Automatic PDF Document Classification with Machine Learning

Sócrates Llácer Luna[1], Dario Garigliotti[2], Fernando Martínez Plumed[1], and César Ferri Ramírez[1]

[1] Universitat Politècnica de València, Spain
[2] University of Bergen, Norway

Abstract. Universitat Politècnica de València (UPV) faces challenges in managing its Alfresco document repository, which contains 600,000 PDF files, of which only 100,000 are correctly categorised. Manual classification is laborious and error-prone, hindering information retrieval and advanced search capabilities. This project presents an automated pipeline that integrates optical character recognition (OCR) and machine learning to efficiently classify documents. Our approach distinguishes between scanned and digital documents, accurately extracts text and categorises it into 51 predefined categories using models such as BERT and RF. By improving document organisation and accessibility, this work optimises UPV's document management and paves the way for advanced search technologies and real-time classification systems.

Keywords: Document Classification · OCR · Machine Learning · Alfresco Repository

1 Introduction

In the digital age, document digitisation and efficient access to information are crucial for the operation and development of modern institutions. The Universitat Politècnica de València (UPV), a public university in Spain, is currently facing a challenge in managing its Alfresco document repository, which contains approximately 600,000 PDF files. In particular, only 100,000 of these documents are correctly categorised, creating an imbalance that hinders efficient information retrieval and the implementation of advanced search and analysis systems.

Manual categorisation of such a large number of documents is not only time-consuming and labour-intensive, but also prone to error, further complicating information management. As a result, there is an urgent need for an automated solution that can handle the diversity and complexity of the documents in the repository while ensuring accurate classification.

Another significant challenge is the imbalance of labelled examples, with many classes having very few correctly categorised instances (see Figure 1). This imbalance complicates the training of machine learning models, as classes with fewer examples may be underrepresented, leading to biased or inaccurate classification results.

This project aims to address these challenges by introducing an automated pipeline that uses optical character recognition (OCR) and machine learning algorithms for document classification. The proposed solution is designed to distinguish between scanned and digital documents, accurately extract text and use this text to feed classification models. Our approach aims to improve the organisation and accessibility of documents within the Alfresco repository, thus optimising document management at UPV and facilitating the integration of new search and analysis technologies.

The main contributions of this work include:

- Development of an automated document classification pipeline tailored to 51 predefined categories, improving the granularity and accuracy of document organisation.
- Implementing techniques to differentiate between scanned and digital documents, ensuring appropriate handling of different document types.
- Testing of different text vectorisation methods, including traditional term frequency inverse document frequency (TF-IDF) and advanced transformer-based embeddings.
- Comprehensive evaluation of multiple classification models, including BERT and Random Forest, to determine the most effective approach to document categorisation.

2 Background

Optical Character Recognition (OCR) technology has revolutionised the conversion of various types of documents into machine-readable text. Early OCR systems relied on pattern matching and feature recognition [7]. More recent developments have incorporated machine learning techniques to improve accuracy and versatility. One of the most popular OCR engines today is Tesseract, originally developed by Hewlett-Packard and now maintained by Google [24]. Tesseract offers multi-language support and seamless integration with image processing libraries such as OpenCV [5].

PyTesseract, a Python wrapper for Tesseract, has become a widely used tool in the OCR community due to its ease of use and high accuracy. It allows for pre-processing steps such as noise reduction, contrast adjustment and rotation, which are essential for improving the quality of OCR results [18]. Applications of OCR range from digitising printed archives to automating data entry processes and improving text search capabilities within large document repositories [25].

Document classification [4] is a well-researched area in machine learning and natural language processing (NLP). Current methods range from traditional algorithms to deep learning models. Traditional approaches include algorithms such as Naive Bayes [19], Support Vector Machines (SVM) [15], and k-Nearest Neighbours (k-NN) [1]. Other supervised learning algorithms, including decision trees [23], random forests [6], and gradient boosting techniques such as XGBoost [9], rely on labelled data for training. These models have been used extensively in various text classification tasks and have been shown to be effective in different scenarios. They usually rely on textual features extracted using techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) [10].

In contrast, recent advances in NLP have mainly focused on classifying short texts, such as sentences or paragraphs, using Transformer-based models such as BERT [11, 17]. However, these models struggle with long documents due to the computational overhead associated with the self-attention mechanism in standard transformers [27]. To overcome this, various methods have been proposed, such as truncating documents to fit within the token boundary [2, 8] or using alternative architectures such as the longformer, which uses sparse attention to efficiently handle long sequences [3, 29], the latter often showing superior performance [16].

3 Empirical Methodology

The empirical methodology for our document classification system involves several distinct stages, from data acquisition to model evaluation. The whole process attempts to ensure a seamless and efficient document classification pipeline, aimed at handling the diverse and voluminous nature of the documents in the Alfresco repository at UPV.

Data The dataset used in this study was taken from the Alfresco document repository at the Universitat Politècnica de València (UPV). The repository contains approximately 600,000 PDF files, of which only a fraction —approximately 100,000— are correctly categorised into predefined categories. The labelled subset are unevenly distributed across 51 pre-defined categories (see Figure 1). This imbalance posed additional challenges for model training, particularly in ensuring that under-represented categories were accurately classified.

The documents in the repository represent a wide variety of types and content. These include, but are not limited to: *academic records* such as transcripts, certificates and degrees; *administrative records*, including financial records, correspondence and official regulations; *research papers and reports*, consisting of theses, dissertations and project reports; and *miscellaneous documents*, comprising minutes of meetings, memoranda and other miscellaneous forms.

Prepossessing Once the documents were downloaded using the Alfresco API [21], the next task was to determine whether each file was a scanned image or a digitally created document. This distinction is important because the method of text extraction differs significantly between the two types. For scanned documents, we used OCR using *PyTesseract* [22] to convert images into machine-readable text. The OCR process also included the application of enhancements such as contrast adjustment, rotation correction and DPI adjustment using the *OpenCV* library [20] to ensure high quality text extraction.

Once the text was extracted, it underwent a series of preprocessing steps to prepare it for classification. These steps included (1) the removal of special

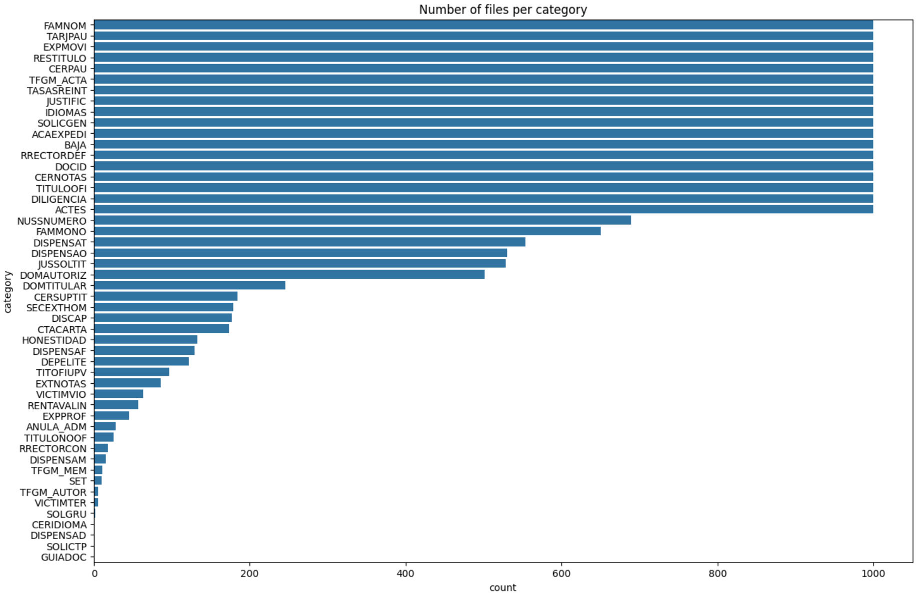

Fig. 1. Class distribution of the labeled documents, highlighting the imbalance across different categories.

characters and numbers that did not contribute to the semantic understanding of the texts were removed; (2) lemmatisation and stemming to reduce words to their root forms, ensuring that different inflections of a word were treated similarly; and (3) the elimination of stop words (e.g., 'and', 'the', etc.) that have no significant meaning were removed. We explored different models for lemmatization, including the SpaCy model *'es_ core_ news_ sm'*[3] and the more advanced *'es_ dep_ news_ trf'*[4] model, which leverages transformer-based techniques to achieve higher accuracy.

After preprocessing, the text data was vectorised. We used two different models for this: the *Term Frequency-Inverse Document Frequency (TF-IDF)*, a traditional text vectorization method that transforms documents into numerical features that reflect the importance of words within the corpus; and the Spacy model *'es_ dep_ news_ trf'*, an advanced model that provides vector representations of the texts using transformer-based techniques. As we show in our comparative analysis, TF-IDF proved to be more robust, providing better performance for this specific classification task compared to the more advanced Spacy model.

[3] An spanish pipeline optimised for CPU that can be found in `https://spacy.io/models/es`

[4] An Spanish transformer pipeline that can be found in `https://huggingface.co/spacy/es_dep_news_trf`

Document classification algorithms We explored several algorithms to find the most effective approach. We started by testing traditional models that have been shown to perform well in the related literature such as *Support Vector Machines* (SVM) [26] and *Random Forest* (RF) [6]. Both models provided a strong basis for classification performance. The *SVM* model was particularly useful in dealing with the complexity of our diverse document set, due to its effectiveness in high-dimensional spaces. However, the RF algorithm showed better performance. Known for its effectiveness with large datasets and its ability to provide robust predictions by averaging the results of multiple decision trees, also reducing overfitting.

To improve classification accuracy, we also used BERT-based language models. The choice of BERT over Longformer [3] for this study is justified by several key considerations: BERT's proven effectiveness and extensive support in the NLP community make it a reliable choice for categorising the document segments typical of this task [2]. In addition, BERT offers computational feasibility and robust performance in capturing contextual embeddings, even for moderately long sequences [12], which is well suited to the needs of the project. Its pre-trained models and ease of integration through libraries such as Huggingface's Transformers [28] facilitate implementation, making BERT a pragmatic and effective choice given the constraints and goals of the study.

Training and evaluation Given the size of the labelled dataset, which consists of approximately 100,000 instances, we decided to use split-validation rather than cross-validation to ensure computational efficiency and faster processing while still maintaining a reliable estimate of model performance. The models were thus trained and evaluated on a labelled subset of the data (80%). We used *precision* (which measures the proportion of true positives among the predicted positives, indicating how many of the predicted categories were correctly identified.), *recall* (which measures the proportion of true positives among the actual positives, reflecting the ability of the model to identify all relevant instances), and *F1-score* (which is the harmonic mean of precision and recall, providing a balance between the two metrics to evaluate the overall performance of the model) as the primary metrics for performance evaluation [13]. The use of a validation set (20% of the labelled subset of the data) ensured the generalisability of the models to unseen data for classification into the 51 predefined categories. The best performing models will be used to categorise the remaining unlabelled documents (500K) in the model deployment phase (which is outside the scope of this work).

BERT training also involved tokenisation (we used *bert-base-uncased*[5]), input sequence truncation and padding, followed by fine-tuning the pre-trained transformer model for our specific task (we used *BertForSequenceClassification*[6]). This computationally intensive process was accelerated using GPU resources.

[5] `https://huggingface.co/google-bert/bert-base-uncased`

[6] `https://huggingface.co/docs/transformers/model_doc/bert`

4 Results

Here we present the performance results and detailed analysis of the classification models used in the automated document classification pipeline. The analysis includes overall model performance metrics and a more granular examination of specific document classes.

4.1 Model performance overview

For our experiments, we focused primarily on traditional vectorisation methods, in particular TF-IDF, due to its superior performance (i.e., better category separation) as observed in our initial evaluations (that will be explained in the following subsections). Figure 2 shows the comparison of precision, recall and F1 scores for the SVM, RF and BERT models on the validation set. The RF model consistently outperformed the others on all metrics —accuracy (0.841), F1 score (0.838), precision (0.847) and recall (0.841). This indicates RF's superior ability to correctly classify documents while minimising false positives and negatives. In contrast, BERT and SVM had comparable but slightly lower performance scores.

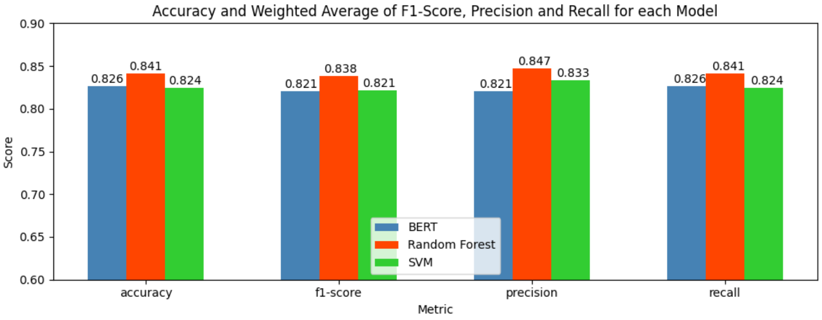

Fig. 2. Comparison of evaluations of the 3 models experimented.

4.2 Class-wise analysis

To provide a more comprehensive understanding, we discuss the performance of the models on specific document classes (i.e., TITOFIUPV, BAJA, ACTES, DOCID, CERNOTAS, TARJPAU, DOMTITULAR, ACAEXPEDI) that are representative of common document types within the dataset, and represent the types of documents in which UPV management is most interested. These classes also serve as benchmarks to illustrate the effectiveness of the model across different levels of content and structural complexity. Table 1 shows the summary of the

Table 1. Class-wise Performance Comparison of Models

Class	Description	Model Performance
TITOFIUPV	Short documents containing the collection record of the official title and the corresponding degree title	BERT showed the highest precision and recall, demonstrating its ability to capture contextual information effectively, even in short documents with limited content.
BAJA	Documents often containing emails and other handcrafted documents with high template similarity.	RF achieved the best recall, accurately identifying this particular class. BERT excelled in precision, using its understanding of context to distinguish these documents from others.
ACTES	Minutes of various events, showing common structural elements.	RF slightly outperformed the other models, reflecting its strength in dealing with structured documents. BERT and SVM also performed well, indicating the general effectiveness of all models for this class.
DOCID	Scanned identification documents with limited text.	BERT significantly outperformed SVM and RF in accuracy, benefiting from its deep understanding of limited textual context.
CERNOTAS	Documents with similar structure but different content, representing different course marks.	All models struggled, especially with accuracy. The diversity of content within similar structures was a challenge, with BERT performing worse than SVM and RF.
TARJPAU	Documents containing university entrance exam grades, often bilingual (Spanish and Valencian).	Similar to CERNOTAS, but with less variety of content. BERT faced challenges due to the bilingual nature, but still performed comparatively well.
DOMTITULAR	Bank account certificates, both scanned and digitised.	BERT achieved higher recall than other models, aided by the presence of contextual expressions that helped to identify this class despite the limited information.
ACAEXPEDI	Diverse documents containing different types of content, often relevant to other categories.	Models showed moderate performance, reflecting the difficulty of classifying documents with mixed or overlapping content.

results for this classes. The F1-scores, precision, and recall for these highlighted classes across the models are shown in Figure 3.

The performance comparison between BERT, RF and SVM over different document classes reveals interesting insights. BERT excels at capturing contextual information for short documents such as those in the TITOFIUPV class, achieving the highest precision and recall. It also achieves remarkable precision for the BAJA class, but is outperformed by RF in terms of recall, highlighting RF's ability to accurately delineate complex classes. For the ACTES class, which is characterised by a common structure but different content, RF slightly outperforms BERT and SVM, highlighting its ability to deal with such scenarios. However, BERT shows a significant improvement in accuracy for the DOCID class, illustrating its superior language understanding capabilities. Challenges for BERT arise with the CERNOTAS and TARJPAU classes. Despite its general strength in context understanding, BERT struggles with these classes due to their similar structures and varied content, as well as the bilingual challenges in TARJPAU. In the DOMTITULAR class, BERT's higher recall indicates his ability to grasp the context effectively, even with limited information. Conversely, the diverse nature of the ACAEXPEDI class is a challenge for all models, but RF and SVM show better performance in distinguishing document structure. BERT's difficulty here highlights its limitations with contextual distortion when faced with content diversity.

Overall, RF proved to be particularly effective in scenarios with common document structures or where high recall is crucial. BERT showed its strength in accuracy and context understanding, especially for documents with limited information. However, BERT struggled with diverse content or bilingual scenarios, highlighting its limitations in certain classification contexts. Therefore, for the task of automated document classification that we address, the RF model is recommended due to its higher overall reliability and accuracy.

4.3 Visualisation of vectorisation

To further assess the effectiveness of our text vectorisation methods, we visualised document embeddings with t-SNE (t-distributed Stochastic Neighbour Embedding) [14] (which offers a visual representation of the high-dimensional data in two dimensions). These visualisations illustrate how well different document categories are separated in the vector space, providing insight into the clustering quality of our vectorisation techniques.

In this regard, Figure 4 (top) shows the t-SNE visualisation of document embeddings using the SpaCy model *'es_ dep_ news_ trf'*. Despite the use of advanced transformer-based techniques, the SpaCy model showed some category overlap, indicating room for improved differentiation. Conversely, Figure 4 (bottom) shows the t-SNE visualisation for the TF-IDF method. This traditional vectorisation technique showed better category separation, consistent with its higher classification performance, suggesting that the concrete problem and specific dataset of this study benefited from this simpler yet effective approach.

Overall, TF-IDF proved to be a more effective vectorisation method for this classification task compared to the advanced SpaCy model. By optimising the vectorisation approach, we improved the overall performance and robustness of our classification models, in particular the RF model, which handled the document diversity in the dataset exceptionally well. We have seen that, in general, a careful selection of both the classification model and the vectorisation technique is crucial for achieving optimal results in automated document classification tasks.

4.4 Interpretation of results

The results of this study highlight the effectiveness of our automated classification pipeline in addressing the challenges faced by the UPV in managing its extensive document repository. Among the models evaluated, RF demonstrated superior performance in accurately classifying documents into predefined categories, in line with our goal of improving document organisation and accessibility.

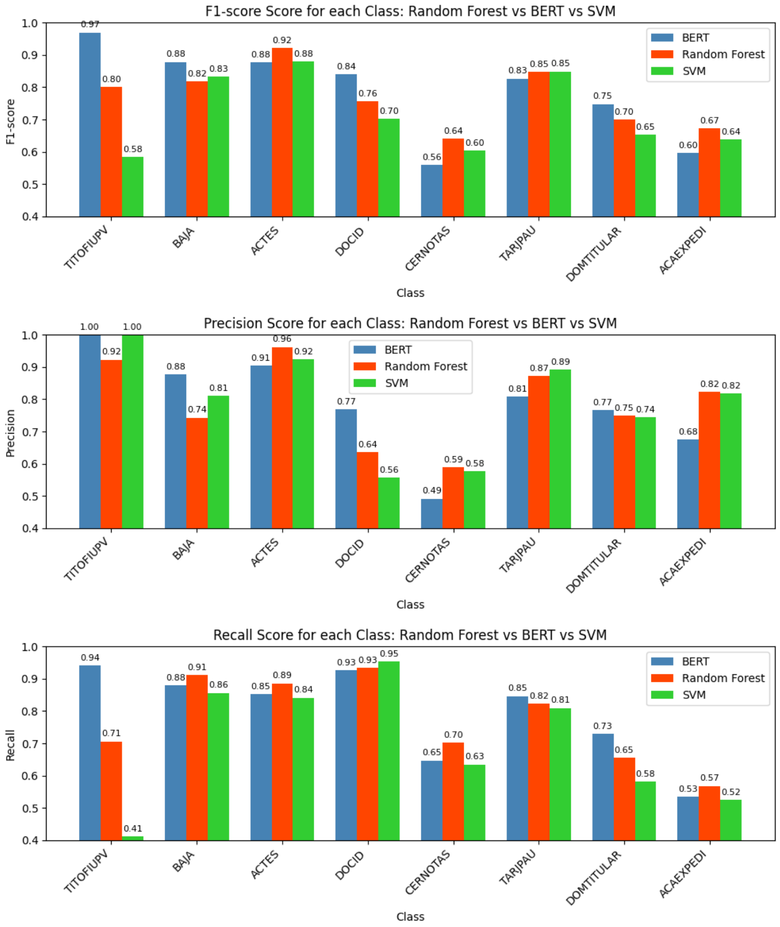

Fig. 3. Performance metrics for highlighted classes across different models.

The RF model's ability to handle the complex task of multi-class classification was particularly noteworthy, as it skilfully used TF-IDF vectorisation to manage the semantic inconsistencies inherent in the dataset due to OCR inaccuracies. While BERT's deep learning capabilities allowed it to capture the nuanced complexities in the text, its performance was not as consistently robust across the diverse document set as RF.

Despite the promising results, there were several limitations to this study. Firstly, the variety of document types, ranging from highly structured forms to unstructured text, posed challenges that could not be fully addressed by a single approach. The quality of the documents and the presence of multiple languages further complicated the OCR process, adversely affecting the accuracy of text extraction and consequently the classification performance. Inconsistent

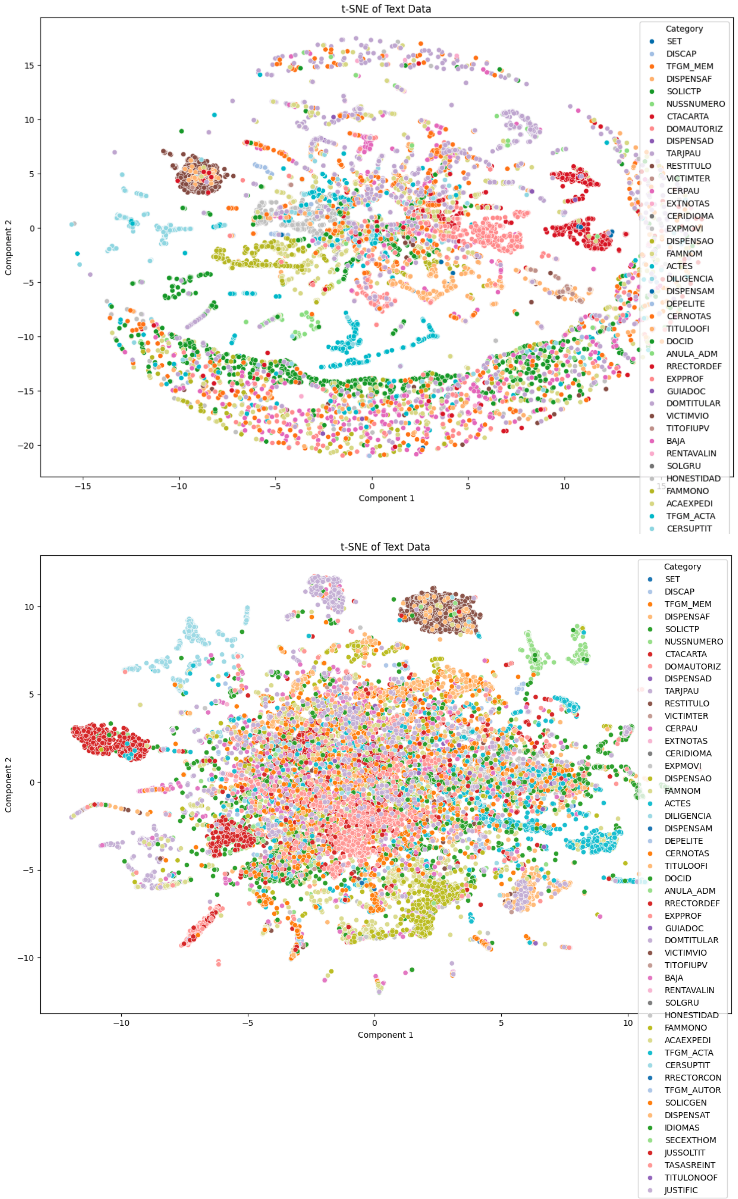

Fig. 4. t-SNE visualization of the embeddings, highlighting the similarity across different categories using TF-IDF (top) and the SpaCy model *'es_ dep_ news_ trf'* (bottom).

OCR text extraction, particularly in documents with complex layouts or noisy backgrounds, often led to misclassifications.

5 Conclusions and Future Work

This study demonstrates the effectiveness of our automated classification pipeline in managing the large and diverse document repository at the UPV. RF outperformed BERT and SVM in all evaluated metrics (accuracy, F1 score, precision and recall) in most of the classes and overall, making it particularly suitable for the diverse document types present in UPV's Alfresco repository. While BERT showed strength in handling contextually rich documents, it was less effective across broader categories compared to RF. In addition, the traditional TF-IDF vectorisation method provided better category separation, which significantly improved the overall performance of the model. The ability of RF to effectively use TF-IDF embeddings demonstrated its robustness in dealing with data inconsistencies due to OCR inaccuracies.

Future work should focus on several specific actions to build on the current study. First, an in-depth analysis of document classes using unsupervised machine learning techniques, such as clustering, will help to identify natural groupings within the data and potentially reveal new, meaningful categories. Secondly, implementing a hierarchical cascade model can break down the classification task into smaller, more manageable components, which can improve overall classification accuracy. Finally, increasing the volume of accurately labelled samples will create a more balanced and comprehensive training dataset, reducing bias towards underrepresented classes and improving the model's ability to generalise.

By addressing these areas, future iterations of the document classification pipeline can achieve greater accuracy, reliability and efficiency, significantly enhancing the performance of UPV's document management system.

References

1. Aas, K., Eikvil, L.: Text categorisation: A survey (1999)
2. Adhikari, A., Ram, A., Tang, R., Lin, J.: Docbert: Bert for document classification. arXiv:1904.08398 (2019)
3. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv:2004.05150 (2020)
4. Borko, H., Bernick, M.: Automatic document classification. Journal of the ACM (JACM) **10**(2), 151–162 (1963)
5. Bradski, G.: The opencv library. Dr. Dobb's Journal: Software Tools for the Professional Programmer **25**(11), 120–123 (2000)
6. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
7. Casey, R.G., Lecolinet, E.: A survey of methods and strategies in character segmentation. IEEE trans. on pattern analysis and machine intelligence **18**(7), 690–706 (1996)
8. Chalkidis, I., Fergadiotis, M., Kotitsas, S., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: An empirical study on large-scale multi-label text classification including few and zero-shot labels. arXiv:2010.01653 (2020)
9. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd int. conf. on knowledge discovery and data mining. pp. 785–794 (2016)

10. Christian, H., Agus, M.P., Suhartono, D.: Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). ComTech: Computer, Mathematics and Engineering Applications **7**(4), 285–294 (2016)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
12. Ding, M., Zhou, C., Yang, H., Tang, J.: Cogltx: Applying bert to long texts. Advances in Neural Information Processing Systems **33**, 12792–12804 (2020)
13. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern recognition letters **30**(1), 27–38 (2009)
14. Hinton, G.E., Roweis, S.: Stochastic neighbor embedding. Advances in neural information processing systems **15** (2002)
15. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European conf. on machine learning. pp. 137–142. Springer (1998)
16. Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**(1), 1–9 (2016)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 (2019)
18. Malathi, T., Selvamuthukumaran, D., Chandar, C.D., Niranjan, V., Swashthika, A.: An experimental performance analysis on robotics process automation (rpa) with open source ocr engines: Microsoft ocr and google tesseract ocr. In: IOP conf. Series: Materials Science and Engineering. vol. 1059, p. 012004 (2021)
19. McCallum, A., Nigam, K., et al.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. vol. 752, pp. 41–48. Madison, WI (1998)
20. OpenCV Contributors: OpenCV Documentation. `https://docs.opencv.org/4.x/`, accessed: 2024-07-17
21. Potts, J.: Alfresco developer guide. Packt Publishing Ltd (2008)
22. PyTesseract Contributors: Pytesseract: A python wrapper for google's tesseract-ocr engine (2024), `https://pypi.org/project/pytesseract/`, accessed: 2024-07-08
23. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1986)
24. Smith, R.: An overview of the tesseract ocr engine. In: Ninth int. conf. on document analysis and recognition (ICDAR 2007). vol. 2, pp. 629–633. IEEE (2007)
25. Srivastava, S., Verma, A., Sharma, S.: Optical character recognition techniques: A review. In: 2022 IEEE int. Students' conf. on Electrical, Electronics and Computer Science (SCEECS). pp. 1–6. IEEE (2022)
26. Steinwart, I., Christmann, A.: Support vector machines. Springer Science & Business Media (2008)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
28. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Huggingface's transformers: State-of-the-art natural language processing. arXiv:1910.03771 (2019)
29. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al.: Big bird: Transformers for longer sequences. Advances in neural information processing systems **33**, 17283–17297 (2020)