Automated Document Classification for HEI research grant awards using Machine Learning

[1] Rebecca Lupyani
University of Zambia
Department of Computer Science
Lusaka, Zambia
rebecca.lupyani@unza.cs.zm

[2] Jackson Phiri
University of Zambia
Department of Computer Science
Lusaka, Zambia
jackson.phir@unza.cs.zm

***Abstract*— The recent advances in information technology has resulted into a continual increase of electronic textual documents. The need to classify these documents according to their subject or related content has become pragmatic for decision making and policy makers. This paper explores the use of the Support Vector Machine Model which is considered one of the most popular text classification models. The model was trained with two different datasets; the S2ORC and the dataset obtained from the University of Zambia-Institutional Repository (UNZA-IR). The model performed generally well using the S2ORC but did not perform well when trained with the UNZA-IR dataset due to its small size. The research therefore recommends merging the two datasets with the hope of improving the performance of the model and/ or building a larger corpus of Zambian electronic thesis, dissertations and articles to make the dataset size satisfactory for training.**

Keywords— text classification, machine learning, research grant award

I. INTRODUCTION

The recent advances in information technology has resulted in a continual increase of electronic textual documents from different sources. The need to classify these documents according to their subject or related content has become pragmatic for decision making in many sectors including the research and development. Policy makers responsible for research and development are interested in monitoring and tracking how much money they are spending on research and development activities in specific fields or topics in Higher Education Institutions (HEIs). However, records are rarely classified in ways that will inform policy and budget decisions. These sets of investments are characterized frequently as "research portfolios," where a funding institution's budget is classified into groups based on key interests such as socioeconomic objective, discipline, program area, to mention but a few [1].

Funding institutions have a specific interest in performing portfolio analysis to aid their decisions. Scrivastava et al [2] suggest a growing increase in the need to apply technology to make the management of research portfolios more scientific and to enable a more quantitative approach to portfolio analysis and management. One common way in which technology is being applied is through the use of tagging systems in which documents are classified by an individual into respective disciplines and manually tagged according to awards given. Keyword searches are then used by funding institutions to search for information in a specific research discipline and therefore use this information for their research and development portfolio analyses for the awarding of grants [2]. Whereas this approach may be effective, it may result in inaccuracies and inconsistencies as the individual applying the tags may not be aware of all possible information in the document, or may not always know under which discipline certain research documents belong to and therefore may not select the most salient tags.

Mutale and Phiri [3] developed a Web Based Document Archiving Using Time Stamp and Barcode Technologies, in an attempt to improve the tagging and classification of documents. Although this system improved the tagging process through the use of barcodes, the classification process is still dependant on an individual to classify the documents according to subject area and content. Thus an automated classification and tagging using machine learning would be a better approach. Text classification algorithms are the machine learning techniques used for document classification. A text classification algorithm can be defined as the method that is used to automatically categorise a group documents into one or more predefined classes according to their subjects [4]. Text classification techniques are paramount to information retrieval systems as they enable users to extract documents in an easier, faster and more efficient way. They stand at the core of many information management and retrieval tasks as they assist in the effective and efficient processes of organizing, classifying, searching and concisely retrieving information [5]
This paper explores how machine-learning techniques specifically text classification techniques can be used to automatically classify research documents according to specific disciplines and to aid funding institutions in performing research portfolio analyses for the awarding of grants in Higher Education Institutions in Zambia.

II. RELATED WORK

With the popularity of electronic documents, a number of text classification techniques of both supervised and unsupervised learning have emerged over the years. Researchers have continually endeavored to determine text classification

methods and techniques that can yield better performance of text classification and document categorisation. Some studies undertaken have revealed that document classification performance is affected by the type of text classification algorithm used, the methods of feature extraction and the datasets used for training and evaluating the models [6]. Thus, a number of studies have been undertaken to compare the performance of text classification algorithms, to determine the best performing feature extraction methods as well as to determine the recommended datasets that can be used. Nidhi and Gupta [7] listed Nearest Neighbour classifier, Bayesian Classification, Support Vector Machine (SVM), Association Based Classification, Term Graph Model, Centroid Based Classification, Decision Tree Induction and Classification using Neural Network as some of the most popular. The study revealed that the Nearest Neighbour classifier also referred to as the KNN classifier is a simple, valid and non-parameter method algorithm. It supports only two parameters and its implementation is easy. However, it does not support a large dataset as it can be costly to implement [6]. Nidhi and Gupta [7] explained that a Bayesian classifier, constructs a probabilistic model of the features and uses that model to predict the classification of a new example. The Naïve Bayes models for text classification are Multi-Variate. Bernoulli Event Model and the Multinomial Event Model. They suggested that Navie bayes is fast and easy to implement so it is most popular and performs well. It is also computationally inexpensive and needs a very low amount of memory [8]. However, Ikonomakis [9] argues that its performance is often degraded because it does not model text well.

Another most popular classifier as discussed by Nidhi and Gupta [7] is the SVM which is a high accurate machine learning method for text classification. They suggested that SVM attempts to find an optimal hyperplane within the input space so as to correctly classify the binary or multiclass classification problems. They further suggested that SVM is less susceptible to over fitting than other learning method and that it produces best result for both test and training data set [8]. Another text classification reviewed by the study is the Association based classification model. The researchers suggest that this model utilises the association rule mining, it has a high classification accuracy and that it is more flexible to handle text data [8]. Another model is the Cetroid based classification model which creates a centroid per class of the document. It is a simple and efficient method and It is also easy to implement and flexible for text data [8].

Nidhi and Gupta [7] also suggested that tree-based classifiers such as decision tree induction is also a widely used inductive learning method with respect to document classification. They explained that a decision tree induction model is represented as a flow chart or like a tree structure with each branch representing the outcomes and the node representing the test. The model also has a leaf node which represents and holds a class label. This model is simple and understandable especially when dealing with noisy data but cannot be guaranteed for global use [8]. LeCun [10] also carried a study that suggested that deep learning approaches such as neural networks have also achieved surpassing results in comparison to some machine learning algorithms on tasks such as image classification, natural language processing, face recognition to mention but a few. The researcher explained that the success of these deep learning algorithms relies on their capacity to model complex and non-linear relationships within data and when underlying assumption are satisfied. [10]

Other researchers have further compared which of the classifiers performs better in document classification. A study carried out by Chauhan and Desai compared three popular text classifiers; Nearest Neighbor Classifier (KNN), Bayesian Classifier and Support Vector Machine (SVM). Their findings revealed that SVM produces the highest accuracy and thus is one of the most popular models in text classification [8]. On the other hand Ikonomakis et al [9] suggest that even though SVM provides excellent precision it has a poor recall and is more complex to implement [9]. Nevertheless, this has not prevented the model from emerging as the most popular text classifier.

In another study, Safder et al [11] proposed a set of methods to automatically identify and extract algorithmic pseudo-codes and the sentences that convey related algorithmic metadata for scholarly documents using a set of machine-learning techniques. The researchers carried out an experiment with over 93,000 text lines and introduced 60 novel features, comprising content-based, font style based and structure-based feature groups, to extract algorithmic pseudo-codes. Their proposed pseudo-code extraction method achieved a 93.32% F1-score, thereby outperforming state-of-the-art techniques by 28%. Additionally, they proposed a method to extract algorithmic-related sentences using deep neural networks and achieved an accuracy of 78.5%, outperforming a Rule-based model and a support vector machine model by 28% and 16%, respectively. [11] It can therefore be concluded that the pseudo-code extracted method using deep neural networks performed better than both the rule-based model and the Support vector machine.

Zhang et al [12] designed a multi-class text classification using character-level Convolutional Neural Networks (CNN) on large-scale datasets. They compared the performance of this model to the performance of traditional models, such as Bag of Words (BOW), N-grams, TF-IDF, and deep leaning word-based models such as CNN and Long Short-Term Memory (LSTM) models. Their results concluded that the baseline models overcame the proposed deep learning models with small sample sets; whereas, in the case of large-scale datasets, character level deep learning approaches were superior [12].

Conneau et al. proposed deep architectures for text classification inspired by computer vision deep networks. The networks operate directly at the character level and employ small convolutions and pooling layers. Their study argued that the deep convolutional layers increase the performance of deep learning models for text classification [13]. In another study, Mai et al. [14] proposed a document title-based semantic subject indexing approach using deep learning Multilayer Perceptron MLP, CNN and Recurrent neural networks (RNN) models. Their results revealed that

their proposed deep learning models outperformed the available full-text systems by a large margin. [14]

Another approach that researchers have explored in seeking machine-learning algorithms suitable for document classification is in examining the feature extraction methods used when building text classification models. Feature extraction is defined as the process of transforming raw data into numerical features that can be understood while preserving the information in the original data set [6]. Texts and documents are unstructured data sets and so they must be converted into a structured feature space when using mathematical modeling as part of a classifier. Feature Extraction is applied to the data after it has been cleaned to omit unnecessary characters and words [6].
After the data has been cleaned, formal feature extraction methods can be applied. The common techniques of feature extractions are Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF), Word2Vec, and Global Vectors for Word Representation [6].

Basarkar [15] carried out a study to discuss the different types of feature vectors through which documents can be represented and later classified. The study aimed at comparing the Binary, Count and TF-IDF feature vectors and their impact on document classification. In order to examine how well each of the three mentioned feature vectors perform. Basakar used 20-newsgroup datasets which were converted to all the three feature vectors. The Naives bayes classifier was trained for each feature vector representation, and then evaluated on using test documents. The results of the study where that TF-IDF performed 4% better than Count vectorizer and 6% better than Binary vectorizer if stop words were removed. If stop words were not removed, then TF-IDF performed 6% better than Binary vectorizer and 11% better than the Count vectorizer. Additionally, the Count vectorizer performed better than the Binary vectorizer by 2%, if stop words were removed but lagged behind by 5% if stop words were not removed. Thus, the study concluded that TF-IDF is the preferred vectorizer for document representation and classification. The above reviewed literature is evidence that the use of machine- learning algorithms for text and document classification has become popular. The rapid advances in information technology has led to a rise in electronic documents and as such document classification has become important in many faculties of life today. This paper explores the use of text classification techniques classifying research documents for the award of research grants.

III. MATERIALS AND METHODS

In this section, the methods and materials used to build the text classification model are briefly discussed.

A. Data Set

Two sets of datasets were used for this research; a commercial dataset that has been tried and tested and the second being a locally generated dataset.

- The Semantic Scholar Open Research (S2ORC) dataset downloaded from www.github.com. S2ORC is a large corpus of 81.1Million English-language academic papers that comprises many academic disciplines. It includes rich metadata, paper abstracts, resolved bibliographic references, as well as structured full text for the 8.1million. However, the only information that was needed for the purpose of this research are the titles of the documents and the classes of disciplines.
- A dataset was generated from the University of Zambia Institutional Repository (UNZA-IR) to evaluate the model. UNZA-IR is corpus of approximately 4000 Electronic Thesis and Dissertations from various disciplines.

B. Materials

- The project was done in a PHP Machine Learning environment and employed the Support Vector Machine (SVM) algorithm which is a supervised learning model. It was selected because of its popularity in test classification problems. Term Frequency-Inverse Document Frequency (TF-IDF) was selected vectorizer for feature selection and extraction.

C. Method

The work flow of the training process of the SVM Model for text classification is shown in Figure 1.

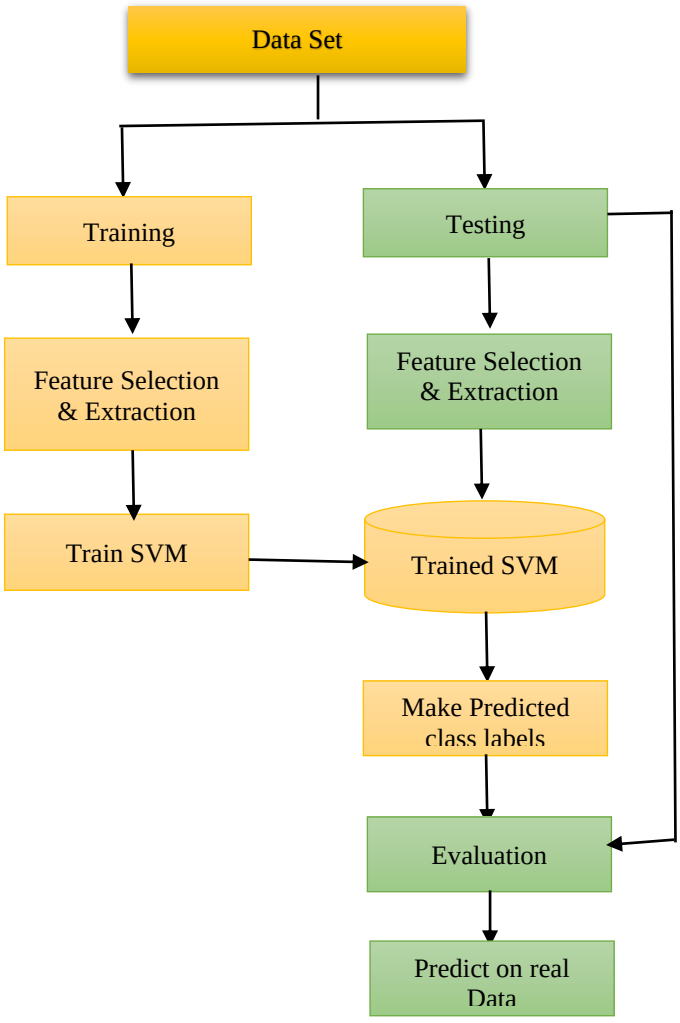

Figure 1: SVM training process work flow

i. Dataset Preparation

A document is made up of words, but not all words may be used to train the classifier. The document may also contain unnecessary things like punctuation marks, misspelled words, slang to mention but a few which can interfere with the performance of the classifier [16]. Such words and symbols need to be removed through a process called preprocessing. Therefore, the document is first broken down into words, phrases, symbols, or other meaningful elements called tokens (tokenisation) [16]. Subsequently the dataset is cleaned to remove stopwords, which refer to unnecessary words such as auxiliary verbs, conjunctions and articles. They are removed because they appear in most of the documents and may affect the performance of the classifier [17]. Another preprocess task is stemming which removes words with the same stem and keeps the stem or the most common of them as a feature. For example, the words "test", "testing", "tested" and "tests" can be replaced with "test". This process is considered to improve the performance of the classifier [17].

ii. Feature Selection and Extraction

Feature selection is a method that aims to reduce the dimensionality of the dataset by removing features that are considered irrelevant for the classification. Selecting only the most important features causes the model to focus on the key variables that have the greatest impact on the outcome, and ignore irrelevant or redundant features that may only add noise to the data [18]. This process results in training the model faster, improve its accuracy and reduce any generalization errors introduced due to noise by irrelevant features. It also reduces the chances of producing models that are prone to overfitting [18]. Feature extraction also aims to reduce the dimensionality of the dataset by extracting or deriving information from the original features set to create a new features subspace [18]. The Term Frequency-Inverse document frequency (TF-IDF) method was applied to extract the new feature set. TF counts the number of words in each document and assigning it to the feature space. IDF assigns a higher weight to words with either high or low frequencies term in the document.

Below is the code used for preprocessing and feature extraction.

```
$vectorizer = new
Phpml\FeatureExtraction\TokenCountVectorizer(new
Phpml\Tokenization\NGramTokenizer(1, 3), new
Phpml\FeatureExtraction\StopWords\English());
$tfIdfTransformer = new
Phpml\FeatureExtraction\TfIdfTransformer();
$samples = [];
foreach ($dataset->getSamples() as $sample) {
$samples[] = $sample[0];
}

## Transforming the dataset

$vectorizer->fit($samples);
$vectorizer->transform($samples);

$tfIdfTransformer->fit($samples);
$tfIdfTransformer->transform($samples);

$dataset = new
\Phpml\Dataset\ArrayDataset($samples,
$dataset->getTargets());

$randomSplit = new
\Phpml\CrossValidation\StratifiedRandomSplit($data
set, testSize: 0.2, seed:156);
```

iii. Training the Model

The Support Vector Machine (SVM) text classification model was trained using the S2ORC corpus. Then in another instance the model was trained using 80% of the dataset from UNZA-IR. Below is the code that was used.

```
$classifier->train($randomSplit-
>getTrainSamples(), $randomSplit-
>getTrainLabels());
$modelManager = new
Phpml\ModelManager();
$modelManager->saveToFile($pipeline,
'./model/dataSet .phpml');
$classifier = $modelManager-
>restoreFromFile('./model/dataSet .phpml');

making predictions with trained models
$predictedLabels = $classifier-
>predict($randomSplit->getTestSamples());
```

iv. Evaluating the Model and making predictions.

The SVM model was evaluated using the UNZA-IR dataset after being trained with the S2ORC dataset. Then in another instance it was evaluate using the 20% of UNZA-IR dataset after being trained with 80% of the UNZA-IR. The SVM model was evaluated to determine the accuracy of the predicted results. Below is the code used for evaluating the model, making predictions as well as calculating the accuracy.

```
$predictedLabels = $classifier-
>predict($randomSplit->getTestSamples());
```

```
$accuracy =
Phpml\Metric\Accuracy::score($randomSplit-
>getTestLabels(), $predictedLabels);
$category=$classifier->predict([$topic]);
```

IV. EXPERIMENTS

In order to classify documents, the developed SVM model was subjected to two different datasets, namely, the S2ORC and the UNZA-IR dataset. The SVM model was first trained using the S2ORC dataset and tested on the data from the UNZA-IR to make predictions of classification labels by title. Secondly the SVM model was trained using the UNZA-IR dataset which was split into 80% training set and 20% testing set. The results obtained are recorded in table 1 below.

Table 1: Accuracy Scores

Model	**Training Dataset**	**Testing Data Set**	**Accuracy**
SVM	S2ORC	UNZA-IR	89.473684210526%
SVM	UNZA-IR	UNZA-IR	67.185185185185%

V. RESULTS AND DISCUSSION

This section discusses the findings as obtained from the experiment carried out using two different data sets; the S2ORC and the UNZA-IR dataset.

- SVM and S2ORC training set and UNZA-IR Test Set

The results show that the model performed generally well when it was trained using the S2ORC dataset and tested on the data from UNZA-IR to make document classification predictions. The model was able to make predictions at an accuracy percentage of 89.473684210526%. The model has the potential of performing even better, the gap is as a result of differences in the jargons of those used in the S2ORC and the jargons used in the UNZA-IR which are more inclined to the local content.

- SVM and 80% UNZA_IR training set and 20% UNZA-IR Test Set

The model did not perform as well as it did when trained with the S2ORC dataset, giving a percentage accuracy of 67.185185185185%. Such an accuracy entails that some of the predictions made were not correct to a high extent. This can be attributed to the fact that the dataset was too small, thereby causing the model to be over-fitted.

VI. RECOMMENDATIONS

Ingram et al. [19] stated that the academic research corpus is an under-explored resource as a data set resulting in difficulties in obtaining satisfactory training data [19]. It was a challenge to obtain one that would be used for the purpose of this research. The greater challenge was trying to obtain a dataset with local content that is adequately large. Even though the UNZA-IR dataset was obtained, it required a lot of preprocessing as there were a number of misspelt words, stopwords, punctuation marks and null values. This research therefore makes the following recommendations:

- To build an adequately large corpus of Zambian electronic thesis, dissertations and articles that should be well labelled and tested on the Model or alternatively, to build a customized dataset that merges the S2ORC and the UNZA-IR datasets as building a novel dataset may take a considerable amount of time.
- To integrate the model into a web-based application that funding institutions in Zambia can use to classify documents for HEIs research grant awards.
- For future works, we recommend that the model be extended to use other semantic information for classification such as keywords and/ or abstracts from the documents.

I. REFERENCES

[1] I. R. Matthew L Wallace, "Research portfolio analysis in science policy: moving from financial returns to social benefits," *Minerva*, vol. 53, no. 2, 2015.

[2] C. V. Srivastava, N.D. Towery, B. Zuckerman, "Challenges and opportunities for research portfolio analysis, management, and evaluation," *Research Evaluation*, vol. 16, no. 3, pp. 152-156, 2007

[3] B. Mutale, J. Phiri, "Web Based Document Archiving Using Time Stamp and Barcode Technologies–A Case of the University of Zambia.," *International Journal of Innovative Research in Science, Engineering and Technology*, 2016.

[4] A.I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Springer link*, vol. 52, p. 273–292, 2019.

[5] R. Power, J. Chen, T. Karthik, L. Subramanian "Document Classification for Focused Topics," *Association for the Advancement of Artificial Intelligence* , vol. 1, 2010.

[6] K.Kamran, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown , "Text Classification Algorithms: A Survey," *MDPI Journals*, vol. 10, no. 4, 23 April 2019.19.

[7] V. G. Nidhi, "Recent Trends in Text Classification Techniques," *International Journal of Computer Applications*, vol. 35, no. 6, 2011.

[8] A. Desai, C. Shrihari, R. Amish "A Review on Knowledge Discovery using Text Classification Techniques in Text Mining," *International Journal of Computer Application*, vol. 111, no. 6, p. (0975 – 8887), 2015.

[9] M. Ikonomakis, S. Kotsiantis, V. Tampakas, "Text Classification Using Machine Learning Techniques," *WSEAS TRANSACTIONS on COMPUTERS*, vol. 4, no. 8, pp. 966-974, August 2005.

[10] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning. Nature," *Google Scholar*, p. 436–444, 2015.

[11] [11] X. Zhang, J. Zhao, Y.LeCun, "haracter-level convolutional networks for text classification," *Advances in neural information processing systems*, p. 649–657, 2015.

[12] A. Conneau, H. Schwenk, L. Barrault, Y Lecun, "Very deep convolutional networks for text classification.," *ECACL*, 2016.

[13] F. Mai, L. Galke,A. Scherp, A "Using Deep Learning for Title-Based Semantic Subject Indexing to Reach Competitive Performance to Full-Text," in *ACM/IEEE on Joint Conference on Digital Libraries*, ACM, 2018.

[14] B. Ankit, "Document classification using machine learning," *Master's Theses and Graduate Research at SJSU ScholarWorks*, 2017.

[15] T. Verma, R. Renu, D. Gaur, "Tokenization and filtering process in RapidMiner," *International journal of applied information systems*, vol. 7, pp. 16-18, 2014.

[16] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM*, vol. 34, no. 1, pp. 1-47, 2022.

[17] A. Kumar, "Feature Selection vs Feature Extraction: Machine Learning," *Data Analytics*, 24 March 2023.

[18] W. A. Ingram, "Summarizing ETDs with deep learning," *Cadernos BAD*, 2019S.-U. H. ,. V. ,. T. N. R. N. S. T. Iqra Safder, "Deep Learning-based Extraction of Algorithmic Metadata in Full-Text Scholarly Documents," *Information Processing & Management*, vol. 57, no. 6, 2020.

[12] X. Z. J. L. Y. Zhang, "haracter-level convolutional networks for text classification," *Advances in neural information processing systems*, p. 649–657, 2015.

[13] A. S. H. B. L. L. Y. Conneau, "Very deep convolutional networks for text classification.," *ECACL*, 2016.

[14] F. G. L. S. A. Mai, "Using Deep Learning for Title-Based Semantic Subject Indexing to Reach Competitive Performance to Full-Text," in *ACM/IEEE on Joint Conference on Digital Libraries*, ACM, 2018.

[15] B. Ankit, "Document classification using machine learning," *Master's Theses and Graduate Research at SJSU ScholarWorks*, 2017.

[16] T. Verma, R. Renu and D. Gaur, "Tokenization and filtering process in RapidMiner," *International journal of applied information systems*, vol. 7, pp. 16-18, 2014.

[17] S. F, "Machine Learning in Automated Text Categorization," *ACM*, vol. 34, no. 1, pp. 1-47, 2022.

[18] A. Kumar, "Feature Selection vs Feature Extraction: Machine Learning," *Data Analytics*, 24 March 2023.

[19] B. B. E. A. F. William A. Ingram, "Summarizing ETDs with deep learning," *Cadernos BAD*, 2019.