

AUTO EMAIL RECOMMENDATION & PREDICTION USING MACHINE LEARNING

Ahmad Fadhil N
Faculty of Computer Science
President University
Bekasi, West Java
fadhil.naswir@president.ac.id

Abstract— *Email has become a widely used communication tool worldwide for disseminating information, and is also a means of disseminating information within businesses and organizations. With the increasing popularity of email, many parties are using it as a means of communication. Many users spend a lot of time reading, replying to, and organizing their emails. With this problem in mind, a method for automatically and efficiently replying to emails is essential, especially for large companies or organizations. Supervised and unsupervised learning is part of Machine Learning which can determine or predict a category for an email that will be replied to according to the available template/category. It is hoped that these two methods will improve performance when it comes to responding to emails and related matters. By using two different classification methods and processes, the experiments will get different results and accuracy to be compared. The algorithms used in the classification process are the algorithm supervised: Naïve Bayes and KNN. This experiment also utilizes TF-IDF for document representation. Using this method, classification results will be obtained that match the categories of the available email dataset.*

Keywords— *email reply prediction, machine learning, supervised learning, Naïve Bayes, KNN*

I. INTRODUCTION

Information is an essential part of human life that cannot be ignored because it is a "tool" through which humans can obtain sources, data, or information about something, where the information obtained is expected to be obtained more quickly. Information obtained by a person can be published in various ways, both directly and using tools to disseminate the information. One of the tools for disseminating information is email, commonly known as electronic mail. Email has now become a communication tool that is very often used by people around the world to disseminate information and also as a means of disseminating information in business and organizational matters. With the increasing popularity of email use, many parties are using email as a means of communication. This can indicate the occurrence of overload and there will be problems in determining the priority of these emails. Many users spend a lot of time reading, replying to, and organizing their emails.

In an organization where email is used to share information with clients or other parties, it is a mandatory rule to create harmony between the organization and other parties. The biggest problem in

organizations is the accumulation of incoming emails, commonly known as email overload and can lead to delays in email replies. This can hinder an organization's work processes due to late email replies.

Email reply prediction is a method to anticipate whether a received email requires a reply or not. Taiwo, et al. (2009) developed a system to manage email inboxes and also manage time efficiently. Whittaker and Sider (1997) analyzed the use of email for task management, personal archiving and asynchronous communication and referred to these three as "email overload". They concluded that:

- Users perform a wide variety of work-related tasks with email and
- As a result, users are overwhelmed by the amount of information in their mailboxes.

Also Dredze et al (2005) provided a solution to answer email prediction by assessing the date and time in email messages because emails containing dates and times are time-sensitive and may require a response. Other studies have focused on how people store their emails, what purposes they serve, and their importance as a tool for coordination in everyday life.

Previously there has been similar research regarding prediction of email replies. Determining whether a received email requires a reply. The system also provides a good and efficient way to prioritize emails and a new method for predicting emails for message replies. With research that is also based on machine learning, it is hoped that this can be the basis for this research to determine predictions and reply to emails according to the template.

In this writing, the author conducted research using an algorithm machine learning and made a comparison of methods, namely the Naive Bayes and KNN using Rapid Miner. By comparing the results of the best algorithms, we can determine which method is suitable for email classification.

II. LITERATURE REVIEW

A. Automatic Text Classification

Automatic Text Classification is a branch of Artificial Intelligence (AI) that is carried out to categorize texts or documents into predetermined categories. The main goal is to classify new text data based on previously prepared texts or documents that already have categories. A comparison is then made with the text to be classified, and from this process, a prediction or conclusion about the new data is produced.

In general, the workflow of text classification according to Ikonomakis, Kotsiantis, and Tampakas (2005) is as follows:

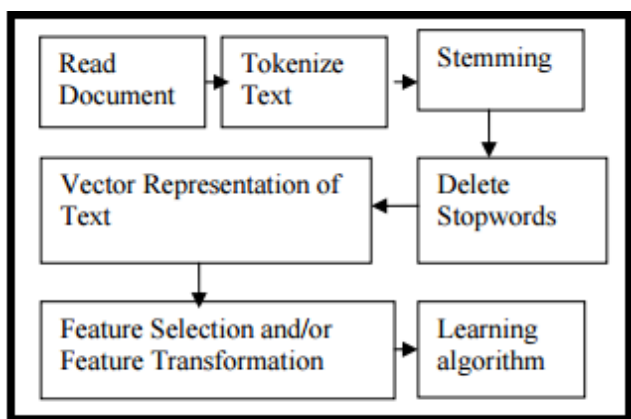


Figure 1. General Classification Process

B. Auto Reply Email and Prediction

Edstrom and Ewald (2019) contributed a sociolinguistic analysis of politeness strategies in auto-reply emails, drawing on Brown and Levinson's framework of positive and negative politeness. Their study demonstrated that effective auto-replies often incorporate greetings, expressions of gratitude, and apologies for unavailability, thereby mitigating potential face-threatening acts. This perspective underscored the importance of tone, politeness, and user perception in determining the success of automated communication systems. The main strength of this research is its novel focus on the pragmatic quality of replies, which is crucial for trust and acceptance in professional communication. However, the study's limitation lies in its lack of integration with computational approaches, meaning its insights remain largely conceptual guidelines rather than implementable design features in automated reply generation models.

Grandhi and Lanagan-Leitzel (2016) approached the email reply problem from a human factors perspective, investigating the decision to use "Reply" versus "Reply All" in group communication. Through a combination of controlled experiments and surveys, they found that errors in reply behavior were more strongly influenced by the clarity of the sender's instructions and the visibility of recipient lists rather than by interface design alone. This study's strength lies in highlighting the social and contextual dimensions of email communication, an aspect often overlooked in purely algorithmic research. Their findings imply that automated reply systems must account not only for textual content but also for recipient context and user intent. The limitation, however, is that the research did not integrate computational modeling, leaving open questions about how such behavioral insights can be operationalized within machine learning pipelines.

In a follow-up engineering contribution, Ayodele, Zhou, and Khusainov (2009) presented a practical implementation of machine learning techniques for email reply prediction, published in the Lecture Notes in Engineering and Computer Science series. Unlike their earlier theoretical framing, this paper focused on the application and system integration aspects, illustrating how reply prediction could be embedded into organizational email workflows. The value of this work lies in its attempt to bridge the gap between research prototypes and operational systems. However, the paper primarily emphasized engineering feasibility rather than algorithmic innovation, and the lack of rigorous evaluation against state-of-the-art baselines reduced its academic impact.

Building on their earlier work, Ayodele, Zhou, and Khusainov (2009) expanded the investigation of email reply prediction using a machine learning approach that compared multiple techniques in both supervised and unsupervised contexts. Their study reiterated the importance of linguistic and structural features, but also emphasized workflow prioritization, where incoming messages could be automatically ranked according to the likelihood of requiring a reply. The research highlighted the promise of machine learning to streamline communication management, particularly in high-volume environments. Its primary strength is the exploration of varied learning paradigms and feature types, providing a more systematic view of the problem. Nevertheless, the study was limited by the absence of standardized benchmark datasets and detailed evaluation metrics, which makes it difficult to compare directly with later systems.

Dredze et al. (2008), in their work presented at the International Conference on Intelligent User Interfaces (IUI), introduced the concept of “Intelligent Email” through two pragmatic tasks: reply prediction, which alerts users when a message requires a response, and attachment prediction, which identifies missing attachments or suggests relevant documents. Their approach utilized a shared text-classification backbone with task-specific features, and importantly, they evaluated performance in both single-user and cross-user settings, marking one of the early acknowledgments of the challenges posed by personalization and domain shift. Although the paper provides limited details on the underlying model and feature set due to its concise format, its main contribution lies in the task formulation and dual problem framing, which became foundational for subsequent research in email automation. The study’s strength is its grounding in real, product-inspired scenarios and its forward-looking consideration of cross-user generalization, while its limitations include the brevity of the work and the lack of transparency in reporting specific features and evaluation metrics.

Ayodele and Zhou (2008) proposed one of the earliest approaches to email reply prediction through an unsupervised learning framework. Their study emphasized the challenge of determining whether an email requires a response, leveraging linguistic cues such as interrogatives and structural metadata like headers to distinguish reply-needed messages from those that did not. By framing the problem in an unsupervised manner, their work addressed the difficulty of acquiring labeled training data, an issue especially relevant in corporate communication environments. The strength of this contribution lies in its attempt to reduce dependence on manual annotation and its recognition of contextual linguistic features as signals for reply necessity. However, the approach was constrained by the limitations of pre-deep learning methods and lacked large-scale experimental validation, making its generalizability to diverse datasets uncertain.

III. METHODOLOGY

In this section, several methodologies employed in this study are presented. The discussion includes a detailed explanation of the research design, the process of collecting the email dataset from the Enron Email Dataset Website, and the proposed method utilized in this research. Furthermore, the author elaborates on the testing procedures applied to preprocess the data prior to algorithm implementation, followed by the application of the algorithm to the prepared dataset. By adopting and applying these methodologies, this study aims to be systematically conducted and to yield valid and reliable conclusions.

A. Research Design

At this stage, the methodology that will be applied in this research will be explained to answer the existing problems. In this research design, the research methods used in carrying out the testing process are as follows:

1. Research object
The object of this research is a collection of emails from Enron Corporation which will be used for automatic classification using *machine learning* by using several different algorithms, which will then undergo a training process to be able to determine conclusions from the data.
2. Data collection
In this section, we will explain where the source of the dataset that will be used to carry out the research process is obtained from the Enron *database* which has the format *excel file (xls)* including the preprocessing.
3. Proposed method
At this stage, we will explain the methods used in the classification process using *machine learning*.
4. Method Implementation
In this process, the stages in implementing the algorithm and testing it with several experiments are explained so that it will produce the best accuracy from the various methods and algorithms used.

B. Data Collection

The dataset collection for this research was taken from Enron Email Dataset which can be accessed via the website Enron From the dataset obtained, it is necessary to read and understand the contents of the collection of emails because random the subjects or categories within the dataset and the relationships between the email

datasets. There are also differences in content and language style, and there are also several abbreviations of words in the content of the email.

Here is an example of Enron Email Dataset which has been modified by adding columns (Status and Category)

Date	From	To	Subject	Content	Status	Category
5/8/2001 19:01	frozenstet({'tim. heize nrade r@enron.com'})	frozenstet({'greg.walley@enron.com'})	North west Power Planni ng Council Studies	Here's a collecti on of recent North west Power Planni ng Council studies showin g the region' s growin g difficul ty in meetin g interna l load, and decrea sing likelih ood of having power availab le for export:	TRUE	WORK

Table 1. Enron Email Dataset

Table 1 presents an example from the Enron dataset. In this table, only a limited portion of the data is shown due to the length of the email content. The dataset consists of several columns, namely: No, Date, From, To, Subject, Content, Status, and Category. *No* represents the sequential number of the email, *Date* indicates the date and time the email was sent or received, *From* is the sender's address, *To* specify the recipient(s), *Subject* refers to the topic of the email (optional), *Content* contains the body of the email, *Status* denotes whether the email requires a reply (TRUE = Reply, FALSE = No Reply), and *Category* defines the type of email (Work, Information, Personal, or Other).

The *Status* and *Category* columns were not part of the original Enron Email Dataset; they were added and modified by the author to enable this research. The categorization was determined using the following conditions:

1. **Work**
 - If the email has only one recipient ($To = 1$)
 - If the sender (*From*) is an Enron staff member
 - If the content is work-related
2. **Information**
 - If the email is sent to more than two recipients ($To > 2$)
 - If the sender (*From*) is an Enron broadcast (e.g., All, Worldwide, or Corp.)
3. **Personal**
 - If the email has only one recipient ($To = 1$)
 - If the sender (*From*) is an individual (name)
 - If the content is personal in nature
4. **Other**
 - If the email has no recipient (*To* is empty)
 - If the sender (*From*) is not part of Enron Corp.

By modifying the dataset based on these conditions, the email data could then be used to compute accuracy and precision, as well as to perform classification using the Naïve Bayes algorithm and K-NN.

C. Pre-Processing

Preprocessing is a stage before implementing an algorithm that aims to process sentences or documents so that the document is more meaningful and structured and can be continued to the next process.

This process has several processes, including:

- a. *Tokenizing*
Tokenizing is a process of separating sentences into phases or words. The content of the dataset will be separated into individual words, making it easier to process further.
- b. *Filtering Stopword*
Filtering is a stage to remove words that are considered to have no correlation to the specified query.
- c. *Stemming*
Stemming is the process of changing a compound word into a base word that is included in the stages *text processing*. Language rules are applied to remove these

affixes. The ones used in this research are *Stem (Porter)*. This is an algorithm that removes common morphology and endings from English words. This is because the dataset used is an English dataset. The use of this algorithm *Stem (Porter)* is also a part of the normalization process which is commonly used as a form of *Preprocessing/Text Processing* in preparation for carrying out the process *Information Retrieval*.

d. *Transform Cases*

Transform Cases is the process of converting the entire content of a dataset. The word conversion process involves converting all letters to lowercase or uppercase. In this case, all content in the dataset is converted to lowercase.

D. *Proposed Methodology*

The research method that will be used in this research is a method using supervised learning in classifying email categories using machine learning. The algorithm of each learning is the algorithm Naïve Bayes and K-NN will be carried out as the main algorithm to be compared.

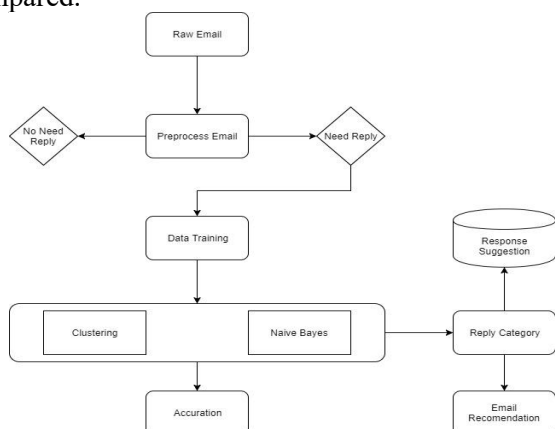


Figure 2. Research Methodology Flow

E. *Method Implementation*

In this study, the author used 2 different methods in testing, namely Naïve Bayes and K-NN. This method is used to determine the accuracy of the available dataset. The method was applied to the Enron email dataset, a dataset from a company in the business sector called Enron. The available dataset has been categorized previously, but this test will also generate predictions or decision results for undefined email categories using several algorithms. machine learning. The dataset used in this study consisted of 1,000 email lists with four

categories. The algorithms used to classify this email dataset include:

- **Naïve Bayes**

Naive Bayes Algorithm is a classification method proposed by a British scientist named Thomas Bayes. This algorithm is based on probability and statistical theory and methods, and its main characteristic is a very strong (naïve) assumption regarding the independence of each condition/event. Based on previous experience as a reference, Naive Bayes algorithm can predict future opportunities, this theory is better known as Bayes' Theorem. The advantage of using this algorithm is that this method only requires a small amount of training data (data training) which is a little to determine the parameter estimates needed in the classification process.

- **K-Nearest Neighbor**

The K-Nearest Neighbors (K-NN) algorithm is a method used for classifying objects based on their proximity to the training data. The training data are projected into a multidimensional space, where the features of the data are represented along each dimension. Regions within this space are defined according to the classification of the training data. A point in this space is assigned to class a if class a is the most frequently represented among its k nearest neighbors. The Euclidean distance is used to determine the closeness of neighbors from each class.

Based on the explanation of the algorithms above, testing will be carried out in accordance with the theoretical foundation of the described algorithms. These algorithms will be processed using the *RapidMiner* tool. Several key stages and components in *RapidMiner* are as follows:

Repositories

Repositories serve as storage for the data and processes used in *RapidMiner*. By default, *RapidMiner* provides *Local Repositories* as the storage location for datasets and processes that have been created.

Process

A process is the workspace where datasets and selected algorithms are utilized. Within this process, additional operations such as preprocessing can also be performed.

Operators

Operators function as connectors between datasets and processes. They are also part of the core program used to perform computations with the chosen algorithms.

Preprocessing, Algorithm, Apply Model, and Performance

In RapidMiner, preprocessing, model application, and performance evaluation are conducted using operators. For example, preprocessing can be performed with the *Process Documents from Data* operator. Each algorithm is also represented as an operator, named according to the corresponding algorithm. The *Apply Model* operator is used to compute predictions on preprocessed data, while the *Performance* operator is employed to record evaluation metrics from the process.

Result

Results represent the output generated from the computations within a process. They include metrics such as accuracy, precision, and recall, as well as the outcomes of predictions presented in tabular form.

IV. RESULT AND DISCUSSION

In preparing the dataset for testing, several steps were necessary to ensure reliable computation and classification. This was due to the presence of missing values in the email dataset, along with typographical errors and the use of non-standard language within the email content. Such issues posed challenges in achieving optimal accuracy during testing.

At the initial stage, the dataset was modified by adding two additional columns, namely Status and Category. Subsequent testing was conducted using RapidMiner as the primary tool for computation, classification, and accuracy evaluation. RapidMiner was selected for this study because of its simplicity, practicality, efficiency, and accuracy, as well as its comprehensive support for preprocessing tasks prior to testing.

Preprocessing steps such as stemming, tokenization, and word filtering were applied before computation. These processes enhanced the quality of the dataset and contributed to obtaining more accurate results.

A. Experiment Result

The dataset used in this study consists of a total of 1,000 emails, divided into 700 emails for training data and 300 emails for testing data.

Based on the calculation results, it can be observed that applying cross-validation leads to slight variations in accuracy, either improvements or reductions, depending on the algorithm used. For example, in the case of the Naïve Bayes algorithm, the accuracy without cross-validation was 80.67%, whereas with cross-validation it decreased to 74.60%. This shows a difference of more than 5% in accuracy when cross-validation is applied.

A comparison of the testing results for each algorithm can be seen in the following tables:

Algorithms	Akurasi
Naïve Bayes	80.67%
Naïve Bayes (Status)	79.33%
Naïve Bayes (Category)	74.00%
Naïve Bayes Cross Validation	74.60%
Naïve Bayes Cross Validation (Status)	77.60%
Naïve Bayes Cross Validation (Category)	68.20%
K-NN	46.67%
K-NN (Status)	52%
K-NN (Category)	46.67%
K-NN Cross Validation	49.80%
K-NN Cross Validation (Status)	77.80%
K-NN Cross Validation (Category)	49.80%

Table 2. Classification Result

B. Result Analysis

From the accuracy results obtained, it can be concluded that the Naïve Bayes algorithm outperformed the K-NN algorithm in this study. The author argues that this is due to the probabilistic approach employed by Naïve Bayes in its computations, whereas K-NN relies on the ratio of classes among the nearest neighbors without using probability. Since the computation in Naïve Bayes is relatively fast and the dataset used in this study is fairly complete with linearly consistent labels, the results obtained with Naïve Bayes are superior to those of K-NN. Conversely, it is possible that K-NN could yield better results than Naïve Bayes when applied to a different dataset or environment.

Testing with cross-validation also produced significant differences in accuracy. However, it cannot be concluded that cross-validation necessarily improves accuracy, as in some cases accuracy increased, while in others it decreased.

A notable difference in accuracy was also observed between the Status column and the combined Status + Category columns. The results indicate that classification based solely on Status yielded higher accuracy compared to the combined attributes, both in Naïve Bayes and K-NN calculations.

V. CONCLUSION & FUTURE WORK

In this study, comparing different methods was difficult due to the systematic differences in each algorithm. Machine Learning as the main learning is very suitable for research needs using this dataset because the data processing and calculations are in accordance with the basic concepts of machine learning and data mining.

In this research stage preprocessing is very helpful in the process of calculating and increasing the accuracy value of each algorithm. Testing was carried out on a company's email dataset which had various subjects and content, from the algorithms that were compared, the method of Naïve Bayes has a higher accuracy value compared to the method K-NN. This is due to differences in the process and calculation methods.

The TF-IDF method is the right method for processing each word, because it is useful in determining the probability and similarity in each email list to the label.

Testing using the same data in each category is appropriate because all conditions in the dataset are the same.

The dataset used in this study was more structured and complete, which facilitated data processing and produced better accuracy. Future testing can be conducted using other classification algorithms.

VI. ACKNOWLEDGEMENT

This research is supported by President University.

VII. REFERENCES

1. S. Whittaker, C. Sider, 1996, 'Email overload: exploring personal information management of email', CHI '96', pp.276-283. ACM Press.
2. Edstrom, Anne & Ewald, Jennifer. (2019). Characteristics of effective auto-reply emails: Politeness and perceptions. *Technology in Society*. 58. 10.1016/j.techsoc.2019.01.005
3. Shay, Moran & Davidson, Roei & Grinberg, Nir. (2024). EnronSR: A Benchmark for Evaluating AI-Generated Email Replies. *Proceedings of the International AAI Conference on Web and Social Media*. 18. 2063-2075. 10.1609/icwsm.v18i1.31448.
4. Grandhi, Sukeshini & Lanagan-Leitzel, Lyndsey. (2016). To Reply or To Reply All: Understanding Replying Behavior in Group Email Communication. 559-568. 10.1145/2818048.2819981.
5. Dredze, Mark & Brooks, Tova & Carroll, Josh & Magarick, Joshua & Blitzer, John & Pereira, Fernando. (2008). Intelligent email: Reply and attachment prediction. *International Conference on Intelligent User Interfaces, Proceedings IUI*. 321-324. 10.1145/1378773.1378820.
6. V, Rajaraman & M, Prakash. (2020). Deepreply - An Automatic Email Reply System With Unsupervised Cloze Translation And Deep Learning. *Ictact Journal on Soft Computing*. 10. 2090-2095. 10.21917/ijsc.2020.0297.
7. Dredze, Mark & Brooks, Tova & Carroll, Josh & Magarick, Joshua & Blitzer, John & Pereira, Fernando. (2008). Intelligent email: Reply and attachment prediction. *International Conference on Intelligent User Interfaces, Proceedings IUI*. 321-324. 10.1145/1378773.1378820.
8. Ayodele, Taiwo & Zhou, Shikun & Khusainov, Rinat. (2009). Email Reply Prediction: A Machine Learning Approach. 114-123. 10.1007/978-3-642-02559-4_13.
9. Ayodele, Taiwo & Zhou, Shikun. (2008). Email reply prediction: Unsupervised leaning approach. 844-849. 10.1109/ICDIM.2008.4746844.
10. Ayodele, Taiwo & Zhou, Shikun & Khusainov, Rinat. (2009). Applying Machine Learning Techniques for Email Reply Prediction. *Lecture Notes in Engineering and Computer Science*. 2176.
11. Tyler, J., & Tang, J. When can I expect an email response? A study of rhythms in email usage. In *Proc. of ECSCW 2003*, Oulu Univ. Press (2003), 238-258
12. M. Dredze, J. Blitzer, F. Pereira, J 2005, 'Reply Expectation Prediction for Email Management', CEAS'2005, 2nd Conference on Email and Anti-Spam, Stanford University, CA
13. Ikonomakis, Emmanouil & Kotsiantis, Sotiris & Tampakas, V.. (2005). Text Classification Using

Machine Learning Techniques. WSEAS
transactions on computers. 4. 966-974.