Email classification via intention-based segmentation

Sanjay Kumar Sonbhadra
Department of IT
IIIT, Allahabad
Prayagraj, India
rsi2017502@iiita.ac.in

Sonali Agarwal
Department of IT
IIIT, Allahabad
Prayagraj, India
sonali@iiita.ac.in

Mohammad Syafrullah
Program of master in CS
Universitas Budi Luhur
Jakarta, Indonesia
mohammad.syafrullah@budiluhur.ac.id

Krisna Adiyarta
Program of master in CS
Universitas Budi Luhur
Jakarta, Indonesia
krisna.adiyarta@budiluhur.ac.id

***Abstract*—Email is the most popular way of personal and official communication among people and organizations. Due to untrusted virtual environment, email systems may face frequent attacks like malware, spamming, social engineering, etc. Spamming is the most common malicious activity, where unsolicited emails are sent in bulk, and these spam emails can be the source of malware, waste resources, hence degrade the productivity. In spam filter development, the most important challenge is to find the correlation between the nature of spam and the interest of the users because the interests of users are dynamic. This paper proposes a novel dynamic spam filter model that considers the changes in the interests of users with time while handling the spam activities. It uses intention-based segmentation to compare different segments of text documents instead of comparing them as a whole. The proposed spam filter is a multi-tier approach where initially, the email content is divided into segments with the help of part of speech (POS) tagging based on voices and tenses. Further, the segments are clustered using hierarchical clustering and compared using the vector space model. In the third stage, concept drift is detected in the clusters to identify the change in the interest of the user. Later, the classification of ham emails into various categories is done in the last stage. For experiments Enron dataset is used and the obtained results are promising.**

***Index Terms*—Concept drift; Intention-based Segmentation; Part of speech(POS) tagging; Vector space model; Hierarchical clustering; Spam.**

I. Introduction

The first ever spam email was delivered on 3 May 1978 through ARPAnet where about 2600 ARPAnet users were sent a message by a marketer Gary Thuerk [1]. This incident demonstrated the power of electronic mail as an advertising platform to the world but introduced a new malicious activity known as spamming. The tremendous growth in the usage of email for personal and commercial use, spam filtering becomes the most important pre-requisite of any email system. Conventionally, spam is considered as an unsolicited commercial email. The importance of these spam email depends on the overall interest of the users too. Hence, a strategic spam filter is essential that can filter the emails with the consideration of users interest. Meanwhile, judicial laws have also been made to fight against the spam activities such as CAM SPAM Act in USA [2].

Many spam filtering techniques have been published during the last two decades [4], but because of technological advancements, spammers found various alternate techniques of spamming and the situation becomes worst even after several continuous safety efforts. The spam filters developed during the last two decades can identify the spam emails which contains extra spaces, HTML tags, links of malicious web pages etc., but they do not consider the noise or presence of concept drift in the emails [3]. Naturally, the interests of a person change frequently, so the class of the email also changes for a user with time. For example, during placement season, interests of the students are towards the subjects which are useful in their placements so during that time emails related to those subjects will be hams for the students but after the placements, interests of the students change and they are more interested in emails related to entertainment, games etc. So the emails which were spam at the time of placements will no longer be spams after placements. This changing effect of interest has not been considered in most of the spam filtering tools. There are a wide variety of spam techniques present on the web as shown in Fig. 1 but at counterpart, the spammers smartly develop new alternate techniques against the spam filters as they come into use [4].

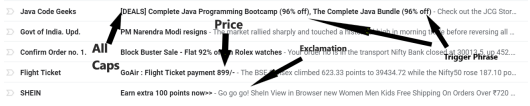

Fig. 1. Techniques Used by Spammers

To classify an email as spam or ham, this paper proposes a novel method based on intention based segmentation. Spam emails are usually commercial emails which are of less interest to the user or contain some malware content, whereas the ham emails are of more interest (current interest) to the user. Conventionally, the whole email is processed to predict its nature, whereas the proposed intention based segmentation approach works on the segments of the content of the email. Ideally, processing of the segments individually reflects more information about the content, which results in higher accuracy. There are two possible ways of segmentation: random and rule-based. Random selection of segments does not add descriptive information so not suitable for segmentation. Hence, the segments should be generated based on the context of the sentence. In this approach, whenever the context of the

sentence is changing, it represents a boundary point between two segments and content should be divided at this point for further processing. After dividing the content into segments change in the content of emails with time is detected. This change is referred as concept drift. Topics present in spam or ham emails change with time for a particular user. These changes can be gradual or abrupt or recurring. For example, the user will have different interests in different weathers, so these type of changes are called recurring change. Students going from school to college will have a gradual change in his/ her interests is an example of gradual concept drift. If a student got failed in any class, these type of abrupt changes also affects the interests of the user known as sudden concept drift.

Along with finding the class of emails, the current interests of the user can be identified by detecting concept drift in the emails, which helps to access the ham emails easily. Generally, the email contains gradual or recurring concept drifts, whereas abrupt concept drifts are found in very less volume. There are different methods to detect different types of concept drifts. For detecting sudden drifts, statistical hypothesis tests are used. Gradual concept drift can be detected using early drift detection method. There are methods based on probability and graph to detect recurring and incremental concept drift [5].

The rest of the paper is organized as follows: in section II, existing works in the field of spam detection and concept-drift detection are explained, whereas section III describes the proposed methodology for concept drift detection in Email dataset. The experimental results and performance analysis are covered in section IV. The last section contains concluding remarks and future scope of the proposed work.

II. Related Work

In past decades, several machine learning based spam filtering approaches. A wide variety of techniques have also been used to find the concept drift in emails to build adaptive spam filters. Ruano et al. [1] proposed a concept drift based methodology to detect the spam messages. In this approach, email messages are divided into two types: spam and ham. Spam messages are defined as messages of less interest whereas ham messages are those which are of more interest to the user. Set of grammar rules are defined and a finite state machine (FSM) is created to calculate every type of concept drift. In this approach, every FSM has five states in which starting with ready state, two states are intermediate states and two states are concept drift found and not found (final states). For every set of input state, transitions are recorded and the stack is updated. After the input string ends, the current state of the machine is recorded. If it is a non-final state, then possible types of concept drift are checked. If it is in the *'not found'* state then no concept drift found and if it is in the *'found'* state, then the specific type of concept drift is found for which the finite state machine is generated. To handle various HTML tags and useless spaces, tokenization is performed using Perl's regex libraries. The content of the email is represented in Java script object notation (JSON) format, which is easy to read.

TABLE I
Features and communication Means

Tense	present	past	future
Subject	i/we	you	it/they/(s)he
Style	interrogative	negative	affirmative
Status	active	passive	-
Part of speech	verb	noun	adjective

The JSON format represents the file in a tree form, from which content extraction is very easy with the help of regex libraries. For finding the drift in the concept, a database of 11700 topics is used. A topic is defined for every email body from this database. Then every email is labelled with the topics included in the body of the mail. These set of topics are used as the input set for state machine transition.

Papadimitriou et al. [6] proposed a method for finding the posts which are of interest to the user. Initially, a forum is filled by the users in which reference posts are presented. Based on the likes or dislikes on the reference posts, other posts are compared with the reference posts and given rank. Comparison of the posts is not done as whole. Every segment of each post is considered, e.g. who has posted that content, what is the topic of the content of the post, what are the hashtags, is there any picture in the post, who are tagged in that post, and content in the post is also divided into segments. To divide the email content into segments, greedy approach is applied where each word is taken as a segment and then in each iteration, border with the worst score is removed. Table I tells about the features and communications means used for segmentation.

Border is defined as a point where two different segments meet, whereas the score of the border is defined as a difference of the concept between the left and right segments of the border. Borders having less score compared to a pre-defined threshold are removed from the content. After removal of all the borders, the segments of the body of the post are merged with the other segments of the post. The response of the user to the posts which have appeared before on users timeline is also recorded. Based on this response, a score is calculated and used as the rank of the post, which is used to improve the prediction system.

Afterwards, Harel et al.[7] proposed a concept drift detection method based on the change in concept for some defined hypothesis. Authors defined a methodology which only detects sudden and gradual concept drifts but does not detect any false changes in the concept. The computation complexity of the method depends on the choice of the hypothesis. For example: in a book store, books are given ratings based on the reviews given to them. If the dictionary of ratings contains bad, good, informative the change from 'fantasy' to 'educational' books will be easily detected because both are correlated to word 'informative'. If the dictionary of ratings contains bad, good, kindle change to kindle will not be easily detected because it is not correlated with the other sentiments. During the detection process, a sub-sequence is taken and changes are evaluated based on some hypothesis. In this research work, they considered two hypothesis: one is equality of the average on sequential test segments, whereas

the other is a substantial difference between the two segments. The hypothesis is chosen based on the type of change to be detected. For detecting abrupt concept drift, a sub-sequence of the data is obtained. Then this sub-sequence is divided into two parts: training window and testing window. Then using some hypothesis change is detected in the testing window. If no change appears in the testing window, then this is added to the training window and a new testing window is introduced. The change is detected by analyzing the loss values of testing window and loss values of the same sized window obtained by the reshuffling of the sub-sequence. At each iteration, risk involved in test-train split is also calculated with a value called permutation loss.

Later, to solve the problem of spam emails, Lu et al. [8] proposed concept drift detection technique, represented as a model (x,y), where x is the feature vector and y is data stream label. For drift detection, a two-sliding-windows model is used, where each window consists of the data points taken from the two infinite random distributions. This method not only ensures the absence of concept drift, but also highlights the local regions where drift occurs. One strategy to detect the changes is through data distribution, where two data samples are compared and checked that they are from the same distribution or not. There are different tests to determine the relation among the samples of different distributions. Authors proposed another concept drift detection strategy is to find the concept drift with the help of learners output. This model traces and controls the error rate of the online learning algorithm, where the errors are defined as samples of data found from the Bernoulli trials and generalized using binomial distribution. A significant change in the error indicates the change in class distribution and the concept drift is detected.

Later, Russell et al. [9] suggested a methodology based on feature extraction to classify the documents into categories. This method uses two metrics to classify a document: popularity and rarity. If a topic is given, the feature set of the topic will contain a set of popular words and a set of rare words which come under the topic. Authors tested their algorithm under a wide range of development centric topics. Instead of using the text similarity to compare the documents, authors used feature extraction algorithm because of the presence of topic ambiguity in the documents, which gave very less accuracy. This approach takes the topics which tend to be very focused, and also take care of the words and symbols present in local or regional languages so that documents do not lose their authenticity. It was observed that to extract the features of a topic, the combined use of popularity and rarity metrics gave higher accuracy compared to the use of either of them. In the first step, the focused topics are defined. Focused topics are those which have following properties:

- The topic is not present on the web with a high frequency.
- Document overlap with other topic should be negligible if not null.

In the next step, documents containing very less information and the popular terms present in the documents are removed, because they create noise for the data. After that, the frequency for every term is calculated using the Linguistic data consortium dataset. Further, TF-IDF value of the term is calculated using the following formula:

$$tfidf(t) = tf(t) * \log(N/N(t)) \tag{1}$$

Here, tf(t) is mean term frequency of t, N is total no. of documents and N(t) is no. of documents in which term t appears. A popular term is the term which satisfy the following constraints -

$$tfidf(t) > T_{th}, \quad LDC(t) < P_{max} \tag{2}$$

Here, T_{th} represents the lower bound of TF-IDF values for popular terms, and P_{max} represents the upper bound of count of popular terms. A rare term is the term which satisfy the following constraints:

$$tfidf(t) > R_{th}, \quad LDC(t) < R_{max} \tag{3}$$

Here, R^{th} represents the lower bound of TF-IDF values for rare terms, and R_{max} represents the upper bound of count of rare terms.

In the next step, a set of popular and rare terms is given as input to the feature extraction algorithm which will give a two-dimensional vector as output containing a feature vector of the term and weighting function for each of the terms. These vectors are given input to any standard classifier, which gives the topic of the given document. In their research work, Naive Bayes and support vector machine classifiers are used.

Many techniques related to the topic under consideration have been proposed earlier. The detailed literature survey shows that each technique has its limitations and drawbacks. Also, none of the techniques considers all the important factors of concept drift. The literature shows that the presence of concept drift in emails plays a very important part in finding the class of the emails due to noise present in the content of the emails whereas the presence of HTML tags, price tags, extra spaces etc. in the content put a significant effect on finding the class of emails. Meanwhile, the intention based segmentation analysis indicates that instead of comparing the two documents as a whole, segment analysis can give better results. Whereas, it is evident that feature extraction algorithm gives higher accuracy compare to the other normal classification algorithms. Classification algorithms should be applied to the features after feature extraction instead of directly applying to the documents. It is also found that instead of comparing the documents with text similarity TF-IDF scheme should be used due to the presence of noise in the documents.

III. Proposed Methodology

This paper proposes a novel spam filter model to detect spam emails via the concept drift occurred in the content. Categorization of the ham emails has also been done, which makes it easier for the user to access them. In the proposed model, the classification of emails is performed in six steps, where at every step, content of the email is processed and compared with the rest of the previous emails. By analyzing

the comparison between the emails, concept drift is detected and the class of the emails is decided. Most importantly, emails containing malware content are also taken care of and, an email some pre-defined specific keywords is identified as spam emails. Fig. 2 shows the proposed methodology and description of the steps is as follows:

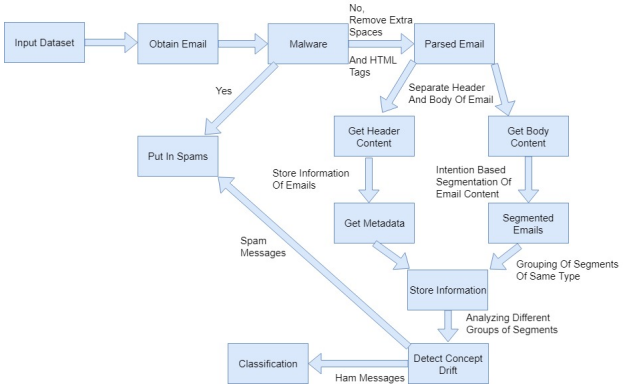

Fig. 2. Proposed methodology

A. Checking for malware content

This is the first step towards finding the real class of the emails. In this step content of the current email is checked. If it contains some specific keywords stored in the database which represents the malware content or if it contains the hyperlink which are related to the malware content present on the web then the email is directly put into the spam emails and no further process is done on that email.

B. Parsing emails

The second step is parsing of the emails. In this, email is parsed i.e. all the information related to the email is stored in a different variable, for example: date, sender, subject, content. After storing the content in a different variable, it is further processed for the further processing. Extra spaces, HTML tags, price tags, exclamation symbols etc. are found in the content, and these things are removed from the emails. Hyperlinks present in the content are also found and removed. The main idea behind removing the extra spaces, HTML tags, price tags and exclamation marks is that these are used by spammers to make a difference in the content of emails so that spam filter recognise them as different and do not mark them as spams [1]. Removal of these words makes it easier for proposed spam filter to focus on the content mainly. The idea behind removing the hyperlink is that once the hyperlink is processed in the previous step no further processing is required [10].

C. Content segmentation

In this step, email content is divided into segments that involves following three stages: Tokenization, Tagging and Classification (as shown in Fig. 3).

a. Tokenization: It is the process to divide the email content into tokens and sometimes the difference between token and type has to be recognized. A token is defined as a group of characters which have a particular meaning in some particular document, whereas a type is defined as the class of a group of characters having the same character sequence [11].

b. Tagging: Tagging is the process where all words present in the document are marked with a particular part-of-speech. This process is known as POS (part-of-speech) tagging or grammatical tagging or word-category disambiguation. A Word is marked based on it's definition and context i.e. how it is related to its adjacent or similar words in the document. POS tagging is a similar process of identification of nouns, verbs, adjectives, adverbs, etc. [12].

c. Classification: Classification is performed based on tenses (present, past or future) or voices (active or passive). If a sentence contains modal verbs then it is categorized as future tense, whereas if it contains verb and past or past participle tense then it is categorized as past tense. If the sentence contains verb and present or present participle tense, then it is claimed to be in present tense [9].
Voice of the sentence can be determined by removal of all the available adverbs. If the sentence has preposition or subordinating conjunction and after first word if the sentence has wh-determiner, wh-pronoun, possessive wh-pronoun, wh-adverb as part of speech tags then the sentence is in passive voice. If the sentence has 'be', 'am', 'is', 'are', 'was', 'were', 'been', 'has', 'have', 'had', 'do', 'did', 'does', 'can', 'could', 'shall', 'should', 'will', 'would', 'may', 'might', 'must' and verb of past participle as part-of-speech then the sentence is also in passive voice. If in place of verb of past participate after one word the sentence has verb of past tense, verb of present tense with third person singular, verb of present tense with non-third person singular or after two words it has verb of present participle tense, verb of past tense, verb of past participle tense, verb of present tense with non-third person singular, verb of present tense with third person singular, base form of verb as part of speech tags, then it is in active voice. If the sentence is too short (less than two words) it is assumed in active voice [13].

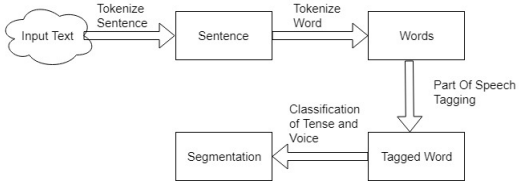

Fig. 3. Segmentation of content

D. Grouping of Segments

In this phase, groups of obtained segments are formed using hierarchical clustering (Fig. 4). Initially, each segment is taken as an individual cluster and later the difference between clusters is calculated. The clusters having difference less than a pre-defined threshold are merged [14]. The difference between

the clusters is calculated via cosine similarity and euclidean distance using vector space model.

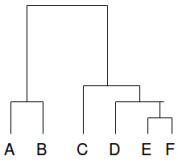

Fig. 4. Hierarchical Clustering

Two clusters are merged repeatedly when the value of $|\cos\theta|$ is less than 0.3 and euclidean distance between them is less than a pre-defined threshold. Fig. 5 shows the cosine similarity between two documents [15]. The newly created cluster is represented by a new vector. Similarity formula between two documents is as below:

$$sim(d_1, d_2) = \frac{\vec{V}(d_1).\vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} \quad (4)$$

Euclidean distance [16] between two vectors is calculated using following formula:

$$D = \sqrt{\sum (V_i - V_j)^2} \quad (5)$$

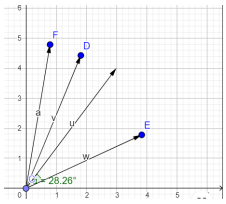

Fig. 5. Cosine similarity

E. Finding concept drift in emails

The final number of generated clusters represents the quantity of topics. If all the segments of an email are present in the clusters in which number of segments are less than a threshold, then that email is marked as spam email, and rest of the emails are put into the ham emails category. Segments made in the third step are mapped with the emails and stored. There are two advantages this approach: firstly, it avoids unnecessary processing of duplicate emails and secondly, this approach ease the process of finding segments related to an email, hence reduces the computation cost. In experiments, step 3 to 5 are performed by comparing the email with the stored emails of last 30 days (the window size is 30 days). As the stored segments and clusters are used for processing, it makes the process fast because only the current email has to be processed and put into the cluster.

F. Classification of ham messages

In this step, ham messages are divided into different categories, which makes it easier for the user to access them. Different classification algorithms are used to classify the messages. The first step of each algorithm is to detect the features from the documents (messages). After feature extraction, classifiers are trained and samples are tested. The features used are ngram_range and TF-IDF values [16].

- ngram_range: ngram_range refers to the size of the n-gram which should be considered by the classifier if it is considering bigrams, trigrams etc. The more the size of the n-gram, the more complex words the classifier can handle.
- TF-IDF value: TF-IDF value refers to the term frequency in the document. It can be calculated by dividing the count of the term with total no. of terms in the document.

Dataset of "20newsgroups" is used to train the classifier. This dataset contains 20000 news articles, and these are in the folders of respective categories. Twenty-three categories have been defined in this dataset. For every news article present in the database, the content of the news article and its category is extracted. Then the TF-IDF value for every term present in the news article is found, and classifier is trained based on the TF-IDF values of the terms. TF-IDF values are used instead of the count of the terms because it creates a problem when there is a big difference in the size of documents. Suppose a document has 10000 words and contains 2000 related to a topic and representing that topic. And there is one more document in the dataset which contains 1000 words and contains 200 related to the same topic. So, if word count is taken as parameter, the classifier does not predict its category correctly [15]. Grid-Based Search method for parameter tuning is used. Unigrams and bigrams have been considered for our classifier, and TF-IDF value threshold is from the range 0.5 to 1. In parameter tuning for every combination of parameters, the classifier is tested, and accuracy is calculated. The set of parameters which gives best accuracy is considered in the classifier. Alongside changing the values of parameters, different classifiers and values can also be tested, which will give the best possible combination. This process takes more time as every set of parameters has to be tested but gives better results compare to the other classifiers [17]. Fig. 6 tells about all the categories in which ham messages are divided.

IV. Result and Discussion

A. Dataset Description

The proposed algorithm has been tested on two datasets: Enron email dataset, and personal emails obtain from a gmail account as a stream by using python libraries. Description of both datasets is as below:

a. Enron email Dataset: The Enron email dataset was published by the "Federal energy regulatory commission" during its Enrons collapse investigation. This dataset contains approximately 50,000 emails of the Enron Corporation and has two fields:

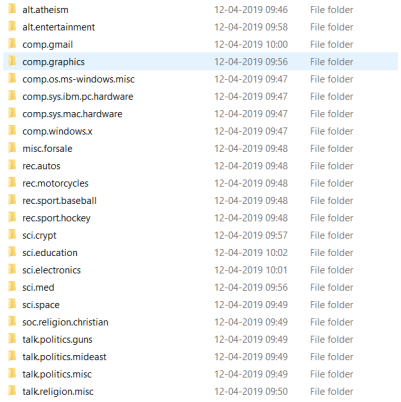

Fig. 6. Ham messages categories

TABLE II
RELATION BETWEEN PARAMETER VALUES AND ACCURACY

n-gram size	TF-IDF Value	NB accuracy	SVM accuracy
1	0.1	72%	75%
1	0.2	75%	77%
1	0.3	78%	79%
2	0.1	79%	80%
2	0.2	82%	83%
2	0.3	89%	90%
3	0.1	77%	79%
3	0.2	79%	80%
3	0.3	81%	83%

- file
- message

File contains the name of the file and directory in which email is present. The name of the user to which an email belongs is the root level directory of the corresponding emails. Message contains the content of the email. Everything related to the text is present in this column [18].

b. Emails obtained from gmail account: These emails are obtained from a gmail account using python libraries. These emails come in the form of stream, where all information is in form of string. All the parameters of an email (i.e. sender, receiver, date, message) can be obtained by using regex libraries [19].

B. Experimental Evaluation

Initially, emails are tokenized and meta-data of emails are collected. Meta-data contains the information of sender, receiver, etc. After collection of meta-data, malware contents are checked to declare an email as spam. In the next step, stop words and extra spaces are removed from the emails to reduce the noise.

For dividing the content based on tenses and voices, helping verbs present in the sentence and their positions with respect to the subject of the sentence are analyzed and current email segments are stored for future reference. Segments of the content look as shown in Fig. 7.

After extensive experiments, Naive bayes gives the accuracy of 89%, whereas SVMs are a set of supervised learning methods used for classification, regression and outliers detection, achieved the accuracy of 90% with proper parameter tuning.

```
I have finished my work.
=>Present Tense
=>Passive Voice

.....................................................
I have known him for a long time.
=>Present Tense
=>Passive Voice

.....................................................
I have been working for ten years.
=>Present Tense
=>Active Voice

.....................................................
I did not sleep well.
=>Past Tense
=>Active Voice

She was weeping bitterly.
=>Past Tense
=>Active Voice

.....................................................
It was raining all night.
=>Past Tense
=>Active Voice

.....................................................
Mahatma Gandhi died on 30 jan 1948.
=>Past Tense
=>Active Voice

.....................................................
At that time he had been editing the newspaper for two years.
=>Past Tense
=>Active Voice
```

Fig. 7. Segmentation of email contents

The relation between parameter values and accuracy is shown in Table II.

C. Validation of Concept Drift

After dividing the emails into different categories, a histogram of number of emails is generated with time for each category. These histograms show how the presence of the emails in different categories is changing, which tells about the change in content of the emails. For number of bins and bin-width of histogram, Sturges formula [21] is used which is shown below:

$$bins = ceil(log_2 n) + 1 \tag{6}$$

$$binwidth = \frac{max(values) - min(values)}{ceil(log_2 n) + 1} \tag{7}$$

Here, n is number of emails present in the category for which histogram is generated. Fig. 8 shows the histogram generated for a category (graphics) after obtaining the results.

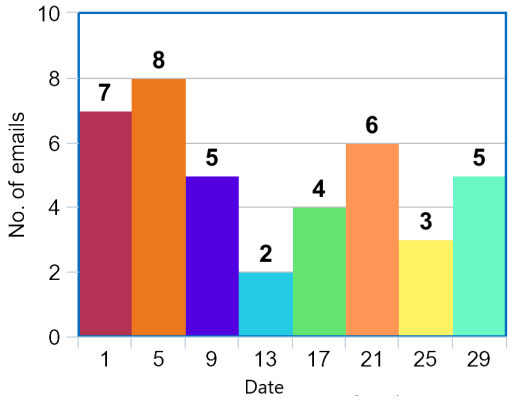

Fig. 8. Change in content of emails with time

V. Conclusion and future Work

In this paper, a method based on intention-based segmentation is proposed to classify an email. Two approaches have been used to determine the intention of sentences: tense (past, present, future) and voices (active and passive voice) based. These two methods are very closely related to the syntax and grammar of English language. To compare two documents vector space model is used. Sliding window technique is used to detect the concept drift in email datasets that helps to give real class of emails with better accuracy in less time. The emails are compared by their segments of the same intention, instead of comparing them as a whole. This approach takes every small detail of the emails and ensures good performance. The concept drift in emails describes the interests of the user. There are methods other than tense-based and voice-based segmentation which can be used for content segmentation that can also be used for text summarization and other text-based processes. The objective of the paper is to classify an email with minimum computation resource, hence Naive bayes and SVM are used for classification. Other advance machine learning approaches can be used in future to achieve higher accuracy with proper parameter tuning but need

References

[1] Ruano-Ords, D., Fdez-Riverola, F. and Mndez, "Concept drift in e-mail datasets: An empirical study with practical implications" Information Sciences, 2018, 428, pp.120-135.

[2] Bhowmick, Alexy and Hazarika, Shyamanta. (2018). E-Mail Spam Filtering: A Review of Techniques and Trends. 10.1007/978-981-10-4765-7_61.

[3] W.A. Awad and S.M. ELseuofi,"Machine Learning Method for Spam Email Classification", Internation Journal of Computer Science and Information Technology, 2011, vlo. 3, No. 1.

[4] Saad, Omar M. and Darwish, Ashraf and Faraj, Ramadan. (2019). A survey of machine learning techniques for Spam filtering.

[5] Y. Sun, Z. Wang, Y. Bai, H. Dai and S. Nahavandi, A Classifier Graph Based Recurring Concept Detection and Prediction Approach 2018 Computational Intelligence and Neuroscience, China, pp. 1-13.

[6] Papadimitriou D., Koutrika, G., Velegrakis Y. and Mylopoulos, " Finding Related Forum Posts through Content Similarity over Intention-Based Segmentation" IEEE Transactions on Knowledge and Data Engineering, 2017, 29(9), pp.1860-1873.

[7] M. Harel, K. Crammer, R. El-Yaniv, S. Mannor, Concept Drift Detection Through Resampling International Conference on Machine Learning, Beijing, China, 2014, pp. 1324-1334.

[8] Lu, N., Zhang, G. and Lu J., Concept drift detection via competence models International Conference on Artificial Intelligence, 2014, China, pp.11-28.

[9] R. Power, J. Chen, T. Karthik, L. Subramanian, Document Classification for Focused Topics International Conference on Machine Learning, 2013, NY, USA, pp. 123-136.

[10] S. Roy, A. Patra, S. Sau, K. Mandal, S. Kunar, An Efficient Spam Filtering Technique for Email Account American Journal of Engineering Research, 2013, pp. 63-73.

[11] R. J. Passonneau and D. J. Litman, "Intention- based segmentation: Human reliability and correlation with linguistic cues in Proc. Annu.Meet. Association Comput. Linguistics, 1993, pp. 148155.

[12] Jeffrey C. Reynar, "Topic Segmentation: Algorithms and Applications", University of Pennsylvania, AUGUST 1998.

[13] D. M. Blei, A. Y. Ng, M. I. Jordan, "Latent dirichlet allocation" J. Mach. Learn. Res. vol. 3 pp. 993-1022 2003.

[14] L. Weng et al. "Query by document via a decomposition-based two-level retrieval approach" Proc. 34th Int. ACM SIGIR Conf. Res. Development Inf. Retrieval pp. 505-514 Jul. 2011.

[15] F. C. Heilbron, V. Escorcia, B. Ghanem and J. C. Niebles, "Modern Information Retrieval" Addison-Wesley Longman Publishing Co., pp. 961-970, 2007.

[16] Cha, Sung-Hyuk; Yoon, Sungsoon; and Tappert, Charles C.,"Enhancing Binary Feature Vector Similarity Measures" (2005). CSIS Technical Reports. 18.

[17] K. Vasa, "Text Classification through Statistical and Machine Learning Models: A Survey", International Journal Of Engineering Design Research, 2016, vol. 4.

[18] V. Bobicev, "Text classification: The case of multiple labels," 2016 International Conference on Communications (COMM), Bucharest, 2016, pp. 39-42.

[19] Kaggle "https://www.kaggle.com/wcukierski/enron-email-dataset" [Accessed March 20, 2020]

[20] Jeffrey "http://jeffreyfossett.com/" [Accessed march 20, 2020]

[21] Scott, David. (2009). "Sturges' rule". Wiley Interdisciplinary Reviews: Computational Statistics. 1. 303 - 306. 10.1002/wics.35.