DOCUMENT FORM UNDERSTANDING USING MULTI-MODAL DEEP LEARNING APPROACH

Final Report

Information Technology Capstone Project

COMP5703

Group Members

1. Kaer Li (510554284) [Group Leader]
2. Xing Xing (500390560)
3. Shuai Liu (510300441)
4. Yuxi Shen (490548486)
5. Yuanyuan Lin (500595497)
6. Wenjie Jiang (510250146)

CONTRIBUTION STATEMENT

Our group, taking project CS62-1, with group members Kaer Li, Xing Xing, Shuai Liu,Wenjie Jiang, Yuanyuan Lin and Yuxi Shen, would like to state the contributions each group member has made for this project during semester 1 2022:

- [Kaer Li]: FUNSD literature review and summary; Bert literature review; code implementation in OCR text extraction, code implementation of manual annotation. Manual annotation of 100 documents (re-annotation included); fine-tune of pre-trained models (including Bert only and Visual features), fine-tune of Bert only and Bert large models on manually-annotated dataset ,visual feature extraction implementation, weekly document management, connection between clients and project team and project management, monitoring of project progress.

- [Xing Xing]: Implement visualisations on manual annotated dataset. Generate temporary datasets for visualisation and analysis. Entity recognition code implementation. Initial pdf destruction and png creation , bounding box extraction, OCR text extraction, code implementation of manual annotation, fine-tune models on manually-annotated dataset, visual feature extraction implementation, final dataset assembling and csv creation, final deliverables packing and report documentation. The entire resources section, formatting, review and decoration.

- [Shuai Liu]: Initial pdf destruction and png creation, bounding box extraction, OCR text extraction, code implementation of manual annotation, re-annotation, bounding box labeling, visual feature extraction implementation, fine-tune of pre-trained model, fine-tune of Bert only and Bert large models on manually-annotated dataset, final dataset assembling and csv creation. Final deliverables packing and report documentation. The entire resources section, formatting, review and decoration.

- [Wenjie Jiang]: FUNSD paper review and summary, bert paper review. Background information review. Participated in bounding box extraction by PDFminer and Google OCR API, code implementation of manual annotation, re-annotation, model initial tuning, document management, statistical analysis, visual analysis, visual poster. weekly deliverable management, group meeting notes recording. The entire project problem section and data analysis section of the report.

- [Yuanyuan Lin]: literature study and summary. Manage, record, and update the weekly deliverables and papers for the group. Participated in PDFminer and Google OCR API bounding box extraction Using a pretrained Bert-base model, train and fine-tune the label classification model. Participated in visual feature extraction, dataset visual analysis, and created a visual poster. Final report, abstract, introduction, and part on literature evaluation.

- [Yuxi Shen]: Research on LayoutLM v2. Code implementation of manual annotation. Manual annotated 100 documents. Project management and task assignment during dataset generation phase. Implementation of the “table” label generation. The label classification model training and fine-tuning with pretrained Bert-base model, and both with visual features and without visual features. Participated in OCR text extraction. The entire methodologies section in final report except for the data analysis subsection.

TABLE OF CONTENTS

All group members agreed on the contributions listed on this statement by each group member.

Signatures:

Kaer Li:		Shuai Liu:	Shuai Liu
WenJie Jiang:		Yuxi Shen:	
Xing Xing:	Xing Xing	Yuanyuan Lin:	

Table of Contents

Abstract

From are generally recognised to be one of the most prevalent ways of data collecting. However, most IT researchers find form understanding difficult. Because available datasets consistently fail to match the requirements of deep learning mechanisms. As a result, the goal of our study is to construct a large-scale forms comprehension dataset based on over 7000 genuine multi-page financial form documents. This dataset may design and manage numerous form comprehension tasks explicitly. This project's implementation was separated into three stages: first, initial dataset creation, The dataset was created mostly using PDFminer and Optical Character Recognition (OCR). Text was extracted from forms using PDFminer. OCR is a cloud deep learning API that is used to do text recognition and character recognition. The dataset is evaluated and analysed in the second step. The measures implemented for analysis and testing included layout analysis, entity recognition, and named entities. The resulting dataset may be used to train a model for future form comprehension problems. Finally, we created an interactive software to examine our dataset in order to test our results visually.

Table of Contents

1. Introduction

Forms are one of the most prevalent techniques of data collection. Forms are frequently utilised in a variety of businesses, particularly the financial sector, due to their ability to include rich visual and textual information. However, due to the irregularity of the forms and the diversity of information contained in them, extracting information is quite challenging. Because of the information stored within them, it is quite difficult to extract information from them properly. There are currently several tables with enormous quantities of legitimate data. It would be extremely beneficial in many industries, such as finance and healthcare, if it could be simply retrieved from the relevant data. Many fields would benefit greatly if the necessary data could be quickly pulled from a large number of tables. Existing datasets make it difficult to extract information and do in-depth research. Most information technology researchers struggle with document comprehension. Form understanding research often focuses on information extraction tasks. Attempts to turn information in document pictures into machine-readable form using techniques such as character recognition, table extraction, and key-value pair extraction. As a result, the goal of this research is to provide a large-scale dataset for tabular understanding in the financial area. Finance uses large-scale datasets. This dataset may be used to train formal understanding models. This dataset may be used to train form understanding models and will handle several critical form understanding tasks, such as assessing layouts and effectively collecting information from forms. This involves assessing layouts, collecting information from tables effectively, and correctly anticipating entity connections.

2. Related Literature

2.1 Literature Review

This project will examine six pieces of literature about form information extraction, form layout analysis, entity relationship prediction, manual annotation, data analysis, and associated model training. All of the literature is up to date and extremely relevant to the entire process of our project. This is a crucial realization for our work. Information extraction necessitates a rapid yet thorough examination of the entire material. Not only must we search for information snippets, but we must also construct the final output for a given entity type, such as aggregating several occurrences of an organization's name into a single entity (Graliński al., 2020). This implies that the results must be presented in a suitable format (for example, data points such as addresses are normalized to a standard form). It should also explain why specific pieces of information have been linked (Graliński al., 2020). An indicator in the input text can do this. For humans, this procedure may be laborious and challenging. As a result, we require automated systems to process various documents and extract the necessary information in a timely manner. However, the gap between what is conceivable with current information extraction technology and what is necessary for real-world commercial use cases remains large. (Graliński al., 2020). Systems that automatically gather personal information, from a business user's perspective, automatically collect information on individuals, their roles, crucial dates, addresses, and sums (Graliński al., 2020). These systems must be trustworthy and able to judge their certainty about the things from which they are derived. However, given today's technology, many machine learning models must be taught to be resilient against named entities. To enhance training efficiency, even more, we may leverage the layout of previously described documents to teach the model how to extract specific information (Graliński al., 2020). On the other hand, there is still a need for more generic extractors that must cope with a wide range of data.

The form is a sort of universal document format that is used to collect data. It is applicable in a variety of fields, including healthcare, academia, and finance (Wang, Zhan, Liu, & Liang, 2020). However, because it comprises a vast number of written and visual information, extracting relevant information from forms in numerous formats or across sectors is challenging. To comprehend these papers, geometric layout analysis methodologies based on an image representation of the document, as well as Optical Character Recognition (OCR) methods, were initially used (Zhong, Tang &Yepes, 2019). Although OCR works well for form understanding, it has limits owing to form layout variations. For example, many manual forms in previous work do not apply to all papers, therefore the structure of its forms is required (Wang, Zhan, Liu, & Liang, 2020). Extracting information from a document requires not only converting the text of the document into a machine-readable format, but also doing layout analysis (Xuet al., 2022).

Hankiewicz (2020) point out that technologies such as natural language processing (NLP) and machine learning (ML) are suited for intelligent document analysis and comprehension. They aid in extracting insights from unstructured data such as written documents, emails, and photos. There are now numerous standard NLP development methodologies that, when coupled, may be utilised for intelligent document interpretation and analysis (Hankiewicz, 2020). Named entity recognition, text classification, sentiment analysis, text similarity, information extraction, and connection extraction are such techniques. Named Entity Recognition (NER) is one of them (Hankiewicz, 2020). It recognises named entities in unstructured data, such as text documents, and categorises them, such as person names, organisations, places, time expressions, percentages, medical codes, amounts, etc. Statistical NER systems often need a considerable amount of manually annotated training data (Hankiewicz, 2020). To prevent incomplete annotation, semi-supervised techniques are commonly used (Hankiewicz, 2020). Text classification is giving labels or categories to text based on its content. It is also known as text labelling or text categorisation. Text may be automatically examined and assigned a set of predetermined labels or categories depending on its content using an NLP text classifier. It has utilised sentiment analysis, topic labelling, spam identification, and intent detection, among other things (Hankiewicz, 2020). Text categorisation will predict categories based on a document's words, entities, and sentences. It frequently flows into other aspects of the decision, such as any titles, information, or pictures in the text (Hankiewicz, 2020).

The identification of the layout of unstructured digital data is a critical step in parsing the files into a structured machine-readable format for downstream applications. Deep neural networks, which were created by computer vision, have shown to be an excellent way for analysing the arrangement of document pictures (Xu et al., 2022). Currently current publicly available layout datasets are several orders of magnitude less than known computational vision datasets (Xu et al., 2022). Transfer learning must be used to train the model, which is built on a base model that has been pre-trained on traditional computer vision. Manual annotation is used in current datasets for document layout analysis. Some of these datasets have been used in document processing competitions (Xu et al., 2022). But when the dataset is a large number of documents by human annotation is time-consuming and error-prone. Text and layout pre-training has been shown to be successful in a variety of visually rich document comprehension tasks, according to Xu et al. (2022), and the LayoutLMv2 pre-training task model is proposed, which is used for a multimodal pre-training strategy for visually rich document comprehension tasks. A multi-modal transformer structure is developed that accepts textual, visual, and layout data as input and produces substantial cross-modal interactions (Xu et al., 2022). In addition, the LayoutLMv2 model's performance is evaluated using six publicly available benchmark datasets. The experiments show that LayoutLMv2 achieves significant results not just on the regular VrDU task, but also on the document image VQA task (Xu, et al, 2022).

Many natural language processing tasks have been proven to benefit from language model pre-training. BERT is a technique for pre-training linguistic representations from unlabeled material that employs transformers (Devlin et al., 2019). By simply adding output layers, these pre-trained models may subsequently be utilised for downstream tasks. This is an output layer used in extractive question answering to forecast the starting and end indices of the answer span (Devlin et al., 2019). Thus, by simply adding an output layer, pre-trained BERT models may be fine-tuned to generate state-of-the-art models for a wide range of tasks such as question answering and linguistic inference using state-of-the-art models language thinking without needing significant changes to the task-specific architecture (Devlin et al., 2019). BERT is both theoretically and experimentally simple. It got new cutting-edge results on 11 natural language processing tasks (Devlin et al., 2019).

3. Research/Project Problem

3.1 Research/Project Aims & Objectives

This project aims to create a large-scale dataset to achieve the goal of defining and solving multiple form understanding tasks. The dataset is created based on a large amount of form-format files in the financial domain. Through the feature extraction, manual annotation, model training and layout and data analysis the dataset will meet the demands of deep learning mechanisms.

The dataset differs from the successful FUNSD dataset in the format of multi-page and the focused financial domain (Jaume et al., 2019). The form understanding challenges which include extracting information, analysing layout and predicting entity relation can be solved by the dataset explicitly. The sequential procedures of form understanding can be possibly defined through the dataset. And it could be used to train models for form understanding tasks in the future. The final deliverable of the dataset should include an interaction function to evaluate the performance of the dataset.

3.2 Research/Project Questions

Jaume et al. (2019) state that extraction of required data and information from forms is an arduous task for most IT researchers, although forms contain abundant valuable information. The origin dataset is huge, the biggest document containing over two thousand pages led to the extraction of information from the document becoming a tough challenge. However, a small-scale dataset does not meet the demand of deep learning mechanisms. The performance of the Optical character recognition platform is another problem, OCR is commonly used to process of the analysing and recognizing image files of text data to obtain text and layout information. However, it has limitations when processing the low-resolution file.

3.3 Research/Project Scope

Project scope statement

This project involves creating a large-scale dataset based on a large amount of real-life muti-pages financial form documents. There will be an interaction system for evaluating the performance of the dataset and testing the accuracy of the bounding box extraction. The project will define and solve the form understanding tasks such as, analysing layout, extracting information and predicting entity relation. And can be used for training models for the form understanding tasks in the future.

Project scope

In scope:

1. The document form understanding dataset: The large-scale dataset based on over seven thousand of real-life multi-pages form documents on financial industry.

2. Manual annotation dataset: The manual annotation for comparing the dataset with the ground truth and test the performance of the bounding box extraction. The cross-validation will be used for evaluating the accuracy of manual annotation.

3. Model training: The dataset will be trained on a pre-trained model, the accuracy of the model training should be over 80 percent.

4. Visual analysis: Visual analysis will be performed for intuitive and accurate presentation of dataset. Several high-level analysis and low-level analysis will be present and a poster will be used to show all the visual analysis.

5. Interaction system: The interaction system is used to present the status of the dataset and evaluate the performance of the dataset.

Out of scope

1. Bounding box extraction for handwriting contents: Although handwritten contents are common in real-life, especially dates and signatures, we omit this task since it is a separate task that requires specific handling. We expect to overcome the handwriting recognition difficulties in the future.

2. Information extraction for low-resolution forms: The information extraction of low-resolution forms is very difficult due to low-resolution forms are difficult to be recognized by OCR.

3. Website system: Due to the lack of experience on website frontend and backend programming, we decide to use data analysis to showcase the dataset instead of the the website system.

Project deliverables

- Large-scale dataset: the large-scale dataset created based on over 7000 real-life multi-page financial PDF documents. The dataset will be a CSV file which contains four columns (serial number, text, label, and visual feature) and 86248 observations.

- Manually-annotated dataset: The manually-annotated dataset is used to compare the bounding extraction result with the ground-truth and do model training on the pre-trained model. The accuracy of the model training is over 80% on both bert-based model and bert-large-based model.
- Visual analysis result: The visual analysis will be present by 2 high-level analysis tables and 6 low-level analysis tables. All of the visual analysis will be presented as a poster.
- Interaction system: the interaction system is used to present the accuracy of the dataset to the users.
- Used codes: The programming coding used during the project will be shown.
- Group presentation: the presentation video and slides. The presentation includes introduction, motivation, methodologies, result, evaluation, discussion, reflection and conclusion.

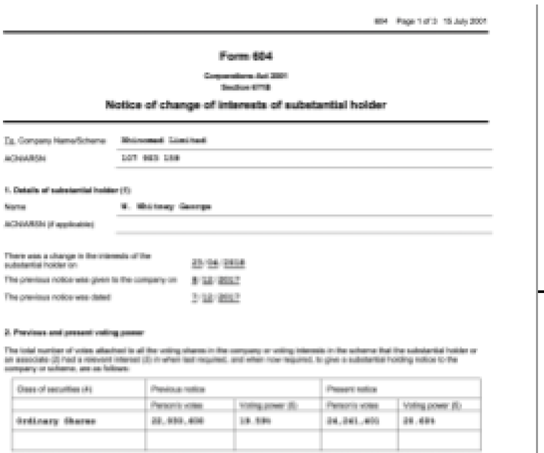
Project acceptance criteria

The project should create a dataset that can meet the demands of applying novel deep learning mechanisms and be able to define and solve several form understanding tasks. The model training of the dataset should perform an outstanding result. The visual analysis should evaluate the dataset and show the performance for users. The interaction system should meet the requirements of bounding box extraction and annotation accuracy evaluation. All of the project deliverables should be finalised before 12 Jun 2022.

4. Methodologies

4.1 Data Collection

The source data in this project is provided as a whole by the project client. It consists of 7130 documents in PDF format. Each document is multi-page, with page numbers ranging from 2 to up to thousands. The contents in the documents are homogeneous: they are Form 604 (Notice of change of interests of substantial holder), see Figure 1. It is a form given to a specific company regarding changes of substantial holders' interests. It has sectioned common structures listing all related information. Like in Figure 1, all information are arranged in headers, sections, paragraphs, tables, question and answer pairs.



604 Page 1 of 3 15 July 2001

Form 604

Corporations Act 2001
Section 671B

Notice of change of interests of substantial holder

To Company Name/Scheme [illegible] Limited

ACN/ARSN 107 863 158

1. Details of substantial holder (1)

Name W. Whitney George

ACN/ARSN (if applicable)

There was a change in the interests of the substantial holder on 23/04/2018

The previous notice was given to the company on 8/12/2017

The previous notice was dated 7/12/2017

2. Previous and present voting power

The total number of votes attached to all the voting shares in the company or voting interests in the scheme that the substantial holder or an associate (2) had a relevant interest (3) in when last required, and when now required, to give a substantial holding notice to the company or scheme, are as follows:

Class of securities (4)	Previous notice		Present notice	
	Person's votes	Voting power (5)	Person's votes	Voting power (5)
Ordinary Shares	22,930,400	19.59%	24,241,401	20.69%

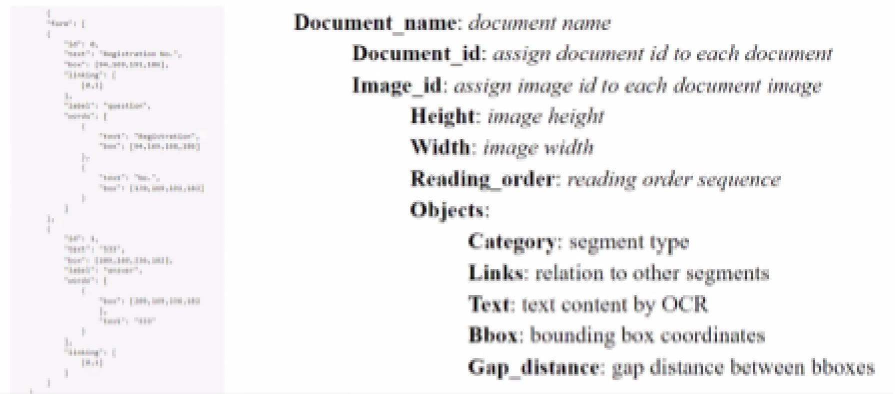
4.2 Methods

4.2.1 Convert PDF to PNG by PyMuPDF

As our source dataset is a group of PDF documents, the first step is to convert them into images with PNG format for ease of handling. Thanks to the advances in computer vision, PDF documents as images can be processed by more available and efficient tools than its original format. We use a python package PyMuPDF to get the pixmap from a PDF page. Pixmap is an object class in the PyMuPDF package. It contains pixel based RGB information of an image, with a variety of useful methods. But we directly save the pixmap object as an image in PNG format. Now, each multi-page source PDF document is converted into images, with each page as a single image. The image file name contains information about document ID and page number ID.

4.2.2 Extract Bounding Boxes by PDFMiner

PDFMiner provides a whole ecosystem for processing PDF documents. It can read in PDF documents and construct required objects like PDFDocument and PDFPage. Then we use PDFPageInterpreter to extract token level bounding boxes from the documents. Note that this step is using the raw documents in the PDF format. And we subsequently project the bounding box coordinates to the documents in image format. At this step, we created a JSON file to store our bounding boxes. The JSON file has the format as requested by the project client as below: the top level entity is documents, with document name and ID; The next level entity is the document pages as images, with image ID and image metadata; Finally, text line objects in each page will have various fields. At the current stage, only the bounding boxes field has valid information from PDFMiner. All other fields are left blank or with placeholders.



```
{
"form": [
{
    "id": 0,
    "text": "Registration No.",
    "box": [illegible],
    "linking": [
        [0,1]
    ],
    "label": "question",
    "words": [
        {
            "text": "Registration",
            "box": [illegible]
        },
        {
            "text": "No.",
            "box": [illegible]
        }
    ]
},
{
    "id": 1,
    "text": "[illegible]",
    "box": [illegible],
    "label": "answer",
    "words": [
        {
            "box": [illegible]
            ],
            "text": "[illegible]"
        }
    ],
    "linking": [
        [0,1]
    ]
}
```

Document_name: *document name*
Document_id: *assign document id to each document*
Image_id: *assign image id to each document image*
Height: *image height*
Width: *image width*
Reading_order: *reading order sequence*
Objects:
Category: segment type
Links: relation to other segments
Text: text content by OCR
Bbox: bounding box coordinates
Gap_distance: gap distance between bboxes

Figure 2: the JSON format of documents

4.2.3 Extract Texts by Google OCR tool

With the JSON file ready and token line objects identified, we need to find the texts in the bounding boxes. This is the most important information and its quality will affect the quality of the final dataset. Taking the quality factor into consideration, we have chosen to use Google's OCR (optical character recognition) service. Google is well-known for its advances in computer vision and large scale pretrained models. OCR, as a deep learning technique, will benefit the most from the large amount of training data and Google has the capacity of the best trained OCR tool. Therefore, we choose to use OCR provided Google Cloud as Google Cloud API. Once an API key is generated as requested, it can be used in python notebooks to connect with the Google Cloud. Next we design the paths for OCR to assess the PNG data and store the results, and OCR can accurately recognise the texts in each bounding box. This process is time-consuming as our image data is information rich, with up to hundreds of token line bounding boxes in each page. And our dataset is also large-scale with thousands of documents and up to thousand pages per document. This is one of the major steps in our project and we have spent many accounts' free quota and an additional $1200. Finally, we have obtained the quality OCR results and stored the extracted texts in the JSON files.

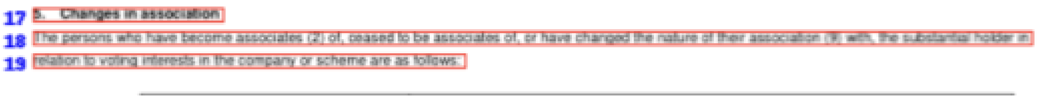
4.2.4 Human Annotation

To enable high-quality downstream tasks' training, we obtain ground truth information by human annotation. Since downstream tasks are most likely deep learning based, it benefits the most by having error-free ground truth labels. Human annotation is the most reliable way to generate high accuracy annotations. And the common structures in Form 604 allows for a more consistent human annotation

Inspired by FUNSD (Jaume et al., 2019), we design a reasonable and comprehensive annotation guideline according to the possible dataset usage and Form 604's structure. During human annotation, a full document has titles and sections. In each section, there is section title and section content.

To account for the errors in PDFMiner, i.e. mostly bounding box localisation errors, we also manually record the bounding boxes that need to be merged or splitted. These two scenarios are shown in Figure 3(a) and 3(b). The main reason behind this is that a text line bounding box should faithfully contain the whole text line and no other text line information. Apart from this, we also ask the annotator to record every single problem presented in the document. After going through hundreds of documents, we categorised all possible problems in the following: number of missing boxes, too small bounding box and empty boxes. These cases are presented in Figure 3(c)-3(e) with the same theory as before. For the missing bounding box case, we record the number of missing bounding boxes in the whole page and we will not manually add the actual bounding box information, as this procedure is troublesome and missing boxes are relatively minor issues. Note, some boxes can contain multiple issues at the same time, see Figure 3(c), where the one box is too small (does not cover the full text line) and too big (covers other textline information) at the same time.

All human annotation results are stored in a single excel format. It is easy to convert into other formats like CSV.

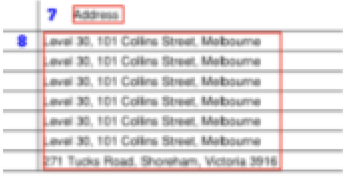


5. Changes in association

The persons who have become associates (2) of, ceased to be associates of, or have changed the nature of their association (9) with, the substantial holder in

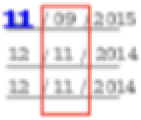
relation to voting interests in the company or scheme are as follows:

Figure 3(a) Merge: Bounding box No.18 and 19 should be merged as they belong to the same text line



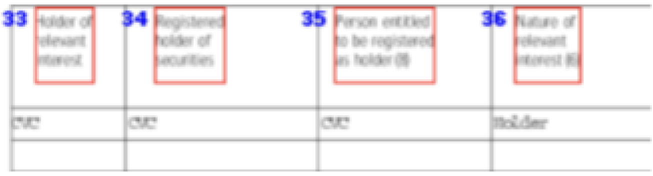
Address
Level 30, 101 Collins Street, Melbourne
Level 30, 101 Collins Street, Melbourne
Level 30, 101 Collins Street, Melbourne
Level 30, 101 Collins Street, Melbourne
Level 30, 101 Collins Street, Melbourne
Level 30, 101 Collins Street, Melbourne
271 Tucks Road, Shoreham, Victoria 3916

Figure 3(b) Split: Bounding box No.8 should be split into 7 different bounding boxes as they belong to different cells in a table.



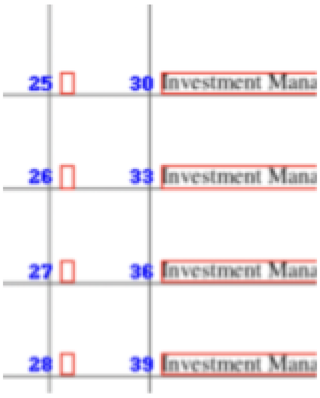
/ 09 / 2015
12 / 11 / 2014
12 / 11 / 2014

Figure 3(c) Too small box : Bounding box No.11 is too small as it does not cover the whole text line. However, it also needs to be split. Hence, two problems present in this one sample.



Holder of relevant interest	Registered holder of securities	Person entitled to be registered as holder (8)	Nature of relevant interest (6)
CVC	CVC	CVC	Holder

Figure 3(d) Missing boxes: There are four missing boxes present. The corresponding texts should be "CVC", "CVC", "CVC", "Holder".



Investment Mana
Investment Mana
Investment Mana
Investment Mana

Figure 3(e) Empty boxes: Bounding box No.25, 26, 27 and 28 are boxes with no content. They shouldn't be recognised and are recorded as empty boxes.

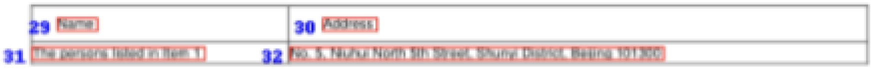
4.2.5 Reannotation

We take two measures to alleviate the human error problem. The first one is that we officially write down the annotation guideline and go through some sample documents with all the annotators. This can align all annotators' understanding and allow a more consistent annotation. The second measure is that we reannotate 100 documents. By reannotation, we mean that the documents are already annotated by a first annotator, and a second annotator should reannotate the same documents according to the same annotation rules. This can both reassure the annotators’ understanding of the guidelines and act as a testing tool. It can identify potential human deviations. Luckily, by manually comparing the two annotations, we found that our annotators do have the same understanding of the annotation guidelines and they show consistent and reliable behaviours.

4.2.6 Label Classification Model

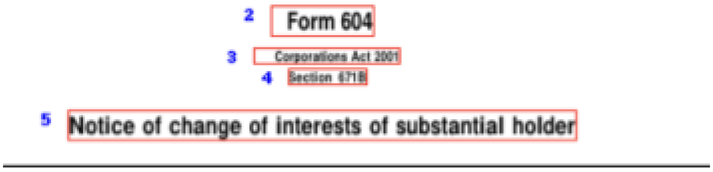
After human annotation, our main dataset is complete. However, to showcase a downstream task that can effectively and efficiently utilise our dataset, we train and fine-tune a label classification model.

4.2.7 Label design



29 Name	30 Address
31 The persons listed in Item 1	32 No. 5, Niuhui North 5th Street, Shunyi District, Beijing 101300

Figure 4(a): “table” label: All bounding boxes in this sample should have the label of “table” as they are all in table cells.



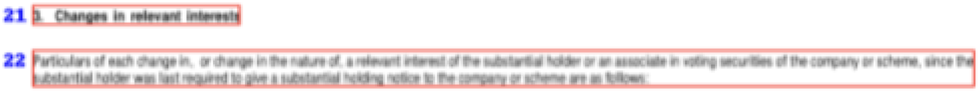
2 Form 604

3 Corporations Act 2001

4 Section 671B

5 Notice of change of interests of substantial holder

Figure 4(b): “title” and “text” label: Bounding box No.2, No.5 have “title” label as they are identified as titles according to human annotation. No.3 and 4 have “text” label as they do not belong in any other labels.



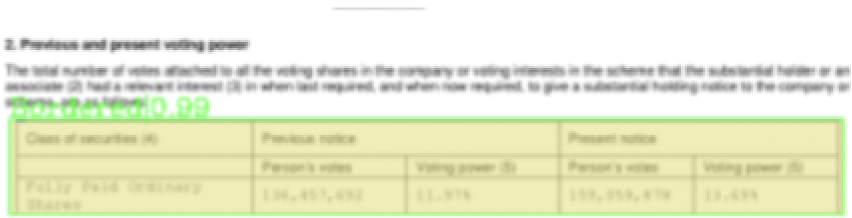
21 3. Changes in relevant interests

22 Particulars of each change in, or change in the nature of, a relevant interest of the substantial holder or an associate in voting securities of the company or scheme, since the substantial holder was last required to give a substantial holding notice to the company or scheme are as follows:

Figure 4(c): “section” and “text” label: Bounding box No.21 has “section” label according to human annotation. No.22 has “text” labels as it is the paragraph body.

We design each bounding box as a single sample, with labels as ground truth. The labels are: table, title, section, text. Any textlines that are inside a table will have the “table” label, see Figure 4(a), and this will be explained in detail below. “title” and “section” label will be given out as per the human annotation results. This ensures the reliability of labels, with credits to the quality-assured human annotations. The “text” label means other contents that do not belong to tables and are not section header or titles. This label can be given out as soon as other three labels are assigned. The well-designed four label design covers the common patterns of our Form 604 PDF documents and it can be useful in other Document Form Understanding tasks.

4.2.8 The “table” Label Generation



2. Previous and present voting power

The total number of votes attached to all the voting shares in the company or voting interests in the scheme that the substantial holder or an associate (2) had a relevant interest (3) in when last required, and when now required, to give a substantial holding notice to the company or scheme, are as follows:

Class of securities (4)	Previous notice		Present notice	
	Person's votes	Voting power (5)	Person's votes	Voting power (5)
Fully Paid Ordinary Shares	136,457,692	11.97%	159,059,478	13.69%

Figure 5: CascadeTabNet generate high-confidence and high-accuracy table bounding boxes that just covers the full table. The green number “0.99” is the prediction confidence.

As of this stage, we have the location information about the bounding boxes. Hence, if we know the outline location about the tables, we can identify which boxes are inside the table. We use CascadeTabNet with official trained model checkpoint to predict the table outline bounding box. Figure 5 is a sample result of CascadeTabNet. It precisely outlines the table as a big bounding boxes and gives a confidence score at the same time. We only consider the boxes with confidence score over 0.8 to ensure quality.

Next we compare the IOU(intersection over union) of all of our bounding boxes from PDF files and the extracted table bounding boxes from CascadeTabNet. IOU is a well-known metrics in computer vision that measures the percentage of two bounding boxes’ intersection area in the two bounding boxes’ whole area. It is useful in our task but it overlooks one scenario: when one bounding box is significantly larger than the other box and the former completely covers the latter. In this situation, the IOU will be measured as low as the size ratio. But instead, we really need it to output IOU as 1 to be classified as “inside the table”. Therefore, we design a modified IOU: it compares the bounding box coordinates to see if the textline bounding box is inside the table box. If it is, the modified IOU will output 1. Otherwise, it computed the regular IOU.

After IOU computation, we label bounding boxes that have IOU higher than 0.5 to have “table” label. Even though we set the IOU threshold as 0.5, it is observed that most IOUs are 1 or higher than 0.8. This is due to the accurate localisation of textline bounding boxes in our dataset and the high quality table bounding boxes from CascadeTabNet. This observation reasserts the quality of our proposed dataset.

4.2.9 Visual Feature Generation

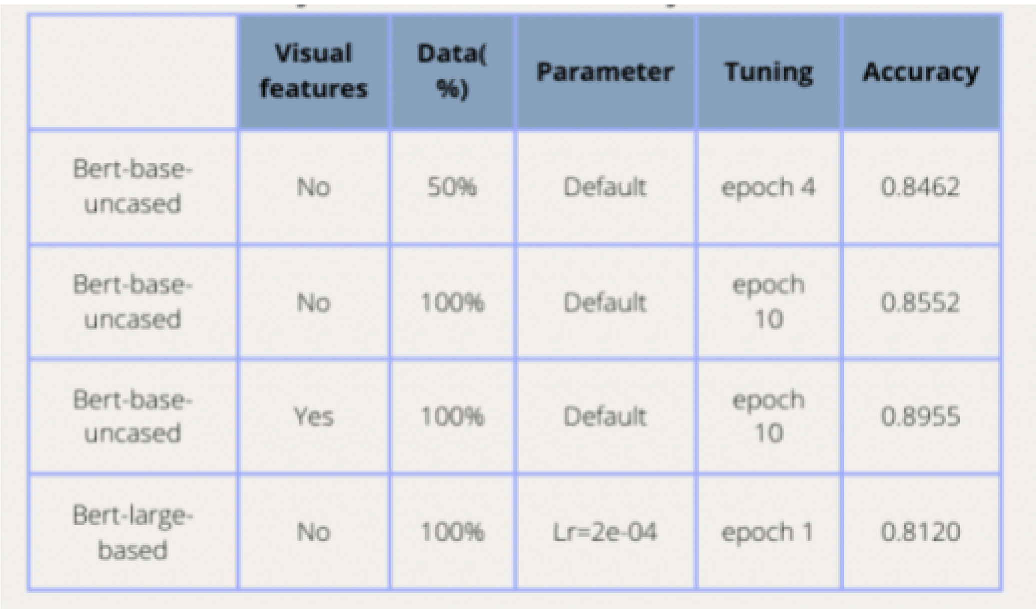
Next we kit our bounding boxes with visual features from the corresponding PDF images generated by detectron2. Detectron2 is a pretrained object detection model by Facebook AI Research. Its high capacity ensures a satisfying feature representation that is applicable for other visual tasks as well. Therefore, we use the feature representation learning backbone in Detectron2 to generate the visual features for each of the textline bounding boxes. Such visual features provide more insights to the label classification task and can aid the task for a better performance (see next section).

4.2.10 Model Training and Fine-tuning

With the above procedures correctly done, the label classification dataset has over 5,000 “table” samples, 1,000 “label” samples, 45,000 “table” samples, 34,000 “text” samples. Then we randomly split the dataset into 70% training data and 30% testing data. To be exact, the training set contains 60,202 samples while the testing set has 25,801 samples.

To be clear, this label classification dataset is not our proposed Document Form Understanding dataset, but derived from the proposed dataset by the above procedure. The label classification dataset have each bounding box as a single sample and our proposed dataset have each PDF document as a single sample.

Then we use the pretrained bert-base-uncased and bert-large-uncased model to fine-tune on the label classification dataset. This leverages the power of transfer learning. In Figure 6, we show our model’s variants’ results for with visual features and without visual features. We show the test results with the best learning rate and best epoch checkpoint. It can be observed that models with visual features have significantly higher accuracy. This solidifies the usefulness of visual features. We also show the effectiveness of only 50% of training data. It only suffers from 0.01 accuracy decrease but saves half of the training time. The unsatisfying bert-large model result (0.03 lower than bert-base under the same condition) can be explained as: the label classification task is relatively too simple for bert-large and the model suffers from overfitting and is not suitable for this task.



	Visual features	Data(%)	Parameter	Tuning	Accuracy
Bert-base-uncased	No	50%	Default	epoch 4	0.8462
Bert-base-uncased	No	100%	Default	epoch 10	0.8552
Bert-base-uncased	Yes	100%	Default	epoch 10	0.8955
Bert-large-based	No	100%	Lr=2e-04	epoch 1	0.8120

Figure 6: Results of pretrained Bert model fine-tuning on our label classification dataset.

4.3 Data Analysis

The data analysis of the project will include two parts which are dataset analysis and visual analysis. The dataset analysis contains three major sections. The layout analysis, entity recognition, and named entities. The main technique used for layout analysis is Read Coop. Read Coop provides a cloud api that has no cost and unlimited request. The Read Coop api can determine the reading order and store it in Json file for each document. NER is used to achieve entity recognition in this project. Hankiewicz (2020) introduces that the Named entity recognition (NER) can extract concurrency value, names and address from the text paragraph. The NER can meet the demands of assistance with the text summarisation which could be a successive work based on final deliverables of this project. In addition, NER applies multiple python packages which do not request fetching data from other platforms which will be the most suitable technique for the massive dataset. The top 10 appearance text of the dataset is the solution for generating the named entities. The entity distribution will present at the visual analysis section.

There are two high-level visualisations and six low-level visualisations created for visual analysis. The high-level visualisations are mainly created by Microsoft Excel applications. One of the visualisations is the dataset comparison table, through comparison with the successful FUNSD dataset to identify the difference between our dataset and the existing related dataset (Jaume et al., 2019). The other one is the model training analysis table, this visualisation is to show the model training result of our dataset. The low-level visualisations are created by Tableau software. The tableau software is an innovative data visualisation tool which is perfectly suitable for our visualisation. Three pie charts and three bar charts are created by Tableau to present the low-level visualisations.

5. Resources

5.1 Hardware & Software

This project mainly adopted Google cloud ecosystem as the development environment. Related softwares is listed below:

Software name	Category	Usage detail and intention
Google Drive	Development	1. Data and deliverables storage. 2. Data and deliverables sharing.
Google Colab	Development	1. Programming. 2. Algorithm implementation.
Google Cloud API	Development	1. OCR implementation.
Github	Development	1. Version control. 2. Code storage.
Trello	Development	1. Project management 2. Task tracking
Google sheet	Documentation	1. Collaborative documentation. 2. Written communication.
WeChat	Communication	1. Communicate with tutors. 2. Daily communication within the team.
Zoom	Communication	1. Weekly online meeting. 2. Milestone completion meeting. 3. Deliverable demo with clients. 4. Workshop.

5.2 Materials

Related materials in this project included several dataset and experiment results published by other researchers which include FUNSD dataset (Jaume et al., 2019) which is very similar with our documents as a dataset. Related essay and experiment record of PubLayNet is also required which is the largest document layout analysis dataset till present, published by Zhong, Tang, and Jimeno Yepes in 2019.

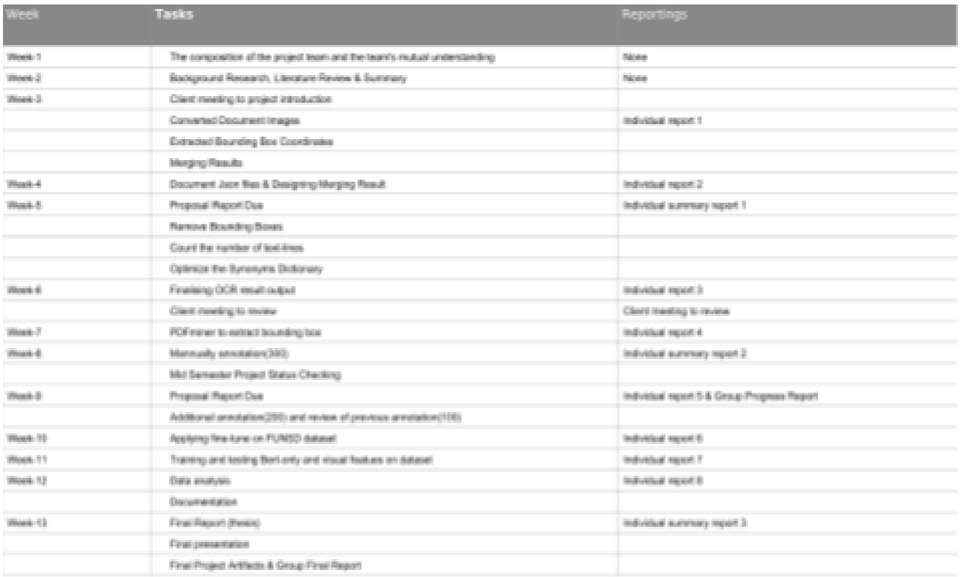
5.3 Roles & Responsibilities

In this interesting project, all team members were responsible for more than one type of task. It is appropriate to conclude that the development team consists of 1 project manager, 3 developers, and 2 document managers. Related responsibility specification for each member is listed below:

Member	**Role**	**Responsibility**
Kaer Li (510554284)	Project manager Technical manager	1. Assign task breakdown to group members. 2. Monitor project progress and adjust the workload. 3. Review code and completed works.
Xing Xing (500390560)	Technical manager Developer	1. Implement python code based on requirements. 2. Support data analysts for visualisation creation. 3. Train, evaluate, and tune machine learning models.
Shuai Liu (510300441)	Technical manager Developer	1. Implement python code based on requirements. 2. Support data analysts for visualisation creation. 3. Train, evaluate, and tune machine learning models.
Wenjie Jiang (510250146)	Technical manager Data analyst	1. Implement python code based on requirements 2. Implement and design data analysis. 3. Manage documentations and report level deliverables
Yuanyuan Lin (500595497)	Document manager Data analyst	1. Implement part python code based on requirements 2. Implement and design data analysis poster 3. Manage documentations and report level deliverables
Yuxi Shen (490548486)	Client liaison	1. Implement python code based on requirements 2. Communicate with clients for specific requirement 3. Deliver deliverables to clients and receive feedback.

6. MILESTONES / SCHEDULE

As mentioned in the group proposal, a schedule has been developed for this project, but unlike the previous schedule, the last eight weeks of uncertainty have been eliminated. These two images are also incorporated into the project, which are the project's final milestones for implementing the plan with the client.



Week	Tasks	Reportings
Week-1	The composition of the project team and the team's mutual understanding	None
Week-2	Background Research, Literature Review & Summary	None
Week-3	Client meeting to project introduction	
	Converted Document Images	Individual report 1
	Extracted Bounding Box Coordinates	
	Merging Results	
Week-4	Document Json files & Designing Merging Result	Individual report 2
Week-5	Proposal Report Due	Individual summary report 1
	Remove Bounding Boxes	
	Count the number of text-lines	
	Optimize the Synonyms Dictionary	
Week-6	Finalising OCR result output	Individual report 3
	Client meeting to review	Client meeting to review
Week-7	PDFminer to extract bounding box	Individual report 4
Week-8	Mannually annotation(300)	Individual summary report 2
	Mid Semester Project Status Checking	
Week-9	Proposal Report Due	Individual report 5 & Group Progress Report
	Additional annotation(200) and review of previous annotation(100)	
Week-10	Applying fine-tune on FUNSD dataset	Individual report 6
Week-11	Training and testing Bert-only and visual features on dataset	Individual report 7
Week-12	Data analysis	Individual report 8
	Documentation	
Week-13	Final Report (thesis)	Individual summary report 3
	Final presentation	
	Final Project Artifacts & Group Final Report	

Figure 7(a) - Milestone

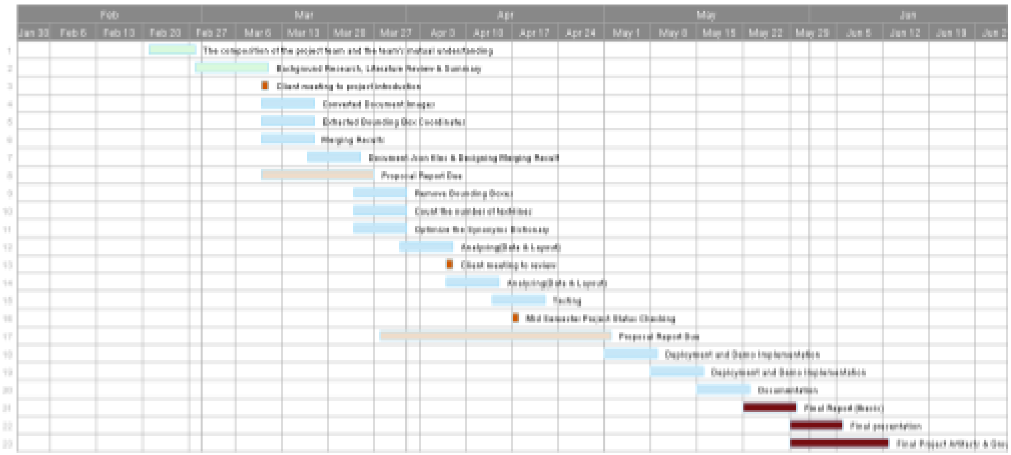


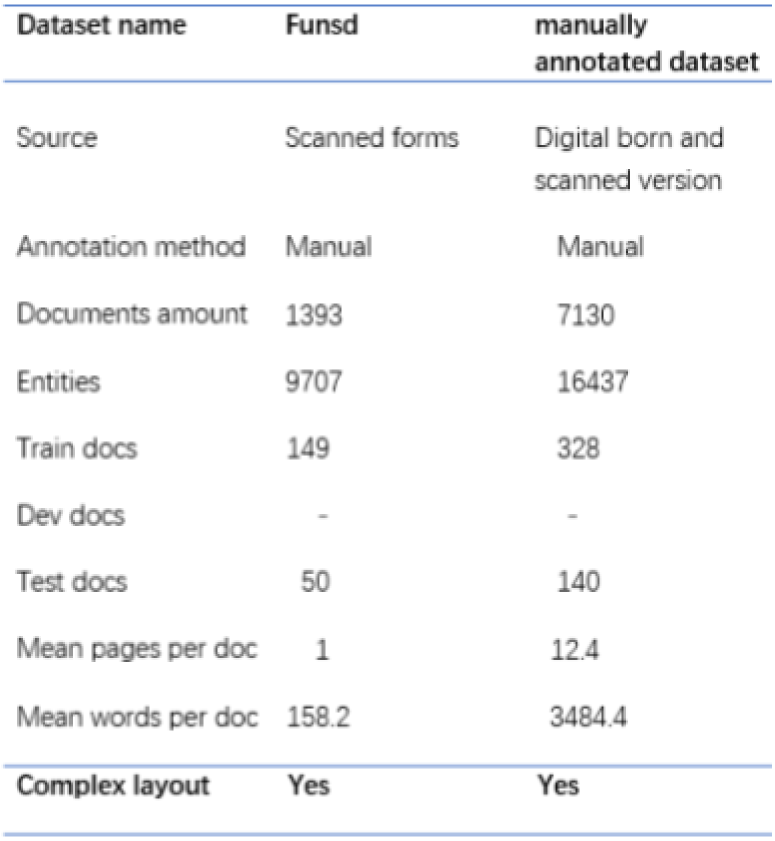
Figure 7(b) - Milestones

7. Results

There are three main results derived from this project which are the final dataset, model testing result and data analysis result.

Dataset

The first and most significant output is our final derived dataset, which is presented in CSV format. **Four columns,** serial number, text, label, and visual feature, are applied to **86,248 observations** in the dataset. This dataset was created by manually annotating 500 PDF documents chosen at random. It is discovered that some annotation rules may be applied slightly differently due to the habits of different annotators. As a result, we coordinated all of the standards in the last 300 documents annotated and randomly selected the first fifty percent of the hundred documents that were re-annotated by two or more team members. Following a comparison, it was determined that over seventy-five percent of the annotations from the original annotation method are consistent with the newly defined annotation rules from the subsequent stage. In addition, the project team compared the FUNSD dataset (Jaume et al., 2019) to the manually-annotated dataset, revealing that the scale of FUNSD (Jaume et al., 2019) is significantly smaller than that of our target dataset.

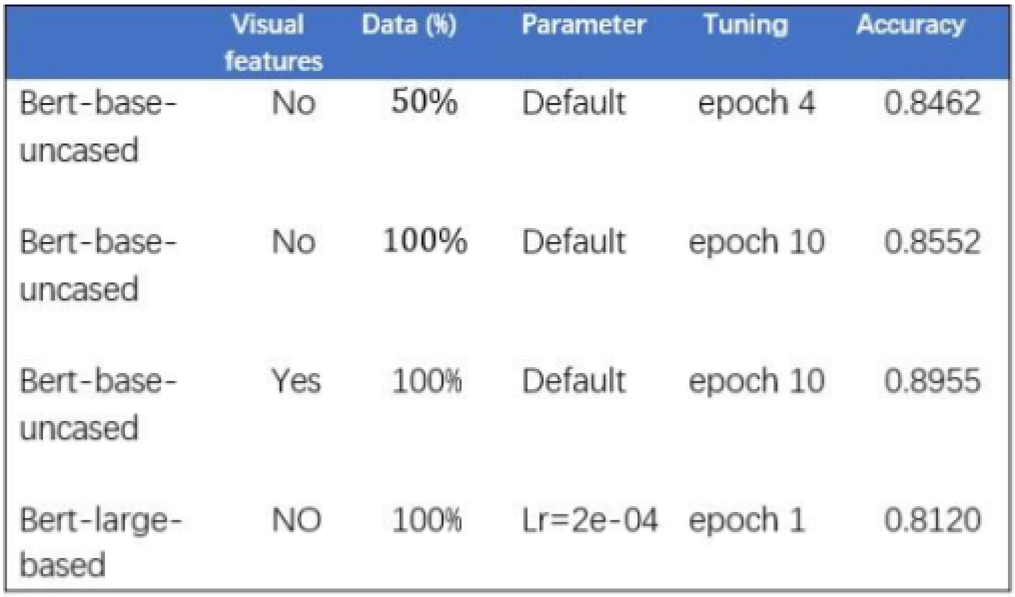


Dataset name	Funsd	manually annotated dataset
Source	Scanned forms	Digital born and scanned version
Annotation method	Manual	Manual
Documents amount	1393	7130
Entities	9707	16437
Train docs	149	328
Dev docs	-	-
Test docs	50	140
Mean pages per doc	1	12.4
Mean words per doc	158.2	3484.4
Complex layout	**Yes**	**Yes**

Figure 8 - Dataset comparision

Model Testing

Based on our comparison of parameter tuning, the optimal configuration is the Bert-based model, which performs well on the test set using 50 percent of the training data and the default parameters, which reaches the highest accuracy of 0.8552. After comparing the visual characteristics, it has been determined that training with 100 percent of the data has the optimal effect on the test set, which reaches 0.8955.



	Visual features	Data (%)	Parameter	Tuning	Accuracy
Bert-base-uncased	No	50%	Default	epoch 4	0.8462
Bert-base-uncased	No	100%	Default	epoch 10	0.8552
Bert-base-uncased	Yes	100%	Default	epoch 10	0.8955
Bert-large-based	NO	100%	Lr=2e-04	epoch 1	0.8120

Figure 9 - Model performance

Data Analysis

In addition, during the data analysis, eight visualisations are created, including two high-level and six low-level visualisations. These are primarily intended for CSV datasets and annotation statistics.This is an analysis of the FUNSD (Jaume et al., 2019) dataset's files. The majority of files are between 2 and 4 pages, accounting for half of the total dataset, while the single-page file is the least.

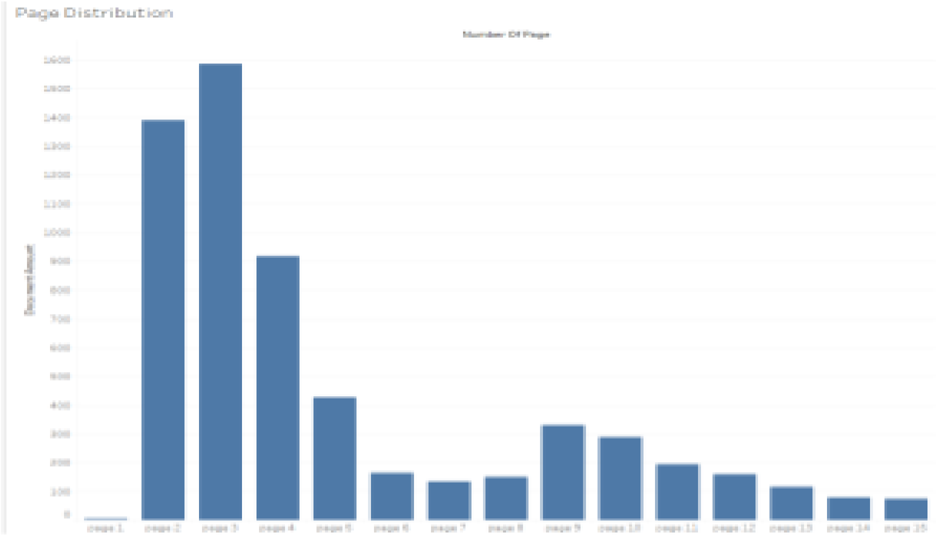


Figure 10 - Page distribution

The subsequent pie chart illustrates the distribution of the number of pages in the dataset of human annotations. Two-page documents account for nearly 70 percent of the total, whereas five-page and one-page documents are uncommon, accounting for less than 1 percent of the total.

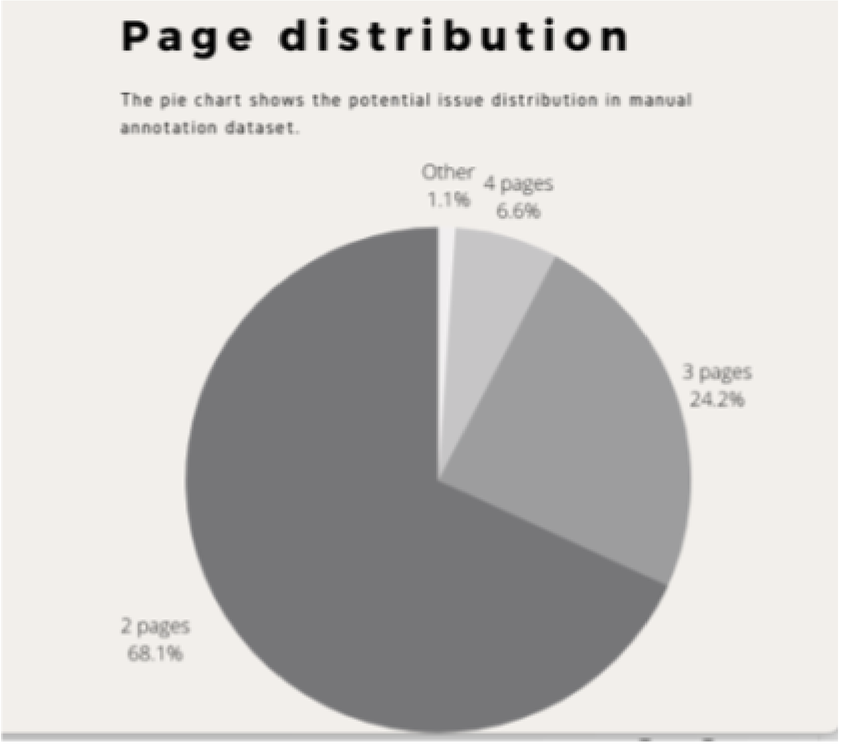


Figure 11 - Page distribution

During the process of implementing the project, particularly during the manual annotation phase, some potential issues were uncovered. By counting all potential problems, it can be seen that nearly 50 percent are missing bounding box, which means there is no bounding box surrounding the content, whereas only 15 percent are caused by the bounding box being too small and not completely encompassing the content.

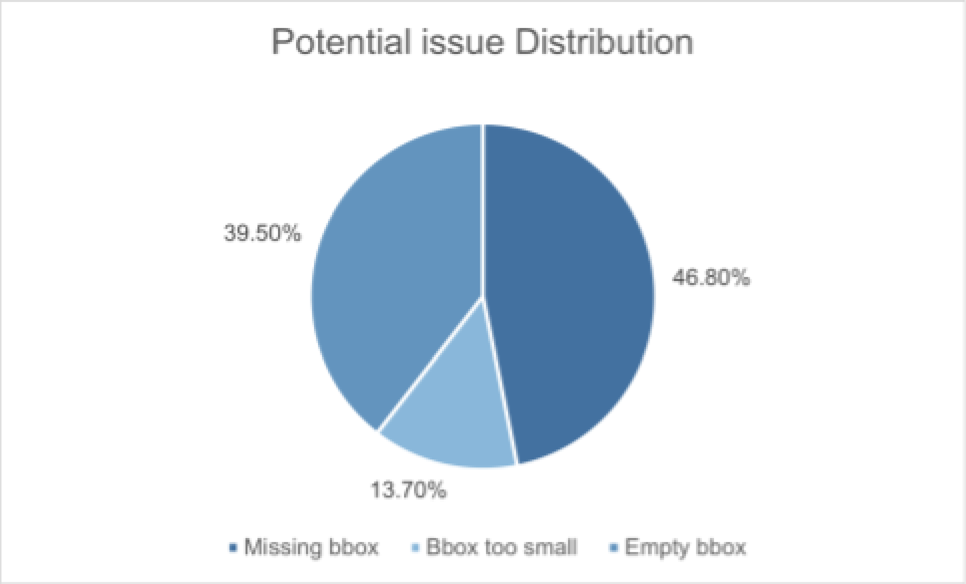


Figure 12 - Potential issue distribution

A total of 439 pdfs were identified in the data set, from which a total of 86248 labels were derived, including Text, Title, Section, and Table. Statistical analysis reveals that Table and Text occupy the largest proportion, while Title occupies the smallest.

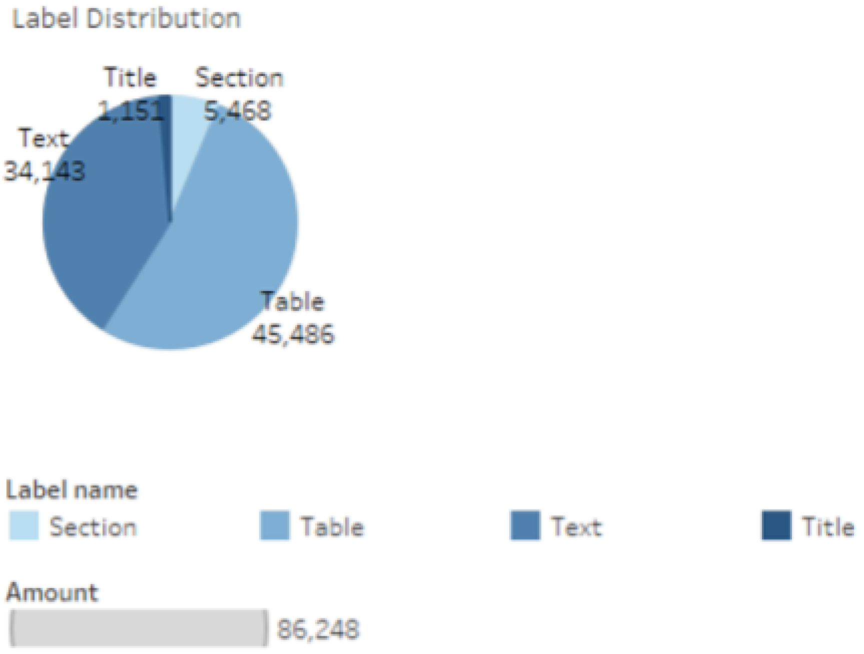


Figure 13 - Label distribution

In the manually-annotated dataset, by aggregating all texts, the organized dataset appears nearly 7500 times, which indicates that the files that generate this dataset may be more focused on the business direction.

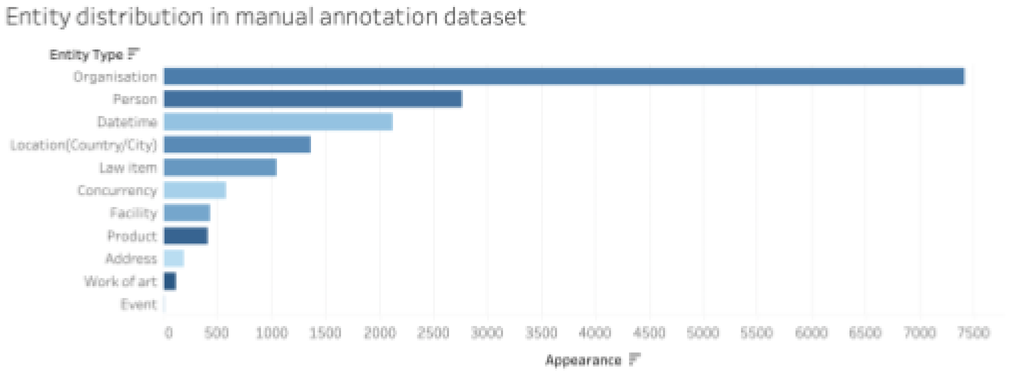


Figure 14: Entity distribution

A year distribution was obtained from the same dataset by identifying the year in the file. The majority of documents are concentrated in 2015 and 2011, with only a few appearing between 2002 and 2013.

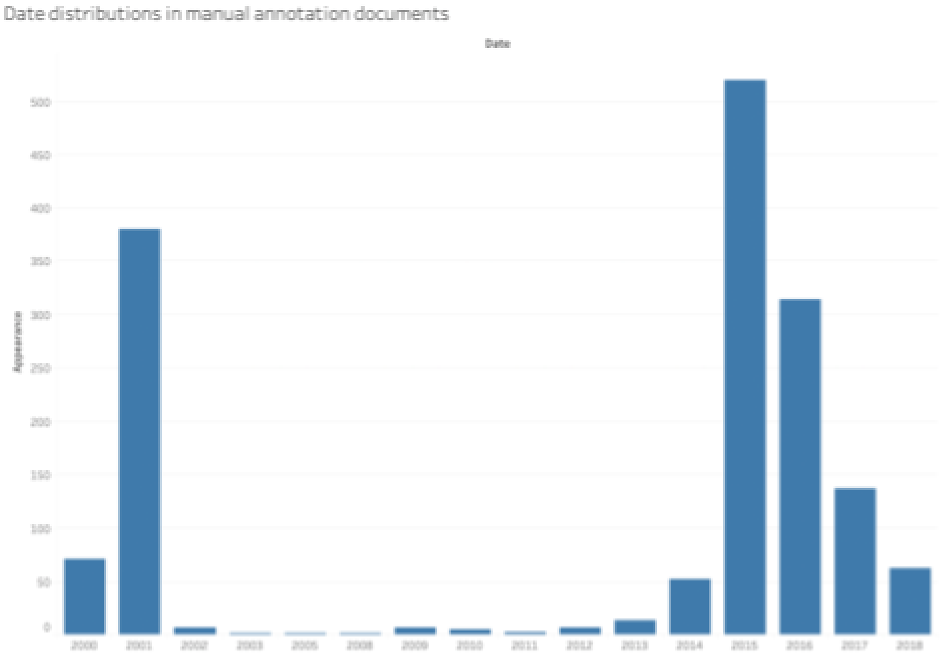


Figure 15: Date distribution

Due to the excessive size of the original raw dataset, the time cost, computational cost, and economic cost of using the OCR API to extract bounding boxes are excessively high, thereby limiting the diversity of the results. In addition, FUNSD (Jaume et al., 2019) displays other features in a more comprehensive manner. Nonetheless, in the process of document analysis, when the project team attempted to display the dataset, since the dataset was too large compared to FUNSD (Jaume et al., 2019) and the time cost was also too high, manually-annotated dataset only displays label classification when displayed, which is unfortunate and can be improved.

8. Discussion

First, for our main deliverable, the CSV file. We compare it with the results of manual annotation, and we find that the accuracy rate for content extraction can reach about 75%, which is acceptable and even exceeds our expectations for extracting 7000 documents with a total of more than 86,000 pages.

Secondly, for the classification model, by trying different models and parameters, the highest model performance can reach 85%. If further visual features are introduced, the accuracy of the model will be further improved, which can reach 89%. This shows that our classification model and parameters are reasonable.

Last but not least, for data analysis, we provide a total of 7 charts to help understand the entire dataset, such as document page count distribution, document year distribution, label distribution, potential issue distribution, and performance tables for different classification models. People who are new to datasets can use these to gain some preliminary understanding of the datasets, which can help with subsequent data analysis on the datasets, which will greatly reduce the time to familiarize themselves with the datasets.

In general, we believe that our project may be helpful for the extraction of PDF file formats. If a completely new file needs to be processed, for the extraction and merging of text boxes, and the classification of text boxes, our model will give a satisfactory result.

9. Limitations and Future Works

Obviously, we know that we still have many shortcomings. For example, if the PDF file format is too complex or too compact, this will lead to varying degrees of problems with text box extraction and merging. For handling more complex cases, we have not tested it yet, but from our experience, it may be necessary to change the text box merging rules. The current text box merging rules are determined according to the distance of the text boxes, and future work may need to find a better method, for example, to judge the association of text boxes through the extracted text. In addition, future work may include the splitting of bounding boxes and the removal of blank bounding boxes.

And moreover, the number of training samples may be relatively small. We only manually annotated 439 files, and used 300 files as the training set and the rest as the validation set, so, in this case the training set only accounted for about 4% of the total 7000 files. However, the size of data is sufficient. Currently, each epoch takes about 1 hour for model training. If the value of epoch is set to 10, it will take more than 10 hours to get a result, and obviously the time cost is also sufficient. Therefore, if we increase the number of samples of the training set in the further, it is best to keep it below 500 files. In addition, the number of document pages we process is mainly concentrated below 5 pages. In the future, in addition to increasing the number of training documents, we can further increase the processing of documents with more pages. However, now due to time and platform constraints, what we can do is very limited. The main limitation is the training of the model on Google Colab, when we train with such a small number of files, there are often problems with running out of time due to long run times. Therefore, if the running speed can be improved in the future, and the running time is not limited, the accuracy of the classification model will be further improved.

In conclusion, our current work may be insufficient for the processing of complex format documents, and the richness of the training set for classification may be low. Therefore, the focus of future work is to further improve the accuracy of bounding boxes and increase the diversity of the training set while ensuring that it is not affected by the running platform.

References

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019, May 24). Bert: Pre-training of deep bidirectional Transformers for language understanding. Retrieved June 9, 2022, from https://arxiv.org/abs/1810.04805

Graliński, F., Stanisławek, T., Wróblewska, A., Lipiński, D., Kaliska, A., Rosalska, P., . . . Biecek, P. (2020, March 06). Kleister: A novel task for information extraction involving long documents with complex layout. Retrieved June 9, 2022, from https://arxiv.org/abs/2003.02356

Hankiewicz, K. (2020, Feb). Applying Natural Language Processing for intelligent document analysis. Retrieved May 27, 2022, from https://medium.com/untrite/applying-natural-language-processing-for-intelligent-document-analysis-a91bcb85919b

Wang, Z., Zhan, M., Liu, X., & Liang, D. (2020, October 15). DocStruct: A multimodal method to extract hierarchy structure in document for general form understanding. Retrieved June 9, 2022, from https://arxiv.org/abs/2010.11685

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., . . . Zhou, L. (2022, January 10). Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. Retrieved June 9, 2022, from https://arxiv.org/abs/2012.14740

Jaume, G., Ekenel, H. K., & Thiran, J. (2019, September 22). FUNSD: A dataset for form understanding in noisy scanned documents. Retrieved March 27, 2022, from https://ieeexplore.ieee.org/abstract/document/8892998?casa_token=gLmkl8ej2ckAAAAA%3AjWoat7B6nIfBTlLSH6wwjRBVSgLrb2vy32UTkg0oMIy5UMeB4J6oZeCcwlGWB5MPe8joPHxG

Zhong, X., Tang, J., & Yepes, A. J. (2019, September). PubLayNet: Largest dataset ever for document layout analysis. Retrieved June 9, 2022, from https://ieeexplore.ieee.org/document/8977963/references#references