International journal of application or innovation
in engineering and management

Ijaiem.com/Mar 2025/ Volume 14/Issue 1/Article No-1/223-225

ISSN: 2319-4847

SPAM DETECTION USING TEXT CLUSTERING

Yethirajam Yuganand Sree[1], Bomma Prashanth [2], Dovoor Shiva Kumar[3], V. ChandraPrakash [4]
[1,2,3] UG Scholar, Department of Information Technology, St. Martins Engineering College, Secunderabad, Telangana, India, 500100
[4]Assistant Professor, Department of Information Technology, St. Martins Engineering College, Secunderabad, Telangana, India, 500100

Article Info

Received: 28-03-2025 Revised: 05 -04-2025 Accepted: 16-04-2025 Published:27/04/2025

Abstract:

We propose a new spam detection technique using the text clustering based on vector space model. Our method computes disjoint clusters automatically using a spherical k-means algorithm for all spam/non-spam mails and obtains centroid vectors of the clusters for extracting the cluster description. For each centroid vectors, the label('spam' or 'non-spam') is assigned by calculating the number of spam email in the cluster. When new mail arrives, the cosine similarity between the new mail vector and centroid vector is calculated. Finally, the label of the most relevant cluster is assigned to the new mail. By using our method, we can extract many kinds of topics in spam/non-spam email and detect the spam email efficiently. In this paper, we describe the our spam detection system and show the result of our experiments using the Ling-Spam test collection.

Keywords : Spam detection, clustering, non-spam, Ling-Spam detection.

1. INTRODUCTION

In recent years, spam email or more properly, Unsolicited Bulk Email (UBE) is a widespread problem on the Internet. Spam email is so cheap to send that unsolicited messages are sent to a large number of users indiscriminately. When a large number of spam messages are received, it is necessary to take a long time to identify spam or non-spam email and their email messages may cause the mail server to crush. To solve the spam problem, there have been several attempts to detect and filter the spam email on the client-side. In previous research, many Machine Learning(ML) approaches are applied to the problem, including Bayesian classifiers as Naive Bayes[1, 3, 7, 11], C4.5[10], Ripper[4] and Support Vector Machine(SVM)[6, 9] etc. In these approaches, Bayesian classifiers obtained good results by many researchers so that it widely applied to several filtering softwares. However, almost approaches learn and find the distribution of the feature set in only the spam and the non-spam messages. Today, there are many type of spam email, for example, advertisements for the purpose of making money or selling something, urban legends for the purpose of spreading hoaxes or rumors the inboxes and construct the spam folders and inspam.

HTML mails contains web bug which is a graphic in an email message designed to monitor who is reading the message. Therefore, some of spam mails are judged to be non-spam email even if we use the existing filtering techniques. In this research, we propose a new spam detection technique using the text clustering based on vector space model. This method construct the spam detection model by the contents of various kinds of mail and find spam more efficiently. The system computes disjoint clusters automatically using a spherical -means algorithm[5] for all spam/non-spam mails and obtains centroid vectors of the clusters for extracting the cluster description. For each centroid vectors, the label('spam' or 'non-spam') is assigned by calculating the number of spam email in the cluster. When new mail arrives, the cosine similarity between the new mail vector and centroid vector is calculated. Finally, the label of the most relevant cluster is assigned to the new mail. By using our method, we can extract many kinds of topics in spam/nonspam email and detect the spam email efficiently. In this paper, we describe the our spam detection system and show the result of our experiments using the Ling-Spam[1, 2, 12] test collection. HTML mails contains web bug which is a graphic in an email message designed to monitor who is reading the message. Therefore, some of spam mails are judged to be non-spam email even if we use the existing filtering techniques. In this research, we propose a new spam detection technique using the text clustering based on vector space model. This method construct the spam detection model by the contents of various kinds of mail and find spam more efficiently. The system computes disjoint clusters automatically using a spherical -means algorithm[5] for all spam/non-spam mails and obtains centroid vectors of the clusters for extracting the cluster description. For each centroid vectors, the label('spam' or 'non-spam') is assigned by calculating the number of spam email in the cluster. When new mail arrives, the cosine similarity between the new mail vector and centroid vector is calculated. Finally, the label of the most relevant cluster is assigned to the new mail. By using our method, we can extract many kinds of topics in spam/nonspam email and detect the spam email efficiently. In this paper, we describe the our spam detection system and show the result of our experiments using the Ling-Spam[1, 2, 12] test collection. In this section, we propose a new spam detection technique using the text clustering based on vector space model. In this method, the system automatically construct the spam detection model by the contents of various kinds of mail and find spam more efficiently. To obtain the spam detection model, we use the clustering algorithm called the spherical -means algorithm[5] for all the received mail. This algorithm divide the mail set into the predefined number of clusters. For each clusters, cluster centroid vectors are calculated as Cluster Representative. By obtaining the clusters, Similarity calculation between a new mail and the clusters can be performed easily. In the previously proposed methods such as Naive Bayes classifier and SVM filter, contents of spam are represented as one term statistic. However, using our method, the contents of various kinds of mail are represented as several term statistics as the centroid vectors. By obtaining the centroid vectors,the label('spam' or 'non-spam') is assigned by calculating the number of spam mail in the cluster. If the ratio of spam mail to all mail in the cluster is higher than the ratio which consisted of 70% to 85%, we consider a cluster as spam. Thus, a set of clusters can be partitioned into spam and non-spam clusters. When we obtain centroid vectors of spam and non-spam clusters, the system judges whether a new mail is spam. First, new received mail is transformed into the vector in

2. LITERATURE SURVEY

The number of people using mobile devices increasing day by day. SMS (short message service) is a text message service available in smartphones as well as basic phones. So, the traffic of SMS increased drastically.

The spam messages also increased. The spammers try to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information, etc.

So, spam classification has special attention. In this paper, we applied various machine learning and deep learning techniques for SMS spam detection. we used a dataset from UCI and build a spam detection model.

Our experimental results have shown that our LSTM model outperforms previous models in spam detection with an accuracy of 98.5%. We used python for all implementations

3. PROPOSED METHODOLOGY

The methods based on edge drawing performs line segment detection **Proposed Methodology of Spam Detection Using Text Clustering**
Spam detection using text clustering involves grouping similar text messages into clusters, with the goal of identifying those that exhibit characteristics of spam. The proposed methodology begins with data preprocessing, where raw text messages are cleaned and normalized. This step includes removing stop words, special characters, and punctuation, followed by tokenization, stemming, or lemmatization to reduce words to their root form. Additionally, techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) are employed to convert the text into a numerical format suitable for clustering, allowing for more efficient processing and feature extraction.

Next, unsupervised machine learning algorithms are applied for clustering the text data. Common clustering techniques include K-means, DBSCAN, and hierarchical clustering, which group similar messages based on their features. The number of clusters and the nature of the features are chosen based on the distribution and variability in the text data. For instance, K-means clustering can be used to identify predefined groups (e.g., spam or non-spam), while DBSCAN may help in detecting outliers that do not fit well into any cluster, which could potentially be spam messages.

After clustering, the model identifies which clusters are predominantly composed of spam messages. This is typically done by analyzing the content of the messages in each cluster, checking for common characteristics such as the use of certain keywords, frequent promotional terms, or suspicious links. The next step involves validating these clusters, where a subset of labeled data is used to verify the accuracy of the identified clusters, ensuring that the method is correctly distinguishing between spam and non-spam messages.

Finally, once the model is trained and validated, it can be used for real-time spam detection. New incoming messages are processed through the same text preprocessing pipeline, and clustering algorithms are applied to assign the messages to the appropriate clusters. If a message is placed into a cluster predominantly associated with spam, it is flagged as spam. Continuous evaluation of the clustering model is essential to refine its accuracy and adapt to evolving spam tactics over time.

To enhance the effectiveness of spam detection using text clustering, additional features and advanced techniques can be incorporated. One common enhancement involves incorporating **semantic analysis** alongside traditional text-based features. While methods like TF-IDF capture the frequency and importance of terms, incorporating **word embeddings** such as Word2Vec or GloVe allows the model to capture the meaning and contextual relationships between words. This can improve the model's ability to detect spam messages that may use varied phrasing but convey similar intent.

Another important aspect of spam detection is **handling imbalanced datasets**, as spam messages are often much less frequent than legitimate messages. Techniques such as **oversampling**, **undersampling**, or using **anomaly detection** methods can be applied to address this issue. Additionally, **active learning** can be employed, where the model selectively requests human feedback on uncertain classifications to improve the model iteratively. This approach can help refine the clustering process and improve accuracy with fewer labeled examples.

Evaluation metrics are also crucial in assessing the effectiveness of spam detection models. In addition to traditional metrics like accuracy, precision, recall, and F1 score, **cluster purity** or **silhouette score** can be used to measure how well the messages in each cluster correspond to the actual spam or non-spam classification. This helps determine how tightly clustered spam and non-spam messages are, offering insights into the quality of the clustering model.

Finally, the system's **scalability** and **adaptability** to new forms of spam are key considerations. Spam techniques are constantly evolving, and as such, the clustering model must be regularly retrained with fresh data to identify emerging trends and adapt to new spam tactics. Using **incremental learning** or periodic model updates can help maintain the accuracy of the system in a dynamic environment. Also, the integration of additional features like **user behavior analysis** (e.g., whether a user frequently flags messages as spam) can further improve detection performance and tailor spam filtering to individual preferences.

Here’s the additional information in points:

1. **Semantic Analysis Integration**:

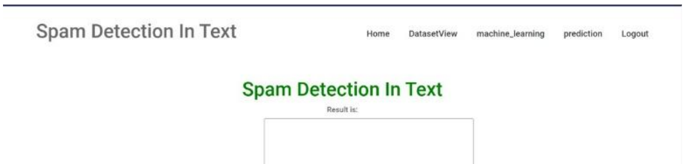
   Incorporating **word embeddings** (e.g., Word2Vec, GloVe) alongside traditional features like TF-IDF improves the model’s ability to understand the meaning and context of words, enhancing detection of spam messages with varied phrasing.

2. **Handling Imbalanced Datasets**:
   Spam datasets are often imbalanced, so techniques like **oversampling**, **undersampling**, and **anomaly detection** help balance the data and improve model performance.
   Active learning can be employed, where the model requests human feedback on uncertain classifications to iteratively improve its predictions.
3. **Evaluation Metrics**:
   Besides accuracy, **precision**, **recall**, and **F1 score**, additional metrics such as **cluster purity** and **silhouette score** measure the quality of clustering and how well messages align with their true classification (spam or non-spam).
4. **Scalability and Adaptability**:
   The system must be scalable and adaptable to handle new forms of spam, which evolve over time. Regular **retraining** with fresh data and **incremental learning** helps keep the model updated.
   User behavior analysis (e.g., identifying messages flagged by users as spam) can help tailor spam detection to individual needs and preferences.
5. **Refinement through Feedback**:
   Continuously refining the model through labeled data and user feedback ensures better accuracy in detecting spam messages and adapting to emerging trends.
6. **Feature Engineering**:
   Additional Text Features: Beyond word frequency, you can use advanced features like **n-grams** (combinations of adjacent words) and **character-level features** (e.g., detecting patterns in URLs or phone numbers). These help capture more detailed patterns, such as spammy phrases that might not be captured by word-based features.
   Sentiment Analysis: Sentiment analysis can be used to detect overly promotional or negative messages, which are often found in spam. A sudden shift in tone can be a strong indicator of spam content.
7. **Hybrid Models**:
   Combining **clustering** with other machine learning models, like **supervised classifiers** (SVM, Random Forest, etc.), can enhance detection. For example, after clustering, you can apply supervised learning to label the clusters as spam or non-spam, combining the strengths of both techniques.
   Another hybrid approach is **ensemble learning**, where multiple clustering algorithms (like K-means and DBSCAN) are used together to achieve more robust clustering results.

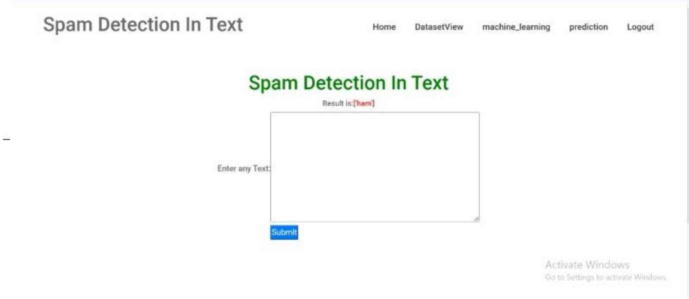
8. **Real-time Detection**:

   For practical applications, such as email or messaging apps, the spam detection system must operate in real time. **Streaming data** can be processed using online clustering algorithms (e.g., **MiniBatch K-means**) that update the model incrementally without needing to retrain from scratch.

   Real-time detection also involves handling **false positives** (legitimate messages flagged as spam) and **false negatives** (spam messages not flagged), which can be mitigated by continuous model improvement based on user feedback.



Result is:



Result is:['ham']

9. **Adaptation to New Spam Techniques**:

   As spam evolves, traditional models may fail to detect new types of spam (e.g., phishing attempts). To combat this, the system can use **incremental clustering** or **active learning** to identify new patterns in evolving spam. The model can learn from new examples of spam messages and adjust its parameters accordingly.

   Another strategy is **transfer learning**, where a pre-trained model (from one domain) is adapted for another. For example, a model trained on spam detection in emails can be transferred to detect spam in SMS messages or social media posts.

10. **Contextual Spam Detection**:

    Spam can sometimes be context-specific. A message that is spam in one context (e.g., a promotional message in a work email) may not be considered spam in another (e.g., a promotional message in a personal email). **Context-aware models** that take into account the nature of the conversation or user preferences can refine detection.

    Techniques like **topic modeling** (e.g., LDA) can identify the subject matter of a message and filter out spam based on topic relevance. This ensures that messages flagged as spam are not only irrelevant but also violate contextual norms.

11. **Handling Multilingual Spam**:

    Multilingual spam detection requires the model to understand spam characteristics in multiple languages. Using **language models** that are multilingual or training separate models for different languages can help detect spam across different linguistic groups.

    Translation-based models or **language-specific feature extraction** (like stop words in different languages) can further improve detection accuracy in diverse contexts.

These additional techniques enhance the robustness of spam detection systems, ensuring they can handle evolving threats while maintaining high accuracy.

When each mail document is represented by a vector, the elements of the vector are assigned two-part values [14] In our experiments, the factor is a local weight that reflects the weight of term within document and the factor is a global weight that reflects the overall value of term as an indexing term for the entire document collection as follows: where is the number of documents in the collection, is the frequency of the -th term in the -th document, and is the number of documents containing the -th term throughout the entire document collection. To evaluate efficiency of our system, we experiment with spam detection using freely available test collection LingSpam.

The Ling-Spam collection consists of 2412 nonspam messages and 481 spam messages by hand categorization. By using stop-list and lemmatizer, this collection consists four collections: bare(untreated), lemm(using lemmatizer), stop(using stop-list) and lemm+stop(using lemmatizer + stop-list). In our experiments, the data set contains 2170 non-spam messages and 432 spam messages and the test set contains 242 non-spam messages and 49 spam messages. Table 1 shows the results of the experiments.

In this figure, our system provides the high-performance for both spam and non-spam messages. The spam precision is more than about 90% and the non-spam precision is more than 96% for all collections. Moreover, to make an objective evaluation of our method, precision of our method is compared with that of other methods. In this comparison, we use Support Vector Machine(SVM)[8] and bogofilter[3].

SVM is one of the most powerful machine learning method and bogofilter is a Bayesian spam filter. We show the result in the table 2. This results show that the precision using our method is better than the bogofilter and is approximately equivalent to the SVM. So it can be concluded that using the spam and non-spam clusters based on the unsupervised clustering is a effective method for detecting spam.

However, we define the threshold value of spam cluster as 70% so that the non-spam precision is not 100%. Thus we define the greater threshold value than 70% and calculate the precision of spam on the condition that the non-spam precision is nearly 100%. the table 3 and 4 show the results of these experiments using TF IDF (Text Frequency Inverse Document Frequency) and TF respectively as term weighting.

The spam precision is about 90% so that our method provides the high-performance for spam messages. Additionally, the spam precision using TF is better than that using TF IDF except result of lemm stop.

REFERENCES

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. In Proceedings of the Workshop on Machine Learning in the New Information Age: 11th European Conference on Machine Learning (ECML 2000), pages 9–17, 2000.

[2] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In Proceedings of the workshop Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000), pages 1–13, 2000.

[3] Bogofilter. http://bogofilter.sourceforge.net/.

[4] W. W. Cohen. Learning rules that classify e-mail. In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access, pages 203–214, 1996.

[5] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. Technical report, IBM Almaden Research Center, 1999.

[6] H. Druker. Support vector machines for spam categorization. In Proceedings of the IEEE Transaction on Neural Networks, volume 10, pages 1048–1054, 1999.

[7] P. Graham. Better Bayesian Filtering. http://www.paulgraham.com/better.html.

[8] T. Joachims. Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer, 2002.

[9] A. Kolcz and J. Alspector. Svm-based filtering of e-mail spam with content-specific misclassification costs. In Proceedings of the TextDM !G01 Workshop on Text Mining, IEEE International Conference on Data Mining, pages 1048– 1054, 2001.

[10] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

[11] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, pages 1048–1054, 1998.

[12] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing(EMNLP 2001), pages 44–50, 2001.

[13] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41:288–297, 1990.

[14] I. H. Witten, A. Moffat, and T. C. Bell. Managing Gigabytes: Compressing and Indexing Documents and Images. Van Nostrand Reinhold, New York, 1994.Mayer, J. D., Roberts, R. D., & Barsade, S. G. (2008). Human abilities:Emotional intelligence. Annual Review of Psycology 5

[15] Hochschild, A. R. (1983). The managed heart: Commercialization of human feeling. University of California Press

[16] Diefendorff, J. M., Richard, E. M., & Yang, J. (2008). Linking emotion regulation strategies to affective events and negative emotions at work. Journal of Vocational Behavior, 73(3), 498-508.

[17] K. Sailunaz and R. Alhajj"Emotion and sentiment analysisfrom Twitter text," Journal of Computational Science, vol. 36Article ID 101003,, 2019.

[18] J. X. Chen, D. M. Jiang, and Y. N. Zhang"A hierarchical bidirectional GRU model with attention for EEG-based emotion classification," IEEE Access , vol.7,pp.118540,2019.

[8] F. M. Alotaibi, "Classifying text-based emotions using logistic regression," VAWKUM Transactions on Computer Sciences,vol. 7, no. 1, pp. 31–37, 2019.

[9] M. Hasan, E. Rundensteiner, andE. Agu, "Automatic emotion detection in text streams by analyzing twitter data," International Journal of Data Science and Analytics,vol.7,no.1,pp.35-51,2019.

[10] D. Acharya, "Comparative analysis of feature emotion technique based on dataset," in Soft Computing for Problem Solving, A. Tiwari, K. Ahuja, A. Yadav, J. C. Bansal,K. Deep, and A. K. Nagar, Eds., vol.1392, Singapore,Springer,2021.2021.

[11] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," Social Network Analysis and Mining, voll. 11, ,no. 1,p. 81,

[12] C. R. Chopade, Text based emotion recognition: a survey," International Journal of Science and Research, vol.2, no. 6, pp.409-414,2015.

[13] A. A. Alnuaim, M. Zakariah, P. K. Shukla et al., "Humancomputer interaction for recognizing speech emotions using multilayer perceptron classifier," Journal of Healthcare Engineering,vol.2022,Article ID 6005446,12 pages,2022

[14] S. M. Mohammad and F. Bravo-Marquez, "WASSA 2017Shared Task on Emotion Intensity," 2017, http://arxiv.org/abs/1708.03700

[15] W. Ragheb, J. Az´e, S. Bringay, and M. Servajean"Attentionbased modeling for emotion detection and classification textual conversations,"2019, http://arxiv.org/abs/1906.07020.