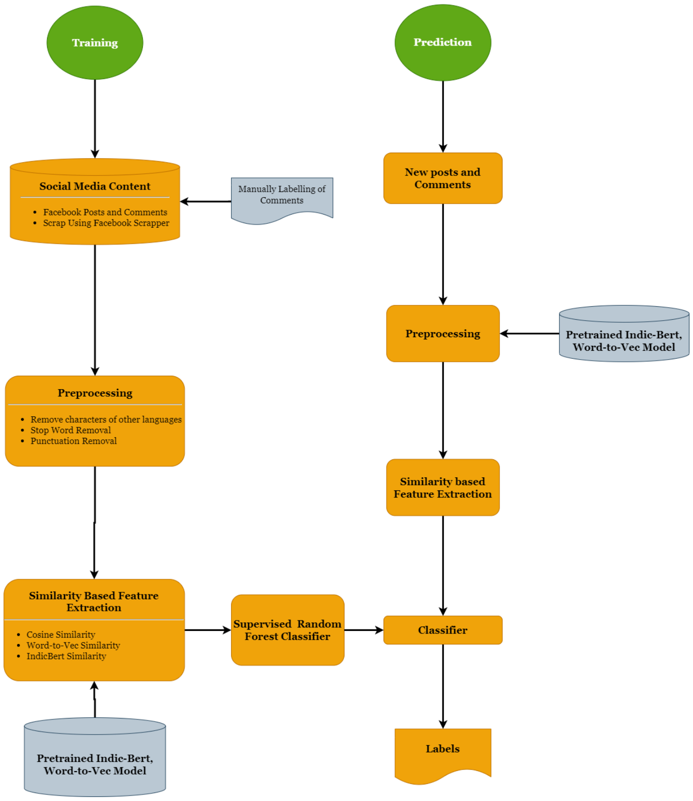

Training
Prediction
Social Media Content
Facebook Posts and Comments
Scrap Using Facebook Scrapper
Manually Labelling of Comments
New posts and Comments
Preprocessing
Pretrained Indic-Bert, Word-to-Vec Model
Preprocessing
Remove characters of other languages
Stop Word Removal
Punctuation Removal
Similarity based Feature Extraction
Similarity Based Feature Extraction
Cosine Similarity
Word-to-Vec Similarity
IndicBert Similarity
Supervised Random Forest Classifier
Classifier
Pretrained Indic-Bert, Word-to-Vec Model
Labels

Fig. 1: Framework

The proposed framework for our model is shown in Figure 1. The major goal is to categorize comments under a post as related or unrelated. The caption for each post P$_i$ and comments C$_ij$ (j = 1,2,3..) under the post are taken as input and the model has to predict if the comment is related/unrelated to the post. Since this is a classification problem, the framework is divided into main parts namely Training and and Prediction. The training part is classified into 3 major parts namely pre-processing the input text, Feature extraction based on Similarities and application of Supervised Algorithm for model construction for prediction. The pre-processing is the cleaning step where the input raw data i.e comments are caption will be cleaned using appropriate techniques. The Feature extraction based on Similarity is the heart of the project. In this step features are defined to calculate the similarity between posts and comments.

In the current framework three techniques are used in this step. They are Syntactic Similarity which is nothing but cosine similarity between the comments and the caption, Semantic Similarity which uses word-embeddings based approach and Similarity using Bert which also takes semantic meaning in consideration to calculate the degree of similarity between post(caption) and the comment. In the framework the semantic meaning of words is being considered which depends on the context of the word, its relation with other words and also the context of the word. This hybrid approach to calculate all the above similarities to predict if the comment is related/unrelated is the first attempt based on the best of our knowledge for content in Hindi language.

The framework uses pre-trained models to generate word-embeddings as well as sentence embeddings in the semantic similarity approach. More on all these is discussed later in this paper. The next step after Feature-Extraction is selection of models for training and prediction purposes. The similarities obtained in step 2 serve as the input to the model and label which is related or unrelated serve as the output. Random Forest is used to evaluate performance of the model in identifying related/unrelated comments. After training the model, the performance of the model is evaluated based on the predictions made on the testing data-set. The detailed information of methodology is discussed below.

A. *Dataset Labeling*

The project makes use of Supervised learning techniques to calculate whether the comment is related to the post or not. For Supervised learning, the target variable must be known in advance. In this case, the label of the comment is the target variable. So it is necessary to label the comments manually to evaluate if they are related to the post or not. We define comments as related when the commentators are discussing the topic of the post. The comments where commentators argue on topics related to the original post gives the reader potentially good information. Most of the unrelated comments are done merely to attract readers attention and don't have any useful information. Given below are the three main rules based on which we label a comment as related or not.

1) Comments with advertisements related to a product, or a company. These also include job advertisements or promotion of a person or a program through social media. Examples for such comments can be promotion of job applications through comment sections.
2) Comments with very little content which have very little significance to what the post says and all the comments where the user expresses his/her sentiment about the post like 'I love that' or 'I hate this', etc. These types of comments don't give readers any additional information about the post.
3) Comments in which commentators argue with each other off-topic to the post. These types of comments have generally no context to the post.
4) Comments in which the commentators give opinion about the news agency but not the post. Due to the wide variety of comments available on social media, these comments are also considered unrelated.

All comments were labeled for training data based on the above four guidelines and a dataset of **3173** comments was created. The labeling was done by the three students who are authors of this report. A comment was marked as related or unrelated based on what majority of them agreed.

B. *Pre-processing*

The comments and captions must be pre-processed before extraction of features. Since the comments are in text format they contain a lot of noise, stop words, punctuation and words in other languages. The text is preprocessed in two ways. One way is for calculation of syntactical cosine similarity and semantic similarity using word-embeddings. For these calculations removal of stop words is necessary. The other type involves similarity using the BERT model for Hindi. Since BERT takes into account the meaning of complete or complete sentences the stop words are not removed for calculation of BERT similarity. First, all words and characters which are not part of the english language are removed from comments. All text in Hindi contains stop words. These words are important for grammar of the sentence but they don't add much meaning in applications related to this project. All the stop words are removed from all posts and comments. Now all the text(comments and captions) only contain hindi words and special symbols in Hindi like punctuation. All the punctuations and special characters are also removed from the text since they don't contribute to the meaning of the sentence. Next step is word-tokenization. Word-tokenization is breaking the sentence into words. Tokenization is done using the NLTK package in Python. All the comments with length less than 3 words are discarded from further analysis. This is because the similarity scores for these will be very less. After cleaning the comments and captions, a pandas dataframe is created which contains processed caption and the comment. All the similarities will be added to this dataframe.

C. *Feature Extraction*

This is the most important step in the framework. We are using three features in this model to describe the degree of similarity between the comments and the caption. They are Cosine similarity which is based on syntactic similarity, Semantic similarity based on word embeddings which makes use of word-to-vec to describe the sentence and one more semantic similarity based on the BERT model. We examine these features to determine similarity between the post and comments following it. Given below is the detailed description for the 3 features.

1) Syntactical Similarity: : The Syntactical Similarity is a measure to calculate the similarity between two sentences based on the words present in the sentences. There are various ways to calculate syntactical similarity like jaccard similarity, cosine similarity, etc. In this model we make use of cosine similarity as a measure to calculate the Syntactical Similarity. This measure of similarity is useful if the commentators used the same words present in the caption. Here the semantic meaning of the word is not considered and only the frequency is considered to calculate the similarity.

1) **Cosine Similarity** : The cosine similarity between two sentences s1 and s2 is calculated as follows. First TF-IDF(Term Frequency- Inverse Term Frequency) is calculated for both the sentences. This is done according to the bag of words approach. Now the two sentences are represented using two vectors. The similarity between them is given by the cosine of the angle between them.

The similarity between the post P$_i$ and comment C$_{ij}$ is given by :

$$similarity(C_{ij}) = cos(P_i, C_{ij}) = \frac{P_i.C_{ij}}{|P_i|.|C_{ij}|}$$

According to Equation 1, the cosine similarity between all the comments C$_{ij}$ and their captions P$_i$ is calculated. In the data frame created above a new column cosine similarity is added and cosine similarities is inserted into that.

2) Semantic Similarity: : Semantic Similarity is a way to calculate similarity between two sentences based on the semantic meaning of the sentences. Unlike syntactic similarity which is considered with the frequency of the words in the sentence semantic similarity focuses more on the meaning of the words present in the sentences. This semantic aspect helps to capture similarity even if synonyms are used. For semantic similarity, two approaches are considered which are given below.

1) **Word-Embedding based Similarity** : The word-embedding based approach calculates similarity between two sentences by constructing a vector for the sentence. In this approach for each word present in the sentence a vector is calculated which represents the word. For calculation of the vector, a pre-trained model from the website[1] is used. The pre-trained model called FastText is used to get word-embedding.For efficient learning of word representation Facebook Research team has created FastText library. This library has became the most popular in NLP community and may be this is a possible substitution to the gensim package which provides the functionality of Word Vectors etc. This library has more benefits over Word2Vec and Glove. Not only common/special words it also helps get vector representations rare words also. Main advantage of this library is if any word not present in the dictionary, the word will be broken down into character n-grams and it will give the vector representation. But the word2vec and glove both will fail here. All vectors calculated from this approach have 300 dimensions. To represent a complete sentence with these vectors we calculate the mean of all the vectors. Now both captions Pi and Cij are represented using vectors. The similarity between them is calculated using the cosine similarity formula mentioned in Equation 1. After calculation of cosine similarity of all comments and posts they are added in a column named semantic similarity (Word-Embeddings) in the dataframe.

2) **IndicBERT Similarity** : IndicBERT is a multilingual ALBERT model trained on a large corpus of Indian languages. Bidirectional Encoder Representations from Transformers(BERT) is used to convert a complete sentence into a vector embedding. The BERT model is trained in such a way so as to learn contextual embeddings for words. IndicBERT model which has support for Hindi Language in this project. All pairs of captions Pi and comments Cij are first converted into vector form. IndicBert Similarity of simply BERT similarity is defined as the cosine of the angle between the vector embeddings obtained by the IndicBert model. After calculation of BERT similarity this is also added to dataframe under the column BERT similarity.

D. Model Architecture for Training and Testing

This is the final stage of the framework where the features obtained from various types of similarities can be used to predict if the comment is related to the post or not. The dataframe in this framework has datapoints and three features. The target varible is the label of the comment. Then data has been divided into two groups of dataset i.e. training and testing dataset. 25% (794 comments)of the data is used for testing and 75%(2379 comments) is used for training. On analysis of the data points, it is observed that data is imbalanced in this case and the number of related comments is far more than that of unrelated comments as shown in Fig 2.

Out of total comments **3173** are **2212** related and **961** are unrelated. The imbalance data can lead to low accuracy score making the model performing poorly. The data imbalance is solved by using a technique called **SMOTE** which stands for Synthetic Minority Oversampling Technique. SMOTE is an algorithm that performs oversampling by creating synthetic data points based on the original data points to balance out the data. After using SMOTE Random Forest Classifier for training and testing of dataset. Training dataset is only used for data augmentation by SMOTE. Before application of SMOTE the distribution in training dataset was 1663:716.

[1] https://fasttext.cc/docs/en/crawl-vectors.html

After SMOTE the distribution changed to 1663:1496. This is illustrated in Fig 2. and Fig 3.

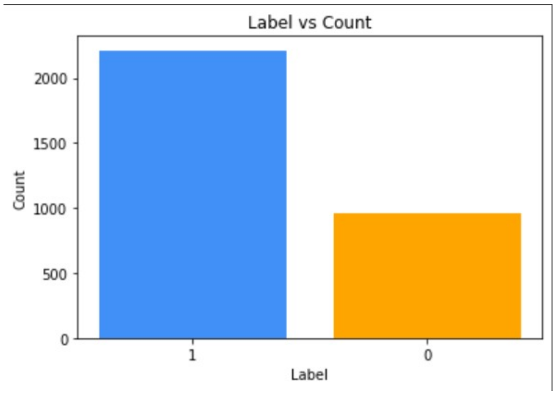


Fig. 2: Data Visualization Of Comments before SMOTE

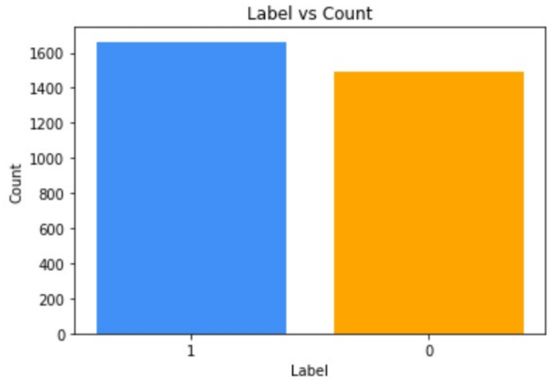


Fig. 3: Data Visualization Of Comments after SMOTE

1) Random Forest Classifier: Random Forest comes under Supervised learning algorithm which basically follows ensemble learning method and many Decision Trees. The technique used in random forest is bagging it fits multiple models on different subsets and then it will combine all the predictions from all models. In case of Random Forest the model randomly selects subsets of features.
The performance of the model is discussed in the Result and Analysis section.

VI. Results and Analysis

The result and Analysis section is divided into two parts. In the first part the performance of the model is discussed. The second part talks about analysis of comments on the BBC News Hindi page based on the model built in this project.

A. Analysis of Model

Table III shows the performance evaluation. The proposed model when used with combination of all three features i.e. Cosine Similarity, Word-To-Vec similarity and IndicBert Similarity gives obtains an accuracy score of 62% on average and it outperforms all other combination of features.

On evaluation of model based on only one factor the following things are observed. Cosine similarity performs the poorest with an accuracy of 52% and precision of 39%. This is followed by IndicBert model which have accuracy of 53% but better precision. The Word-To-Vec model performs the best individually with an accuracy of 57%.

To show necessity of combining three features the results are examined by dropping one feature at a time and the following results are observed. Dropping one of the feature has no major impact on the accuracy which indicated that all features are equally important for the model.

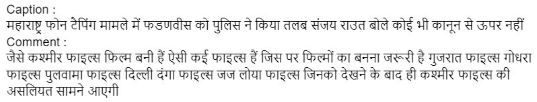
Analysis of Scores				
	Accuracy	**Precision**	**Recall**	**F1 Score**
All features	0.62	0.73	0.72	0.73
W/O Cosine	0.60	0.68	0.73	0.70
W/O Word-To-Vec	0.62	0.70	0.73	0.71
W/O IndicBert	0.61	0.68	0.74	0.70
Just Cosine	0.52	0.39	0.80	0.53
Just Word-To-Vec	0.57	0.60	0.73	0.66
Just IndicBert	0.53	0.58	0.69	0.63

TABLE III

B. Case Study

The learned classifier using the Random Forest is applied on the dataset scrapped for the Case Study. The dataset contains 180 posts and had a total of 3380 comments. According to the classifier 6454(95%) were related to the posts and 306(5%) comments were unrelated. In the original dataset about 70% of the comments were related to the posts and 30% were unrelated. This raises an important observation that distribution of related and unrelated comments depends on the comments and may vary with content of the post. The presence of unrelated comments biases lot of readers and their perspective on the posts. This creates a lot of noise on the user feedback.

Fig 3. shows an example of unrelated comment with respect to a given post. In the post an issue related to Maharashtra politics is being discussed but the commentator is trying to divert the conversation to Kashmir files movie.



Caption :
महाराष्ट्र फोन टैपिंग मामले में फडणवीस को पुलिस ने किया तलब संजय राउत बोले कोई भी कानून से ऊपर नहीं
Comment :
जैसे कश्मीर फाइल्स फिल्म बनी हैं ऐसी कई फाइल्स हैं जिस पर फिल्मों का बनना जरूरी है गुजरात फाइल्स गोधरा फाइल्स पुलवामा फाइल्स दिल्ली दंगा फाइल्स जज लोया फाइल्स जिनको देखने के बाद ही कश्मीर फाइल्स की असलियत सामने आएगी

Fig. 4: Unrelated Comment

VII. Conclusion

Using the content of the comments, a model has been created to detect related and unrelated comments to the respective Facebook postings. Syntactical and semantical are the two main categories of characteristics in the framework. For syntactic cosine similarity and for semantic word-embeddings based word-to-vec and IndicBert model has been used. These similarities have been employed to obatain a mix of word embeddings in semantic characteristics to be independent of the whole story of a post or other webpage contents connected to the post. The findings reveals that the algorithm can accurately detect related and unrelated comments submitted to the posts with a rate of 62%. Following that, if anyone looks at the distribution of related/unrelated comments throughout the articles, as well as the most commonly discussed themes in each cluster. This contributes to a better understanding of the phenomenon of social media remarks that are unconnected.

The future work is to analyze at the percentage of connected and unrelated comments across many categories on Facebook, such as politicians, celebrities, and businesses. Also thought of adding more linguistic or regional languages, a part from Hindi so that more contributions and development can also be added to indigenous languages. The unrelated comments can be further classified based on content variety and look for machine-generated remarks in unrelated content. Finally, deployment of this model for all social media platforms like Youtube, Instagram, Twitter is the goal.

References

[1] M. Mozafari, R. Farahbakhsh and N. Crespi, "Content Similarity Analysis of Written Comments under Posts in Social Media," 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2019, pp. 158-165, doi: 10.1109/SNAMS.2019.8931726.

[2] P. Rajapaksha, R. Farahbakhsh, N. Crespi, and B. Defude, "Inspecting interactions: Online news media synergies in social media," CoRR, vol. abs/1809.05834, 2018. [Online]. Available: http://arxiv.org/abs/1809.05834

[3] S. Xie, J. Wang, M. S. Amin, B. Yan, A. Bhasin, C. Yu, and P. S.Yu, "A context-aware approach to detection of short irrelevant texts," in IEEE International Conference on Data Science and Advanced Analytics (DSAA), Oct 2015, pp. 1–10

[4] N. C. Dang, F. De la Prieta, J. M. Corchado, and M. N. Moreno, "Framework for retrieving relevant contents related to fashion from online social network data," in Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection. Springer International Publishing, 2016, pp. 335–347.

[5] H. Liu, J. Han, and H. Motoda, "Uncovering deception in social media," Social Network Analysis and Mining, vol. 4, no. 1, p. 162, Feb 2014.

[6] A. Suarez, D. Albakour, D. Corney, M. Martinez, and J. Esquivel, "A data collection for evaluating the retrieval of related tweets to news articles," in Advances in Information Retrieval, G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds. Springer International Publishing, 2018, pp. 780–786.

[7] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD, 2013, pp. 632–640.

[8] J. Wang, C. T. Yu, P. S. Yu, B. Liu, and W. Meng, "Diversionary comments under blog posts," ACM Trans. Web, vol. 9, no. 4, pp. 18:1–18:34, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2789211

[9] A. Bhattarai, V. Rus, and D. Dasgupta, "Characterizing comment spam in the blogosphere through content analysis," in 2009 IEEE Symposium on Computational Intelligence in Cyber Security, March 2009

[10] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement," in AIRWeb, 2005.

[11] J. Wang, C. T. Yu, P. S. Yu, B. Liu, and W. Meng, "Diversionary comments under blog posts," ACM Trans. Web, vol. 9, no. 4, pp. 18:1–18:34, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2789211

[12] G. Fei, A. Mukherjee, B. Liu, M. Hsu, and M. C. et al, "Exploiting burstiness in reviews for review spammer detection," in ICWSM, 2013.