

RESEARCH ARTICLE



OPEN ACCESS

Received: 10.12.2021

Accepted: 30.12.2021

Published: 21.01.2022

Citation: Naïve AFE, Barbosa JB (2022) Efficient Accreditation Document Classification Using Naïve Bayes Classifier. Indian Journal of Science and Technology 15(1): 9-18. <https://doi.org/10.17485/IJST/v15i1.1761>

* **Corresponding author.**

jocelyn.barbosa@ustp.edu.ph

Funding: None

Competing Interests: None

Copyright: © 2022 Naïve & Barbosa. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](#))

ISSN

Print: 0974-6846

Electronic: 0974-5645

Efficient Accreditation Document Classification Using Naïve Bayes Classifier

Anna Fay E Naïve¹, Jocelyn B Barbosa^{2*}

¹ Instructor, Department of Information Technology, University of Science and Technology of Southern Philippines, C.M Recto Ave., Lapasan, Cagayan de Oro City, 9000, Philippines

² Associate Professor, Department of Information Technology, University of Science and Technology of Southern Philippines, C.M Recto Ave., Lapasan, Cagayan de Oro City, 9000, Philippines

Abstract

Objectives: To develop a desktop application that automatically classifies a document as to which area of accreditation documents it should belong to. Specifically, it aims to: a) To create a predictive model that addresses document classification tasks. b) To design and develop an application that classifies documents according to document classification. c) To evaluate the performance measures of the automatic document classification. **Methods:** We introduce an innovative approach for the automatic classification of accreditation documents. Specifically, an approach of including scanned or captured documents in classification task using Optical Character Recognition (OCR); use TF-IDF (Term-frequency Inverse Document Frequency) with stopwords removal, n-gram of 1-2 in preprocessing of the text documents; and Naive Bayes algorithm with additive (Laplace/Lidstone) smoothing as a classifier in building our model. **Results:** Performance measures such as accuracy, precision, recall, and f-score were conducted to evaluate the efficiency of the study. The results showed 82% accuracy, 84% precision, 82% recall, and 82% F-1 score. As we explore the use of OCR for text extraction, TF-IDF for text preprocessing, and Naive Bayes classifier, the results indicate that the proposed approach is efficient. **Conclusions:** Classification of input documents in whatever forms, may it be captured image, scanned or simple text documents were obtained using OCR, TF-IDF, and Naive Bayes classifier. It provides an efficient way of automatic classification of accreditation documents and it gives an avenue to address limiting factors of the previous works, i.e classifying documents based on one's opinion and time-consuming classification.

Keywords: Accreditation Document Classification; Document Classification Objective Evaluation; TF-IDF; Term frequency-inverse document frequency; Multinomial Naive Bayes; OCR; Optical Character Recognition

1 Introduction

Accreditation is one way for HEIs (Higher Education Institutions) to keep themselves in check with the standards. Accreditation requires documents if the standards are being met of a particular program in the HEI. Accrediting Agency of Chartered Colleges and Universities in the Philippines (AACUP), Inc. is one of the accrediting bodies in the Philippines. There are 10(ten) key areas that are used in the assessment of programs particularly; Area I - Mission, Goals and Objectives, Area II - Faculty, Area III - Curriculum and Instruction, Area IV - Students, Area V - Research, Area VI - Extension and Community Involvement, Area VII - Library, Area VIII - Physical Facilities, Area IX - Laboratories, Area X - Administration.

In each HEI, there are accreditation tasks that are assigned to collect and classify documents. These documents are in the forms of Portable Document Format (PDF), Document File Format (Doc), and Scanned PDF. Lighten Software Inc. ⁽¹⁾ defined Scanned PDF as an image. When you scan a paper using a scanner it becomes an image. Figure 1 shows the sample scanned document of accreditation. The traditional way of classifying these documents is dependent on the assigned accreditation task force's judgment and is, therefore, subjective. Since it is subjective, it is time-consuming. Therefore, a system that automatically classifies documents is invaluable to the assigned accreditation task force.

There are existing approaches that have been introduced. Berong et al. innovated a system that stores documents and categorizes them through tagging. They proposed a Document Management System for AACUP Accreditation Preparation with Suggestive Document Identifier ⁽²⁾. On the other hand, Estrera et al. introduced the Electronic Document Management System for Higher Education Institutions, a catalog-based system used for tracking, keeping, and transferring documents ⁽³⁾. Another study proposed an Enhanced Document Management System with Embedded Middleware for Document Uploading presented that would immediately upload a scanned document to the system with the use of middleware ⁽⁴⁾. Unfortunately, these existing approaches i.e tagging, cataloging, etc. though utilized computerized approaches, still, are influences on personal's judgment. Alejandria et al. ⁽⁵⁾ developed a system that would automatically classify documents using domain ontology. The ontology was created by writing down related words for each area of accreditation. However, the method of writing down the related words in each area using existing attachments is still subjective. Therefore, they are prone to human errors. To address these drawbacks, automatic document classification using machine learning has been introduced. Document Classification is a well-proven approach to organizing a huge volume of textual data. Organizing related documents is necessary for decision-making ⁽⁶⁾. SpoorthiM et al. ⁽⁷⁾ use supervised machine learning to classify documents that belong to four educational departments namely Civil, Computer Science, Mechanical, and Electrical Engineering. Mowafy M. et al. ⁽⁸⁾ made an efficient classification model for unstructured text documents using the 20-Newsgroups dataset and validated using statistical measures such as precision, recall, and f-score. Other previous works, Basarkar ⁽⁹⁾ and Jothi et al. ⁽¹⁰⁾ were also introduced but none of them does not include scanned documents.

Our attempt to address this challenge is to develop an automatic document classification not only for text-based but also for image-based or scanned documents used for accreditation purposes. A. Chaudhuri et al. ⁽¹¹⁾ defined Optical Character Recognition (OCR) as a widespread technology to recognize text inside images, such as scanned documents and photos. Neil Francis B. Castellano utilizes Optical Character Recognition(OCR) in the Development of a Number System Converter Application ⁽¹²⁾. Tesseract identifies characters in foreground pixels called blobs and then it finds lines. Recognition involves converting images to character streams representing letters of recognized words ⁽¹³⁾. In short, it extracts texts out of the images (scanned documents) and then recognizes them. It means that its accuracy depends on how pixelated an image is. Consequently, an increasing number of algorithms have been developed to accomplish automatic document classification. Among these approaches, Naive Bayes classifier is still highly useful to classify the categorical and numerical variables ⁽¹⁴⁾ especially comparing its performance with other classifiers. Sebastian Rashca ⁽¹⁵⁾ presents that for small datasets naïve Bayes classifiers can outperform the more powerful alternatives. Feng Jiang et. al experiments' on the comparison and analysis of Naive Bayes classifier with Term Frequency Inverse Document Frequency (TF-IDF) and the results showed that TFIDF NM or Term Frequency Inverse Document Frequency Multinomial Naive Bayes algorithm can be used for text categorization as well ⁽¹⁶⁾. Given the size of our dataset and the consideration for computational time, we find this technique appropriate for our automatic document classification.

2 Methodology

2.1 Framework of the proposed objective accreditation document classification

This study performs an objective classification of accreditation documents whether text-based or image-based documents. Image-based documents are scanned documents (pdf/png) and text-based documents are normal documents (.docx,pdf). We upload the document in the system and it then classifies as which area it belongs to (e.g. Area I, II,..., X.)

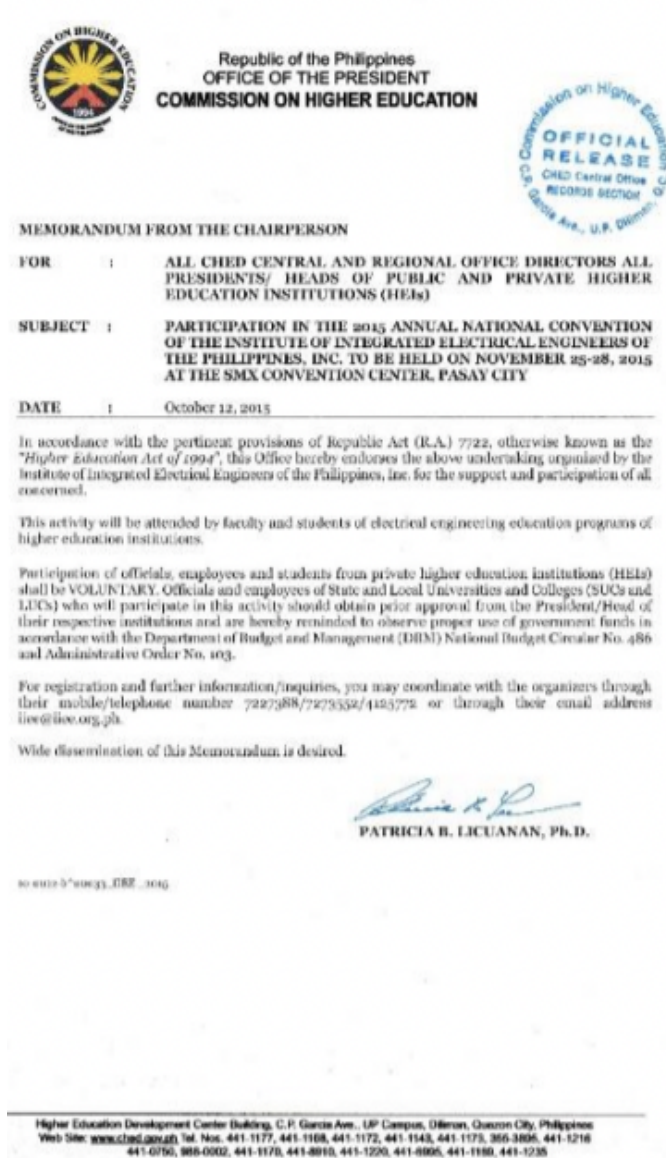


Fig 1. Sample scanned accreditation document

The framework of the proposed objective accreditation document classification is presented in Figure 2. Documents are collected from each area of accreditation. For scanned or image-based documents, texts are being extracted through Optical Character Recognition (OCR).

Once texts are collected, it is ready for pre-processing. In the rest of this section, we describe the details of the pre-processing of texts such as removal of the stop words, and a bag of words. To extract features from the texts, we use the technique term frequency-document inverse frequency (TF-IDF) followed by training our classifier which is Naïve Bayes.

2.2 Text Collecting

We collected both normal files and scanned files. We then selected the scanned files and converted them into text files or extracted the text out from them. After which, we then combined the extracted text files to the normal files as part of the collection of documents for an area e.g Area I.

We extracted the text using Optical Character Recognition (OCR). The following are the steps in OCR: 1. Pre-processing 2. Segmentation 3. Feature Extraction 4. Classification Once we extracted all the texts from both scanned and text-based

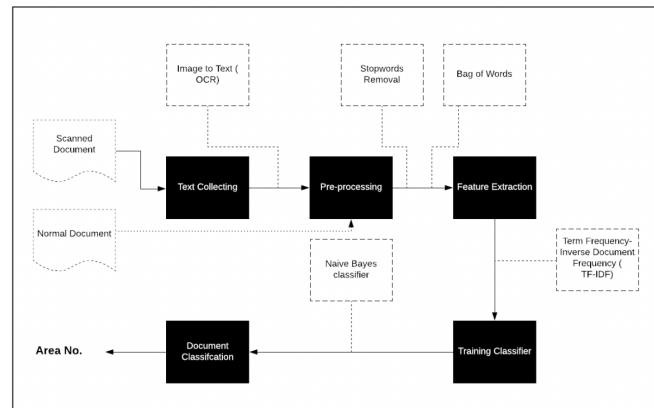


Fig 2. The framework of the proposed objective accreditation document classification

documents, we now have our set of text data ready for pre-processing. Figure 3 is a snippet of a text data set.

Out[3]:	Category	Content
0	Area1	VISION A nationally-recognized Science and Tec...
1	Area2	MISAMIS UNIVERSITY Oroquieta City TO \$ ARCHIE ...
2	Area3	The Bachelor of Science in Information Technol...
3	Area4	AREA IV\nSUPPORT TO STUDENTS\nSTUDENT SERVICES...
4	Area5	\nResearch is one of the major functions of th...
5	Area6	the university extension and community outreac...
6	Area7	Chapter 7. _The Library Services Division\n\nA...
7	Area8	Rules and Regulations\n\nPriority in the use o...
8	Area9	UNIVERSITY OF\n\nSCIENCE AND TECHNOLOGY\nOF SO...
9	Area10	PART ONE. GOVERNMENTAL AND ADMINISTRATIVE AFFA...
10	Area1	ABOUT US The University of Science and Technol...
11	Area2	Capitol University Cagayan de Oro City PERSONN...

Fig 3. Text data set

2.3 Pre-processing

For this study, the researchers find it's best fit to use the bag-of-words approach with stop word removal only as stemming can create non-real words, such as "provid" (from provide) while lemmatization, though can form individual words (which are grammatically correct), it is computationally more difficult and expensive.

2.4 Stopwords Removal

Stopwords are words that are filtered out before processing the text. Stopwords are common words in the English language such as 'the', 'is', 'at' etc. Removing stop words can potentially help improve the performance as there are fewer and only meaningful tokens left. Also, it won't take up space in the valuable processing time. This is done once we extract the text from the uploaded file. Figure 4 shows the terms being removed in the text data set.

2.5 N-gram of 1-2

The contiguous sequence of items in NLP – words, letters, or symbols is n-gram. The choice of the number n in the n-gram depends on the particular application⁽¹⁵⁾. Since accreditation words may require 1-2 words such as "information technology" etc., we find this method appropriate for this study.

frozenset({'hereafter', 'latterly', 'describe', 'may', 'how', 'empty', 'thereupon', 'through', 'ourselves', 'under', 'were', 'h
erself', 'hence', 'serious', 'for', 'first', 'few', 'everywhere', 'each', 'give', 'then', 'bottom', 'here', 'by', 'our', 'hereb
y', 'would', 'whose', 'latter', 'into', 'therein', 'therein', 'except', 'twelve', 'com', 'nothing', 'who', 'fill', 'what', 'our
s', 'above', 'much', 'name', 'last', 'so', 'and', 'eleven', 'noone', 'thus', 'around', 'my', 'on', 'namely', 'same', 'became',
'part', 'most', 'often', 'me', 'still', 'the', 'an', 'wherein', 'very', 'forty', 'perhaps', 'take', 'none', 'will', 'when', 'ar
e', 'found', 'myself', 'as', 'beforehand', 'from', 'inc', 'due', 'yourself', 'amongst', 'whether', 'former', 'while', 'among',
'de', 'your', 'another', 'in', 'wherever', 'elsewhere', 'moreover', 'ten', 'cannot', 'system', 'why', 'any', 'him', 'otherwis
e', 'his', 'two', 'sincere', 'such', 'they', 'being', 'amongst', 'until', 'that', 'do', 'onto', 'somewhere', 'because', 'get',
'neither', 'others', 'third', 'against', 'those', 're', 'about', 'an', 'over', 'between', 'anyway', 'could', 'fifteen', 'toward
s', 'everything', 'find', 'has', 'he', 'less', 'also', 'sixty', 'seemingly', 'her', 'during', 'made', 'was', 'its', 'whither', 'a
nyone', 'whole', 'beyond', 'therefore', 'per', 'there', 'whoever', 'next', 'already', 'etc', 'done', 'five', 'it', 'nine', 'n
o', 'whereafter', 'this', 'alone', 'indeed', 'again', 'sometime', 'detail', 'along', 'will', 'or', 'these', 'either', 'else',
'upon', 'via', 'whereby', 'since', 'she', 'even', 'whereas', 'thereafter', 'twenty', 'is', 'move', 'besides', 'off', 'hasnt',
'yet', 'now', 'show', 'out', 'mine', 'should', 'go', 'everyone', 'some', 'someone', 'one', 'all', 'sometimes', 'ie', 'have', 'm
ust', 'four', 'might', 'rather', 'eg', 'throughout', 'its', 'somehow', 'whereupon', 'enough', 'with', 'can', 'anywhere', 'bil
l', 'to', 'cant', 'yours', 'become', 'eight', 'itself', 'many', 'toward', 'together', 'below', 'keep', 'thence', 'hundred', 'af
ter', 'thru', 'i', 'hereupon', 'hers', 'seemed', 'themselves', 'thin', 'side', 'however', 'own', 'becomes', 'had', 'co', 'becom
ing', 'three', 'too', 'back', 'least', 'though', 'be', 'although', 'within', 'anyhow', 'once', 'whom', 'further', 'full', 'up',
'seem', 'which', 'if', 'before', 'every', 'couldnt', 'been', 'front', 'other', 'where', 'us', 'several', 'even', 'please', 'we
never', 'fire', 'something', 'see', 'meanwhile', 'without', 'beside', 'but', 'you', 'itd', 'whatever', 'interest', 'cal
l', 'nevertheless', 'never', 'amount', 'not', 'nowhere', 'thereby', 'in', 'cry', 'almost', 'anything', 'fifty', 'nobody', 'a',
'formerly', 'we', 'himself', 'then', 'behind', 'nor', 'mostly', 'top', 'their', 'whence', 'thick', 'than', 'of', 'both', 'yours
elves', 'always', 'across', 'well', 'only', 'afterwards', 'down', 'seems', 'at', 'put'})

Fig 4. Removed terms from the text data set

Bag-of-words (BOW) model allows us to represent text as numerical feature vectors or we simply build feature vectors from a text data set⁽¹⁵⁾. The following are the steps of BOW.

1. Build a vocabulary of words or tokens from a text data set (all documents).
2. Construct a **feature vector** from each document containing the frequency of how often each word occurs based on the vocabulary.

Figure 5 presents the vocabulary of words from a text data set. A total of 74,580 words.

```
Out[15]: dict_items([('vision', 3278), ('nationally', 2826), ('recognized', 2524), ('science', 2744), ('technology', 3869), ('universi  
ty', 3280), ('providing', 2440), ('vital', 3282), ('link', 1814), ('education', 1814), ('economy', 1813), ('mission', 1972),  
'(bring', 432), ('world', 3348), ('work', 3341), ('industry', 1554), ('actual', 153), ('higher', 1454), ('training', 3149),  
'(students', 2988), ('offer', 2094), ('entrepreneurs', 1183), ('opportunity', 2127), ('mainline', 1921), ('business', 451),  
'(potential', 2123), ('game', 1135), ('service', 2798), ('product', 2388), ('conceptualization', 652), ('commercializatio  
n', 613), ('contribute', 728), ('significantly', 2838), ('national', 2825), ('development', 801), ('goals', 1368), ('food', 1  
287), ('security', 2765), ('energy', 1876), ('sufficiency', 2994), ('solutions', 2876), ('misamis', 1103), ('coroqueta', 214  
8), ('city', 562), ('archie', 279), ('public', 2170), ('office', 2097), ('president', 2398), ('subject', 2974), ('appointmen  
t', 202), ('date', 818), ('june', 1711), ('2006', 43), ('appointed', 268), ('probationary', 2378), ('instructor', 1684), ('ef  
fective', 1817), ('end', 1863), ('semester', 2775), ('school', 2742), ('year', 3359), ('2007', 44), ('understood', 3158), ('e  
xisting', 1356), ('law', 1761), ('policies', 2280), ('standards', 2929), ('rules', 2699), ('regulations', 2566), ('promulgat  
ed', 2418), ('shall', 2807), ('terminate', 3084), ('expiration', 1172), ('indicated', 1545), ('hereof', 1448), ('written', 33  
54), ('notification', 2864), ('guide', 1489), ('accordingly', 126), ('bachelor', 358), ('information', 1559), ('built', 44  
8), ('program', 2401), ('includes', 1530), ('study', 2972), ('utilization', 3238), ('hardware', 3428), ('software', 2871),  
'(technologies', 3868), ('involving', 1656), ('planning', 2270), ('installing', 1591), ('customizing', 880), ('operating', 21  
28), ('innovating', 1573), ('managing', 1885), ('administering', 160), ('maintaining', 1878), ('infrastructure', 1552), ('pro  
vides', 2439), ('computing', 659), ('address', 168), ('needs', 2037), ('organization', 2136), ('installs', 1593), ('utps', 32  
33), ('graduate', 1378), ('attributes', 244), ('prepare', 2355), ('various', 3251), ('user', 2240), ('limited', 1818), ('sele  
ction', 2778), ('application', 254), ('innovation', 1574), ('integration', 3212), ('management', 1884), ('local', 1833), ('pl
```

Fig 5. Word Vocabulary

As an example, consider these first 5 documents in our text data set as shown in Table 1. As step 1 in BOW stated earlier, we build a vocabulary out of these text documents as shown in Figure 6. We reserve the 6th document as our test data, see Figure 7. With stopwords removal, the size of the vocabulary is 13. The following terms were generated (with their index no.): 'education':0,'higher':1,'higher education':2,'information technology':4 and so on. See Figure 8.

Document	Category	Content
D1	Area1	VISION A nationally-recognized Science and Tec...
D2	Area2	Chapter 46. Leave Privileges Art. 248. Vacatio...
D3	Area3	The Bachelor of Science in Information Technol...
D4	Area4	AREA IV SUPPORT TO STUDENTS STUDENT SERVICES P...
D5	Area5	Research is one of the major functions of th...

Fig 6. (Sample)Traindata

D6	Area5	research Priority Areas in Advance Science for...
----	-------	---

Fig 7. (Sample)Test data

As step 2, **Figure 9** shows the feature vector from each text document. This means that the rows are the documents, and the columns are the terms. Others call this a term-document matrix. So, the first column here shows that there are 2 counts for the word 'education' in the first document (Area I) and 1 count for the word 'higher', and so forth.


```
{'science': 6,
'technology': 10,
'university': 11,
'education': 0,
'higher': 1,
'students': 9,
'leave': 3,
'vacation': 12,
'sick': 7,
'information': 2,
'program': 4,
'student': 8,
'research': 5}
```

Fig 8. (Sample)Vocabulary

```
Out[16]: array([[ 2,  1,  1,  0,  0,  0,  0,  0,  0,  1,  2,  1,  0],
 [ 0,  0,  0,  0,  0,  9,  0,  0,  7,  0,  0,  1,  9],
 [ 0,  0,  0,  5,  3,  0,  2,  0,  0,  1,  3,  0,  0],
 [ 0,  0,  0,  2,  2,  0,  4,  0,  0,  3,  2,  0,  0],
 [ 4,  4,  4,  0,  0,  0,  1, 10,  0,  1,  1,  4,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  3,  0,  0,  0,  0,  0]], dtype=int64)
```

Fig 9. Feature Vector from each text document

2.7 Feature Extraction

A very common feature extraction procedure for sentences and documents is the bag-of-words approach (BOW). In this approach, we look at the histogram of the words within the text, i.e. considering each word count as a feature⁽¹⁵⁾.

The Bag of words (BOW) is simply an algorithm that counts the frequency of a word appearing in a document. However, the drawback of BOW is that the "rareness" of the term is not considered⁽¹⁷⁾. To overcome this drawback, we use the TF-IDF technique. John Mueller of Google⁽¹⁸⁾ affirmed that Google has been using the TF-IDF algorithm in search engine optimization (SEO). Bartosz Goralewicz⁽¹⁷⁾ explains the TF-IDF algorithm as a guide to all content writers and SEO experts.

2.8 Term Frequency –Inverse Document Frequency

This study uses TF-IDF technique to down weight those frequently occurring words in the feature vectors as shown in Figure 6. It can be defined as the product of the term frequency and the inverse document frequency as shown below.

$$tf-idf(t, d) = tf(t, d) \times idf(t, d) \quad (1)$$

where $tf(t, d)$ is the term frequency for each document and the $idf(t, d)$ can be calculated as:

$$idf(t, d) = \log \frac{n_d}{1 + df(d, t)} \quad (2)$$

where n_d is the total number of documents, and $df(d, t)$ is the number of documents d that contain the term t . Note that adding the constant 1 to the denominator serves the purpose of assigning a non-zero value to terms that occur in all training samples; the log is used to ensure that low document frequencies are not given too much weight.

Figure 10 below shows the TF-IDF values of the feature vector from each text document. This means that the first column here shows that the TF-IDF value for the word ‘education’ is 0.6414933 and 0.32074667 for the word ‘higher’, for the first document (Area I), and so forth. Terms with values other than zeros are rare terms and thus, important words.

[0.64149333	0.32074667	0.32074667	0.	0.	0.
0.	0.20039539	0.	0.23205192	0.46410384	0.27079638	
0.]
[0.	0.	0.	0.61888304	0.	
0.	0.	0.48135348	0.	0.	0.04760671	
0.61888304]						
[0.	0.	0.	0.8641825	0.	0.2918409
0.	0.10798441	0.	0.12504274	0.37512822	0.	
0.]
[0.	0.	0.	0.42070898	0.	0.71038286
0.	0.13142481	0.	0.45655811	0.30437207	0.	
0.]
[0.3147026	0.3147026	0.3147026	0.	0.	0.06642339
0.78675649	0.0491548	0.	0.0569198	0.0569198	0.26569356	
0.]
[0.	0.	0.	0.	0.	0.
0.9789949	0.20388476	0.	0.	0.	0.	
0.]]

Fig 10. Tf-idf values for each text document

2.9 Naïve Bayes

Empirical and theoretical results, Naive Bayes classifier is still highly useful to classify the categorical and numerical variables⁽¹⁴⁾ especially comparing its performance with other classifiers.

In this study, we use the Multinomial Naïve Bayes classifier as this calculates the likelihood to be the count of word or token and considering that we have 10 classes (areas).

Shuo Xu et. al. applied Multinomial Naïve Bayes to documents and classes, it has the following:

$$P(c \setminus d) = P(d \setminus c)P(c)/P(d) \quad (3)$$

Where “d” represents a document and “c” for class.

As naïve bayes classifier, we use “maximum posteriori (MAP)” or most likely class, applying bayes rule and dropping the denominator:

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(d \setminus c)P(c) \quad (4)$$

$c \in C$

Now as we represent the document as the feature (terms) we have the following:

We assume the feature probabilities are independent given class c.

$$\begin{aligned} c_{MAP} &= \operatorname{argmax}_{c \in C} P(d \setminus c)P(c) \\ &= \operatorname{argmax}_{c \in C} P(x_1, x_2, x_n \mid c)P(c) \end{aligned} \quad (5)$$

$$P(x_1, \dots, x_n \mid c) = P(x_1 \mid c) * P(x_2 \mid c) * P(x_3 \mid c) * \dots * P(x_n \mid c) \quad (6)$$

Now our final formula for multinomial naïve bayes classifier is:

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n \mid c)P(c) \quad (7)$$

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \| P(x \mid c)$$

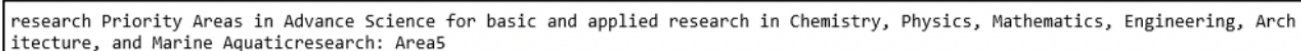
$c \in C, x \in X$

In what follows, we describe the steps of naïve bayes classifier:

1. Convert documents to feature sets where the attributes are the possible words and the values are the tf-idf of the word occurring in the document.

Looking at documents classified as "Area1", "Area2", "Area3", and so on 3 Classification

Figure 11 shows the sample result after feeding the test data in the created predictive model. The terms, "research Priority Areas in Advance Science for basic and applied research in Chemistry, Physics, Mathematics, Engineering, Architecture, and Marine Aquatic research," were fed in the model and it correctly classified "area5".

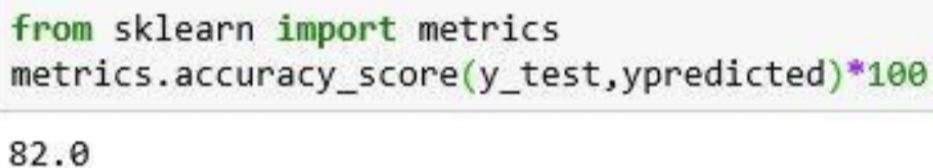


```
research Priority Areas in Advance Science for basic and applied research in Chemistry, Physics, Mathematics, Engineering, Arch
itecture, and Marine Aquatic research: Area5
```

Fig 11. Sample result executed in Python Jupiter

3 Results and Discussion

In our experiment, we used a total of **1000 documents** (pages vary), ten (10) for each area. This has 74,580 terms after removing the stop words. We have split the text data set into train and test data. The size of our train data is 90% of the entire dataset while our test size is 10%. First, we trained the train data and created a classification model out of this. After doing so, we then test the model using unseen test data to assess the accuracy model. Figure 12 shows the train test of 90-10 split results. The result shows an 82% accuracy score.



```
from sklearn import metrics
metrics.accuracy_score(y_test,ypredicted)*100

82.0
```

Fig 12. Train test split result

Scanned documents in accreditation are image documents. An approach of using optical character recognition (OCR) tesseract engine helped the researchers to extract text from the image-based documents and used these for pre-processing and feature extraction.

The term frequency-inverse document frequency (Tf-idf) is another alternative for extracting features in corpora. The TF-IDF approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents. Although it has been used by

Google for search engine optimization, true enough, it can also be applied to document classification via multinomial naïve Bayes. Theoretical and empirical results Forman et. al.(2004) and Ng et. al. (2001) reveal that the Naïve Bayes classifier is a good algorithm in text classification. A closer look at our result as shown in Table 1 reveals that this classifier is efficient.

Table 1 shows the precision, recall, and f-1 score of the experiment.

The average precision is 84%, recall is 82% and f-1 score is 82%.

Since our dataset is small, we adopt the k-fold cross-validation test scheme in forming the predictive model. In each iteration, we leave one fold-out and consider it as our test set and utilize the remaining 9 folds as our training set to learn a model. Since we have 10 folds, we do this process for 10 repetitions.

Table 2 shows the accuracy scores for each iteration. In the given result, the average of the total accuracy is 82%.

Table 3 shows the performance of the program in terms of its speed in uploading and classifying documents. There are 100 trials performed by the researcher and the result shows that there are variations of the runtime score per trial. The research performed the testing with 10 documents tested in 10 trials each with the same number of pages. Each classification not only classified 1 area but the top 3 areas the document could belong to. The computer specifications used were Intel Core i-5, 64-bit Operating System, RAM 4.00 GB. The average speed of the system is 9.037 seconds.

Table 1. Precision, recall, and f-1 score per area

	precision	recall	f1-score	support
1	0.62	0.73	0.67	11
2	0.80	0.73	0.76	11
3	0.50	0.75	0.60	4
4	0.83	0.83	0.83	12
5	1.00	1.00	1.00	7
6	0.82	0.90	0.86	10
7	1.00	0.80	0.89	10
8	0.75	0.90	0.82	10
9	1.00	0.92	0.96	12
10	0.90	0.69	0.78	13
avg/total	0.84	0.82	0.82	100

Table 2. Accuracy scores per iteration

Iteration	Score
1	80%
2	82%
3	75%
4	84%
5	82%
6	86%
7	82%
8	79%
9	79%
10	89%

Table 3. System Performance Testing

	Trial(s)									
Doc	1	2	3	4	5	6	7	8	9	10
1	8.92	7.76	15.5	8.6	8.12	11.9	5.99	7.64	9.85	6.58
2	6.17	11.57	7.84	7.02	5.96	10.45	6.12	5.7	6.21	6.13
3	5.53	7.41	13.09	12.94	15.6	11.84	5.9	6.04	5.95	5.6
4	5.56	7.18	15.06	9.3	14.77	9.51	5.79	5.71	5.99	5.87
5	5.61	7.1	9.92	12.82	11.06	10.87	5.93	7.55	6.47	5.83
6	10.34	12.4	10.56	24.95	16.01	17.04	10.77	11.71	8.742	10.72
7	3.87	11.68	12.16	12.19	19.9	19.38	8.56	8.99	11.46	10.39
8	7.04	0.65	11.21	12.8	12.89	12.63	5.49	6.62	6.96	5.93
9	5.74	8.41	14.12	11.88	13.12	6.85	5.58	6.95	5.73	6.11
10	8.78	14.1	9.71	12.41	11.96	7.8	8.77	8.58	7.28	7.57
AVE	6.756	8.826	12.55	12.939	11.827	6.89	7.549	7.3642	7.073	9.379s

4 Conclusion and Recommendations

The study has undergone tests to know the effectiveness of the automatic classification of accreditation documents. Based on the results, the accuracy score is 82%, the average precision is 84%, recall is 82% and f-1 score is 82%. Moreover, the average speed of the system in classifying documents is 9.037 seconds. Therefore, extraction of text from text-based or image-based documents with TF-IDF feature selection and Naive Bayes classifier provides an efficient way of objective automated classification of accreditation documents. It gives an avenue to address limiting factors of the previous works, i.e. subjective and time-consuming classification.

However, there are weak points such as the classification of documents with tables. Anyways, several merits can be applied in real-life scenarios in accreditation. It gives aid to whoever prepares the documents. Furthermore, this is not only applicable to accreditation but to other endeavors that may require document classification.

References

- 1) Lighten Software Inc. How can you distinguish scanned PDF from a normal PDF file?. Available from: [https://www.lightenpdf.com/knowledge-base/scanned-pdf-ocr.html\(12.11.2018\)](https://www.lightenpdf.com/knowledge-base/scanned-pdf-ocr.html(12.11.2018)).
- 2) Berong. Document Management System For AACUP Accreditation Preparation with Suggestive Document Identifier. (Unpublished manuscript). University of Science and Technology of Science and Technology of Southern Philippines (USTP), Cagayan de Oro, Philippines. Philippines. 2017.
- 3) Estrera P. Electronic Document Management System for Higher Education Institutions. *International Journal of Innovative Science and Research Technology*. 2017;2(5). Available from: <https://ijisrt.com/wp-content/uploads/2017/06/Electronic-Documents-Management-System-for-Higher-Education-Institution-1.pdf>.
- 4) Mata. Document Management System with Embedded Middleware for Document Uploading. (Unpublished manuscript). University of Science and Technology of Science and Technology of Southern Philippines (USTP), Cagayan de Oro, Philippines. 2017.
- 5) Alejandria. Semantic Analysis of Accreditation documents using domain ontology. *International Journal of Innovative Research in Science, Engineering and Technology*. 2018;6(5).
- 6) Bafna P, Pramod D, Vaidya A. Document clustering: TF-IDF approach. In: IEEE international conference on electrical, electronics, and optimization techniques (ICEEOT). 2016;p. 61–66. doi:10.1109/ICEEOT.2016.7754750.
- 7) Spoorthi M. Automatic educational document classification using natural language processing. *International Journal of Engineering Trends and Technology (IJETT)*. 2016;35(4):152–155. Available from: <https://studylib.net/doc/12917730/automatic-educational-document-classification-using-natur>.
- 8) Mowafy M. An Efficient Classification Model for Unstructured Text Document. *American Journal of Computer Science and Information Technology*. 2018;6(1):16. doi:10.21767/2349-3917.100016.
- 9) Basarkar A. Document Classification Using Machine Learning, San Jose State University Scholar Works. 2017. Available from: <https://doi.org/10.31979/etd.6jmu-9xdt>.
- 10) Saranyajothi C, Thenmozhi D. Machine Learning approach to Document Classification using Concept based Features. *International Journal of Computer Applications*. 2015;118(20):33–36. doi:10.5120/20864-3578.
- 11) Chaudhuri A, Mandaviya K, Badelia P, Ghosh SK. Optical Character Recognition Systems for Different Languages with Soft Computing. 2017. doi:10.1007/978-3-319-50252-6.
- 12) Casillano NFB. Utilization of Optical Character Recognition (OCR) in the Development of a Number System Converter Application. *Indian Journal of Science and Technology*. 2019;12(16). doi:10.17485/ijst/2019/v12i16/137794.
- 13) Selcukalgun. Review for Tesseract and KrakenOCR for text recognition. 2018. Available from: [https://medium.com/datadriveninvestor/review-for-tesseract-and-kraken-ocr-for-text-recognition-2e63c2adedd0\(12.07\)](https://medium.com/datadriveninvestor/review-for-tesseract-and-kraken-ocr-for-text-recognition-2e63c2adedd0(12.07)).
- 14) Rasjid ZE, Setiawan R. Performance Comparison and Optimization of Text Document Classification using kNN and Naïve Bayes Classification Techniques. *Procedia Computer Science*. 2017;116:107–112. Available from: <https://doi.org/10.1016/j.procs.2017.10.017>.
- 15) Raschka S. Python Machine Learning 1st Edition September 23, 2015. Packt Publishing; 1 edition.. 2016.
- 16) Jiang F, Zhang Z, Chen P, Liu Y. Naive Bayes Text Categorization Algorithm Based on TF-IDF Attribute Weighting. *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence - CSAI '18*. 2018;18. doi:10.1145/3297156.3297256.
- 17) Goralewicz B. The TF-IDF Algorithm Explained. Available from: <https://www.onely.com/blog/what-is-tf-idf/>.
- 18) Mueller J. Google's John Mueller Discusses TF-IDF Algo. 2019. Available from: <https://www.searchenginejournal.com/google-tf-idf/304361/#close>.