Letter Detection: An Empirical Comparative Study of Different ML Classifier and Feature Extraction

Aji Prasetya Wibawa [a,1,*], Nastiti Susetyo Fanany Putri [a,2], Prasetya Widiharso [a,3]

[a] Electrical Engineering, Universitas Negeri Malang, Malang, Indonesia
[1] aji.prasetya.ft@um.ac.id; [2] nastiti.susetyo.2005348@students.um.ac.id; [3] prasetya.widiharso.2005348@students.um.ac.id
* corresponding author

ARTICLE INFO

Keywords
Official and private latter classification
Machine learning classifier
Accuracy Measure

ABSTRACT

Work and communication activities are inextricably linked. Letters are an example of a communication medium that is still widely utilized. When it comes to significant job, however, simply an official letter is required. Official and private letters must be distinguished and classified. Different feature extraction methods, such as the count-vectorizer and TF-IDF vectorizer, are employed to transmit the detection of this official and personal letter. To categorize letters by type, various machine learning (ML) techniques are employed. Nave Bayes, Support vector machine, and AdaBoost are the algorithms. The accuracy measurements used in this study include accuracy scores, F1-mean, recall, and precision. The best working algorithm is Naïve Bayes for two vectorizer methods used, with an accuracy value of 98%.

1. Introduction

Today the wave of digitalization is very strong, it also affects the activities of offices, schools, and public service agencies. In carrying out these activities, it cannot be separated from correspondence activities. In today's modern era, letters are also not always in printed form but have penetrated into electronic form (email) using the internet. In simple terms letters are grouped into two types, official letters and unofficial letters.

An official letter is a letter used for the official benefit of individuals, agencies, and organizations [1]. Examples of official letters include invitations, notification letters and circulars. The characteristics of official letters include using letterhead, there are letter numbers, attachments, subjects, using the standard language, and including stemples from institutions [2]. In addition to official letters there are also unofficial letters (personal letters), that is, letters used for personal interests [3]. This personal letter connects friends or family [4]. The characteristics of this personal letter include not using letterhead, letter numbers, not using standard language, and no attributes related to agencies or organizations [5].

The main point in running administration in both large and small offices is to determine the type of letters received. Employees often struggle and make mistakes in carrying out the work. Therefore, this study tried to group the types of official and private letters with the help of machine learning, by adopting previous research on the topic of email spam-ham classification.

In this paper, the dataset consists of pieces of official and personal letters. The sentence snippets in this letter are grouped using several algorithms, namely naïve bayes, support vector machine (SVM) with linear kernel and AdaBoost with two feature extraction words count vectorizer and TF-IDF.

10.31763/simple.v5i1.45

Count vectorizer was chosen because the two methods are often used in similar studies. Furthermore, this paper will discuss the methods used in section 2, the discussion in section 3.

2. The Proposed Method

2.1. Countvectorizer

The countervectorizer method applies a bag of words that does not use text structure and only processes information from the number of words [6]. This method works by converting the string representation into a numeric vector [7]. The input of vector comes from the number of unique words in a document and will be assigned an index for each word.

The countvectorizer will construct a sparse matrix A measuring m times n of text document B, where m is the total number of documents and n is the total number of different words used in B. All inputs aij= total number of words jth appear in the document ith.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

If in a document the desired unique word is not found then, the input of this word line is zero. This zero value can be filled by converting it into the todense() method which is a representation of the sparse matrix to make the formed matrix better [8].

2.2. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF vectorizer is an old feature extraction method that is popular because it is considered effective and applied in the pre-training stage [9]. This method combines the concepts of term frequency (TF) and document frequency (DF). TF represents how often a term or word appears [10] and how important it is in a single document [11]. The TF matrix is composed of the number of documents in a row and the number of terms that differ from the sum of all documents [12].

DF represents the number of documents that contain a unique term, indicating how common the term is [13]. Inverse document frequency (IDF) is the weight of a term, used to reduce the weight of a term when the term appears spread across all documents [14]. The calculation of the IDF is listed in formula (1).

$$idf_i = \log\left(\frac{n}{df_i}\right) \tag{1}$$

idf_irepresents the IDF value for the term I, df_i is the number of documents containing the term I, and n the total number of documents. The higher the DF value of a term, the lower the IDF value [15]. If the term appears in all documents the DF value will be equal to n and the IDF value equal to zero. This is because the value of log(1) is zero [16]. The TF-IDF score is generated from the multiplication of the TF matrix by the IDF as listed on (2).

$$w_{i,j} = tf_{i,j} \ x \ idf_i \tag{2}$$

Where $w_{i,j}$ is TF-IDF is the term I in the document j, $tf_{i,j}$ is the frequency of terms for I terms [there is a document] j. idf_i is the IDF score for all terms [17].

2.3. Dataset Description

The dataset is compiled from snippets of official letters and personal letters. These snippets come from various sources and parts of the letter. It consists of 100 data. The number of these datasets is divided into two equally large categories between private and official letters. The distribution of the dataset is the same, it can be said that if the dataset is balanced so that there is no need for oversampling to balance the dataset.

2.4. Research Overview

In this trial, it aims to find out which algorithms and feature extractors work optimally to detect official and personal letters. The trial steps are illustrated in Fig. 1.

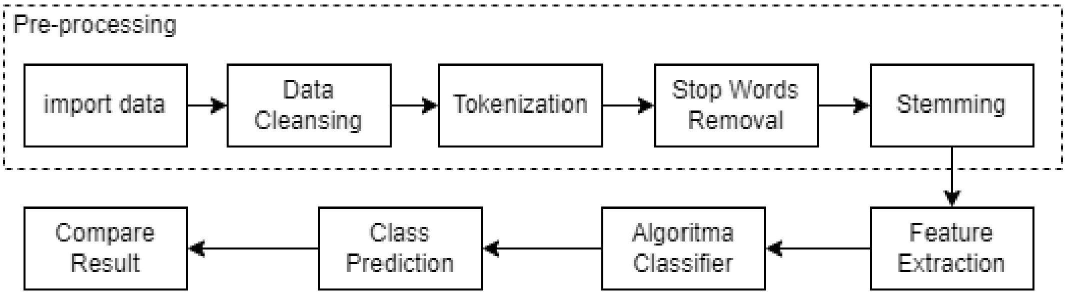

Fig. 1. Official and private mail detection architecture

2.5. Pre-processing

Text data in documents has an unstructured tendency that makes it more difficult to know the information contained [18]. Therefore, pre-processing is needed to change the format of the data set according to what is needed by the algorithm used. This pre-processing consists of several steps including tokenization, stop words removal and steaming

a. Import Data
The first step in mail detection is to import a dataset consisting of official and personal letters. After performing this stage, the data will be stored in the variables surattrain and surattest in the library.

b. Data cleaning
This stage of data cleaning is used to remove punctuation marks from documents, since punctuation is not needed in the classification process. In this stage, the conversion of letters into lower-case is carried out, to avoid the occurrence of different treatments in the classification process if the typeface size used is different.

c. Tokenization
Tokenization divides sentences into vocabulary meaning so-called token[19]. In HTML tokenization, XML scripts, special characters, and punctuation marks in a document do not affect in the performance of the algorithm used [20]. Examples of tokenization are as follows, showing vocabulary generation, feature selection from training and tokenization. Example sentence = [“ hello, apa kabar”, “ Dengan hormat”]. From that sentence it can be tokenized to “hello”, “apa”, “kabar”, “Dengan”, “ hormat”. Punctuation and spaces are often used in reference tokenization separators between tokens [21].

d. Stopword Removal
The purpose of this stage is to eliminate tokens or terms that usually refer to 'functional words' because they do not have an important meaning [22]. The word is like this, that, that and but. The term or word is often repeated in a document, although the term has no effect on the classification results but can affect the computational complexity [22].

e. Stemming
Steaming is done to remove suffixes so that the core word of the sentence is found. Different tokens consisting of the same one native word can be identified as the same token. The terms 'present' and 'presented' come from the same main word which is 'present'. In English classification, the porter rule is often applied in stemmers because it has good effectiveness and accuracy.

2.6. Classification Algorithm

a. Naïve Bayes
A simple method of supervised learning to perform classification. In the classification of letters belonging to multidimensional datasets naïve bayes are very commonly used. This algorithm has often been used in similar problems such as email spam detection and filtering, opinion analysis or sentiment []. The positive side of this method is its ease and effectiveness that has been tested well. Having a computational time tends to be short and easy to build. Naïve bayes multinominal is also responsible for the number of words that appear in a given letter.

b. SVM
Processing labeled data in classification. SVM is suitable for text classification [23]. Discriminatory classification method which requires both positive and negative training sets. Works well for high data dimensions. Deploying a hyperplane to distinguish positive and negative data values in a multidimensional data set [24]. Support vector works to detect sets that are closer to the surface, if the set is not part of the support vector then the email will be removed from the data so as not to affect the performance of the SVM classification.

c. Boosting Classifier (AdaBoost)
Work on the basis of reassessment of weak classifying weights [25]. The error value will be calculated again and reweighted, which will strengthen the accuracy of the classification.

3. Results and Discussion

This official and private letter detection trial uses two different extraction features and three classification algorithms. In this paper, compare the accuracy values, to see which algorithm is the best and which feature extraction works optimally. To do so is used as a matrix. The results issued include accuracy, precision, and F1-meansure. Here is the discussion.

Accuracy is the value of the probability of algorithm being correct in doing its job of classifying data. Accuracy calculations are presented in (3).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

Where TP is the amount of positive data with Positive truth value, FN is the amount of negative data that the system assumes is false, FP is a lot of positive data that the program considers to be worth the wrong truth and TN is the amount of negative data that the system assumes has a true truth value.

Precision shows how accurate the desired data is and the predicted results produced by the model. This precision is produced with (4).

$$Precision = \frac{TP}{(TP + FP)} \tag{4}$$

Recall or sensitivity is the success rate of the system in finding information that has been generated in previous stages shown on (5).

$$Recall = \frac{TP}{(TP + FN)} \tag{5}$$

Table 1 shows the test results of the algorithms used to measure the accuracy of the model in the data set in (6). Used to evaluate binary classification systems, where classifications only group into 'positive' and 'negative'.

$$Recall = 2\frac{(Precision\ x\ Recall)}{(Precision + Recall)} \tag{6}$$

Table 1. Results of official and unofficial letter detection trials

Algorithms	Feature Extraction	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Naïve Bayes	Count Vectorizer	98	98	98	98
SVM		88	88	88	88
AdaBoost		86	86	86	86
Naïve Bayes	TF-IDF Vectorizer	98	98	98	98
SVM		96	96	96	96
AdaBoost		86	86	86	86

From the test results that have been carried out the naïve bayes algorithm has the best results with both 98% in count-vectorizer and TF-IDF vectorizer. The results of the processing are visualized in Fig. 2.

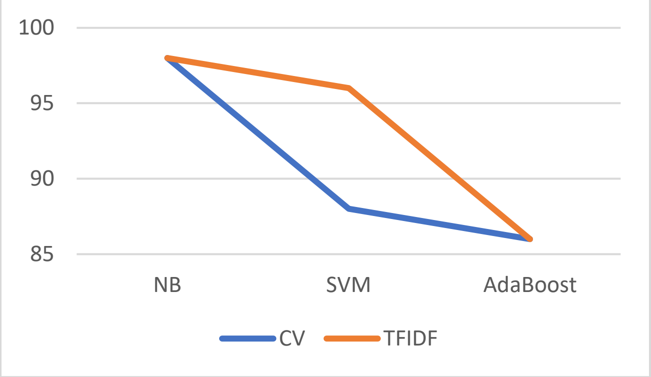

Fig. 2. Visualization of trial results

There is no difference between count-vectorizer and TF-IDF for the Naïve Bayes and AdaBoost algorithms. But the striking difference is in the SVM algorithm. Where the SVM count-vectorizer method works less optimally but with TF-IDF it can work more optimally with 96% accuracy. This research belongs to simple supervised learning research with special interpretation requirements, and SVM belongs to algorithms that are difficult to interpret [26]. Algorithms that can work well in cases like this include naïve bayes, decision trees and linear as well as logistic regression.

AdaBoost comes out with the smallest accuracy value for each vectorizer used. This can be due to the presence of out layers in the dataset used, thus disrupting the performance of this algorithm. On the other hand, it can also be caused because AdaBoost uses samples that were misclassified by the previous classifier as inputs in subsequent groupings [27]. So that the probability of selecting the incorrectly grouped sample is higher than the correctly grouped sample in the previous iteration is decreasing and has an impact on the accuracy value.

TF-IDF works more optimally than count vectorizer because TF-IDF considers the weight of the document as a whole word [28]. This is very helpful in overcoming repetitive words. The TF-IDF vectorizer also gives weight to the way in which words appear in the document so that the resulting matrix is better [29]. As for count- vectorizer only counts how often a word appears in a document which often results in bias for other words [30]. This algorithm tends to ignore unique words that can help increase effectiveness in data processing.

4. Conclusion

In this paper, several classification algorithms are applied using different extraction features. The experiment is applied with both official and unofficial letter datasets. As a result, naïve bayes work optimally with both count-vectorizer and TF-IDF vectorizer with 98% accuracy, this can be caused because naïve bayes can still work optimally with a small number of datasets. On the other hand, this algorithm can be adjusted to the wishes of researchers in grouping official and unofficial letters. The smallest accuracy value occurs with AdaBoost with 86% accuracy value.

The results of this study are expected to provide an overview for the next related research. The research can be in the form of the use of other classifier methods such as deep learning or hybrid method. Another research direction will apply other feature extraction methods.

References

[1] Nushashikin, S. Ramadhan, and Nurizzati, “Error Analysis in Indonesian Language at The Letter of the Education And Culture of Bukittinggi City,” *Proc. 4th Int. Conf. Lang. Lit. Educ. (ICLLE-4 2021)*, vol. 604, pp. 210–212, 2021.

[2] A. Nurachmana, “Penerapan Model Brainstorming pada Materi Menulis Surat Resmi dalam Mata Kuliah Menulis Mahasiswa PBSI FKIP UPR Semester Genap 2018/2019,” *J. Pendidik.*, vol. 21, no. 1, pp. 29–35, 2019.

[3] G. W. Saputra, “Mendampingi Siswa untuk Mengenal dan Memahami Surat Pribadi dan Surat Dinas pada Kelas VII MTs NU Umbul Sari,” *Jurnal Pengabdian Masyarakat (ABDIRA),* vol. 2, pp. 69–72, 2022.

[4] I. Kemal, “Penerapan Pendekatan Konstruktivisme dalam Meningkatkan Keterampilan Menulis Surat Pribadi pada Siswa Kelas IV SD Negeri 11 Tanah Jambo Aye Kabupaten Aceh Utara*,” J. Tunas Bangsa*, vol. 2, no. 2, pp. 41–66, 2015.

[5] I. Darussalam, C. Najimudin, and D. Firmansyah, “Pengaruh Metode Pembelajaran Grup Investigasi Dalam Menelaah Unsur-Unsur Dan Ciri Bahasa Serta Menulis Surat Pribadi,” *Parol. Jurnal Pendidik. Bhs. dan Sastra Indones.*, vol. 2, no. 1, pp. 67–72, 2019.

[6] S. Chowdhury and M. P. Schoen, “Research Paper Classification using Supervised Machine Learning Techniques,” *2020 Intermt. Eng. Technol. Comput. IETC 2020*, pp. 1-6, 2020.

[7] S. Vijayaraghavan *et al*., “Fake News Detection with Different Models,” *arXiv preprint arXiv:2003.04978*, 2020.

[8] A. Amirullah, I. Aulia, and D. Arisandy, “Implementing Cosine Similarity Algorithm to Increase the Flexibility of Hematology Text Report Generation*,*” *2020 Int. Conf. Data Sci. Artif. Intell. Bus. Anal. DATABIA 2020 - Proc.*, pp. 76–82, 2020.

[9] W.-C. Chang, F. X. Yu, Y.-W. Chang, Y. Yang, and S. Kumar, “Pre-training Tasks for Embedding-based Large-scale Retrieval,” *arXiv preprint arXiv:2002.03932*, pp. 1–12, 2020.

[10] S. W. Kim and J. M. Gil, “Research paper classification systems based on TF-IDF and LDA schemes,” *Human-centric Comput. Inf. Sci*., vol. 9, no. 1, 2019.

[11] N. S. Mohd Nafis and S. Awang, “An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification,” *IEEE Access*, vol. 9, pp. 52177–52192, 2021.

[12] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, “A novel approach for dimension reduction using word embedding: An enhanced text classification approach,” *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 1, p. 100061, 2022.

[13] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” *Concurr. Comput. Pract. Exp*., vol. 33, no. 23, pp. 1–12, 2021.

[14] Z. Tang, W. Li, and Y. Li, “An improved term weighting scheme for text classification,” *Concurr. Comput. Pract. Exp*., vol. 32, no. 9, pp. 1–19, 2020.

[15] T. Dogan and A. K. Uysal, “A novel term weighting scheme for text classification: TF-MONO,” *J. Informetr.*, vol. 14, no. 4, p. 101076, 2020.

[16] Z. Jiang, B. Gao, Y. He, Y. Han, P. Doyle, and Q. Zhu, “Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports,” *Math. Probl. Eng*., pp. 1-30, 2021.

[17] G. Sidorov, “Vector space model for texts and the tf-idf measure,” *SpringerBriefs Comput. Sci*., pp. 11–15, 2019.

[18] K. Adnan and R. Akbar, “Limitations of information extraction methods and techniques for heterogeneous unstructured big data,” *Int. J. Eng. Bus. Manag*., vol. 11, pp. 1–23, 2019.

[19] N. Mohapatra, N. Sarraf, and S. sarit Sahu, “Ensemble Model for Chunking,” *CS & IT Conference Proceedings,* pp. 113–119, 2021.

[20] C. Toraman, E. H. Yilmaz, F. Şahinuç, and O. Ozcelik, “Impact of Tokenization on Language Models: An Analysis for Turkish,” *arXiv preprint arXiv:2204.08832*, vol. 1, no. 1, 2022.

[21] A. N. Ulfah and M. K. Anam, “Analisis Sentimen Hate Speech Pada Portal Berita Online Menggunakan Support Vector Machine (SVM),” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 1, pp. 1–10, 2020.

[22] S. Sakthi Vel, “Pre-Processing techniques of Text Mining using Computational Linguistics and Python Libraries,” *Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021*, pp. 879–884, 2021.

[23] H. T. Sueno, “Multi-class Document Classification using Support Vector Machine (SVM) Based on Improved Naïve Bayes Vectorization Technique,” *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3937–3944, 2020.

[24] W. T. Meshach, S. Hemajothi, and E. A. M. Anita, “Real-time facial expression recognition for affect identification using multi-dimensional SVM,” *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 6, pp. 6355–6365, 2021.

[25] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, “Revisiting Deep Learning Models for Tabular Data,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 18932-18943, 2021.

[26] J. A. Jupin, T. Sutikno, M. A. Ismail, M. S. Mohamad, S. Kasim, and D. Stiawan, “Review of the machine learning methods in the classification of phishing attack,” *Bull. Electr. Eng. Informatics*, vol. 8, no. 4, pp. 1545–1555, 2019.

[27] A. Taherkhani, G. Cosma, and T. M. McGinnity, “AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning,” *Neurocomputing,* vol. 404, pp. 351–366, 2020.

[28] S. Krimberg, N. Vanetik, and M. Litvak, “Summarization of financial documents with TF-IDF weighting of multi-word terms,” Proc. 3rd Financ. Narrat. Process. Work. FNP 2021, pp. 75–80, 2021.

[29] M. N. Sahono *et al.*, “Extrovert and Introvert Classification based on Myers-Briggs Type Indicator(MBTI) using Support Vector Machine (SVM),” *Proc. - 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Secur. Age Digit. Disruption, iSemantic 2020*, pp. 572–577, 2020.

[30] A. Rajmohan, A. Ravi, K. O. Aakash, K. Adarsh, A. D. Raj, and T. Anjali, “CoV2eX: A COVID-19 Website with Region-wise Sentiment Classification using the Top Trending Social Media Keywords,” *2021 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2021*, pp. 113–117, 2021.