Cheque Information Processing and Validation using Deep Learning

A project report submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Electronics & Computer Engineering

by

Darshan N Shenoy

20BLC1076

School of Electronics Engineering,

Vellore Institute of Technology, Chennai,

Vandalur-Kelambakkam Road,

Chennai - 600127, India.

April 2024

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Declaration

I hereby declare that the report titled ***Cheque Information Processing and Validation using Deep Learning*** submitted by me to the School of Electronics Engineering, Vellore Institute of Technology, Chennai in partial fulfillment of the requirements for the award of **Bachelor of Technology** in **Electronics and Computer Engineering** is a bona-fide record of the work carried out by me under the supervision of***Dr.K Suganthi***.
I further declare that the work reported in this report, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Sign: _____

Name & Reg. No.: _____

Date: _____

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering

Certificate

This is to certify that the project report titled ***Cheque Information Processing and Validation using Deep Learning*** submitted by ***Darshan N Shenoy*** **(20BLC1076)** to Vellore Institute of Technology Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Computer Engineering** is a bona-fide work carried out under my supervision. The project report fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Supervisor

Signature: .....

Name: .....

Date:

Head of the Department

Signature: .....

Name: .....

Date:

Examiner

Signature: .....

Name: .....

Date:

(Seal of the School)

Abstract

Understanding document images, such as invoices, is a difficult task. It necessitates reading the text as well as comprehending the document's general structure. Conventional techniques first extract text using Optical Character Recognition (OCR), yet this method has drawbacks. In addition to being computationally costly, OCR can have trouble converting between various languages and document formats and introduce errors that can compound throughout the analysis process. Donut is an OCR-free VDU (Visual Document Understanding) model that challenges this paradigm. It does not even bother with the text extraction stage; it analyzes the document image straight. This creative strategy produces notable advancements. Donut performs at the cutting edge, which means it is quick and precise. This is a significant development in the realm of visual document understanding since it enables Donut to handle a greater variety of languages and document types. The model has been tested using BLEU scores and ROUGE scores over 3 parameters namely – payee name, amount in words and amount in figure. The model has also been tested over variety of sample sizes ranging from 50 to 1000.

Acknowledgements

I wish to express our sincere thanks and deep sense of gratitude to my project guide, Dr. K Suganthi, Associate Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to me in a pleasant manner throughout the course of the project work.

I am extremely grateful to Dr. Susan Elias, Dean Dr. Reena Monica, Associate Dean (Academics) & Dr. John Sahaya Rani Alex, Associate Dean (Research) of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

I express our thanks to our Head of the Department Dr. Annis Fathima A for her support throughout the course of this project.

I also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to me throughout the course.

I thank our parents, family, and friends for bearing with me throughout the course of our project and for the opportunity they provided me in undergoing this course in such a prestigious institution.

Contents

List of Figures

Chapter 1

Introduction

Document images, such as those of business cards, invoices, and receipts, are widely used in today's digital environment. Visual Document Understanding (VDU) is necessary to uncover the valuable information they hold. While VDU is important for many businesses, it also poses a huge research challenge. Fortunately, VDU provides a variety of applications, including document classification, which effectively sorts diverse document kinds. It also helps with information extraction, allowing us to automatically extract crucial data such as names, dates, and quantities from document images. VDU even has visual question responding, in which users can ask questions about the document's content and receive responses based on visual analysis. Overall, VDU plays an important role in efficiently handling document pictures, resulting in developments in a variety of applications.

Visual Document Understanding (VDU) is a two-step process that starts with text reading and ends with holistic understanding; nevertheless, the success of VDU depends on precise text extraction. This is where optical character recognition (OCR) technology is frequently used in VDU procedures. OCR, however, comes with a lot of difficulties. Firstly, it can be computationally costly to achieve high-quality OCR results. Second, commercial OCR engines are not flexible enough to accommodate many languages or specialized fields. Extensive datasets and significant resources are needed for training specialized OCR models to overcome these constraints. Above all, mistakes made during the OCR process have the potential to affect the whole VDU system and cause problems. The complicated character sets present in languages like Chinese and Korean exacerbate this problem even more. Post-OCR correction modules are an attempt by VDU to address these issues, however they add complexity and raise system maintenance costs. OCR limitations and accompanying expenses are a challenge for VDU systems. For researchers at VDU, finding a balance between efficiency and precision is still a major issue.

The Donut model is an innovative method to VDU that researchers have developed. This approach is unique in that it does not require OCR at all; instead, it only uses a Transformer architecture. This invention addresses the drawbacks and expenses that are frequently connected to conventional VDU approaches' reliance on OCR. Donut goes through a pre-training phase to acquire text reading skills. At this point, the model effectively gains the ability to "read" text by making predictions about the following words using the image and any previously recognized text as guidance. After that, Donut is adjusted to suit particular activities with the ultimate goal of comprehending the meaning of the entire document. Donut's pre-training with synthetic data is a major strength. By using this method, the model can overcome the constraints of off-the-shelf OCR engines and become flexible and adaptive across a variety of languages and disciplines. Surprisingly, Donut has achieved state-of-the-art performance in terms of speed and accuracy across several VDU tasks, demonstrating remarkable results despite its simplicity. Donut is positioned as a viable VDU solution going forward due to its capacity to avoid OCR reliance.

Donut, an end-to-end trained, OCR-free Transformer model for VDU, is a major advancement. The difficulties and restrictions that have historically been connected to OCR—such as problems with quality, cost, and error propagation—are eliminated by this breakthrough. Donut's clever pre-training program is key to its success. In this stage, by forecasting the subsequent words based on the image and any previously recognized text, the model effectively learns to "read" text. This completely removes the requirement for a different OCR engine. Furthermore, Donut uses artificial data produced by the specialized data generator SynthDoG. One significant benefit of this synthetic data methodology is that Donut may be readily extended to multilingual applications, in contrast to traditional methods that necessitate retraining for multiple languages. Donut's abilities have been validated by extensive testing. On both industry-private datasets and publicly available benchmarks, it delivers state-of-the-art performance. Beyond its remarkable accuracy, Donut has a lot of useful features. For practical VDU applications, its cost-effectiveness and efficiency make it an extremely appealing option. Donut, in a sense, sets the stage for a time when OCR reliance won't impede VDU. This heralds in a new era of reliable, affordable, and adaptable document understanding.

Chapter 2

Literature Survey

Cheque processing remains a crucial task for banks and financial institutions, necessitating efficient and accurate automation. Traditional methods rely on Optical Character Recognition (OCR) for data extraction, but can struggle with factors like handwritten information, complex layouts, and even smudges or tears on the cheque itself [2]. This section explores recent advancements in deep learning approaches for cheque processing automation, highlighting the move towards OCR-free techniques.

2.1 Deep Learning and Automation

[1] propose a deep learning-based system for automated cheque verification. Their approach utilizes image processing techniques for cheque image pre-processing, such as noise reduction and binarization. Convolutional Neural Networks (CNNs) then extract features from the processed image, allowing the model to learn patterns and identify crucial elements like account numbers, signatures, and amounts. Finally, the system classifies the cheque based on pre-defined categories, such as valid/invalid or sufficient/insufficient funds. Similarly, [2] present a deep learning framework that combines CNNs with OCR for cheque transaction automation. While effective, these methods still rely on OCR for text recognition, introducing a potential source of error and inefficiency.

2.2 Moving beyond OCR

Recent research explores overcoming limitations of OCR by venturing into OCR-free document understanding. [11] introduce Donut, a document understanding transformer

model that bypasses the need for a separate OCR stage. Donut directly learns from the visual features of the document image, focusing on spatial relationships between elements and leveraging techniques like self-attention to understand the context of different regions within the cheque. This approach achieves promising results, particularly in scenarios where traditional OCR struggles. This concept is further explored in Kim et al. [2023], where the authors address text localization within an end-to-end OCR-free framework. Here, the model not only extracts information but also identifies the specific locations of text elements on the cheque, improving data extraction accuracy.

2.3 Multimodal Pre-training

While both Donut and [8] move beyond OCR, they differ in their approach to pre-training. LayoutLMv2 leverages a multi-modal architecture that incorporates both textual and layout information during the pre-training phase. This allows the model to learn relationships between textual content and its visual representation on the document. Imagine a pre-training dataset containing a vast collection of labelled cheques. LayoutLMv2 wouldn't just learn the character sequences for "account number" but also how this text typically appears relative to other elements on the cheque, like borders or logos. This pre-training on multi-modal data potentially improves the performance of document understanding tasks like cheque processing by providing the model with a richer understanding of the document structure.

2.4 Evaluation Benchmark

It's important to acknowledge the role of standardized evaluation benchmarks in comparing and advancing cheque processing automation techniques. Competitions like the ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction [2,4,7] provide datasets and evaluation metrics that allow researchers to objectively assess the performance of their models. This fosters healthy competition and drives innovation in the field.

2.5 Future Directions

Research in cheque processing automation continues to evolve. Future advancements might explore integrating contextual information, such as account holder details or past

transaction history, to further enhance the accuracy and robustness of cheque processing systems. Additionally, research on explainability of deep learning models could be crucial for building trust and transparency in automated cheque processing, especially within the financial sector.

In conclusion, deep learning presents significant opportunities for automating cheque processing. While initial approaches relied on OCR, recent advancements explore OCR-free techniques and multi-modal pre-training, paving the way for more robust and efficient cheque automation systems. As research progresses, incorporating contextual information and ensuring explainability will be key to building reliable and trustworthy cheque processing solutions.

Chapter 3

Methodology

3.1 Dataset

The dataset available on Kaggle comprises 10,000 cheque images sourced from various financial institutions. These cheques represent a diverse range of banks, including prominent Indian banks such as Axis Bank, State Bank of India (SBI) and Canara Bank as well as international banks like HSBC and Bank of America.

1. .**Payee Name:** This field captures the recipient's name—the individual or entity to whom the cheque is payable. It plays a crucial role in verifying the legitimacy of the transaction.
2. **Amount in Words:** The "Amt in Words" section represents the monetary value of the cheque expressed in words. This textual representation is essential for cross-referencing with the numerical amount.
3. **Amount in Figures:** The "Amt in Fig" corresponds to the numerical value of the cheque. It is the actual monetary amount written in digits. Verifying consistency between the amount in words and the amount in figures is critical for accuracy.
4. **Bank Name:** The bank that issued the cheque is identified in this parameter. Whether it is an Indian bank or an international one, this information helps contextualize the transaction.

3.2 Document Understanding Transformer

Donut stands out as a self-contained document understanding model. At its core lies a powerful duo: a visual encoder and a textual decoder. The visual encoder, built on

Transformer architecture, analyzes the document image, extracting key visual features without relying on external OCR engines. This extracted information is then passed to the textual decoder, another Transformer component. The textual decoder interprets these features and translates them into a sequence of sub word tokens. Finally, these tokens are assembled into a structured format, like JSON, allowing Donut to represent the document's content in a human-readable way. This end-to-end workflow, facilitated by Transformers, empowers Donut to efficiently process document images and unlock their textual meaning.

Donut dives deep into document images to extract their meaning. The first step is the visual encoder. Imagine this encoder as an image analyst. It receives a document image as input, with each pixel represented by its height, width, and color channel. The encoder then meticulously converts this image into a set of mathematical representations called embeddings. These embeddings are like compressed descriptions of the image's key features. The number of embeddings (n) corresponds to the image being chopped up into a grid of features, and the size of each embedding (d) reflects the complexity of the information it captures.

The choice of encoder architecture is crucial. Donut can leverage two options: Convolutional Neural Networks (CNNs), which are image processing workhorses, or Transformer-based models like the Swin Transformer. In this case, the researchers opted for the Swin Transformer due to its impressive performance in understanding document layouts during their initial tests.

The Swin Transformer itself is a multi-stage process. First, it meticulously dissects the image into non-overlapping patches, like tiles in a mosaic. Each patch then gets scrutinized by specialized Swin Transformer blocks. These blocks employ a sophisticated technique called "shifted window-based multi-head self-attention" to analyze relationships between the patch and its neighbors. Additionally, a two-layer Multi-Layer Perceptron (MLP) adds further depth to the analysis. As the Swin Transformer progresses through stages, it merges information from these processed patches, gradually building a comprehensive understanding of the visual content. Finally, the output of the Swin Transformer, a collection of refined embeddings, becomes the visual representation of the document, ready to be deciphered by the next stage.

Donut then utilizes a textual decoder to translate the visual understanding into human-readable text. Imagine this decoder as a skilled translator. It receives the set of visual features (z) extracted by the Swin Transformer, which act like a compressed description of the document's content. The decoder's goal is to generate a sequence of tokens, the building blocks of text. Each token is represented by a unique "one-hot vector," essentially a code indicating which specific word it represents out of the entire vocabulary (v) the model knows. The length of this generated sequence (m) is controlled by a predefined parameter.

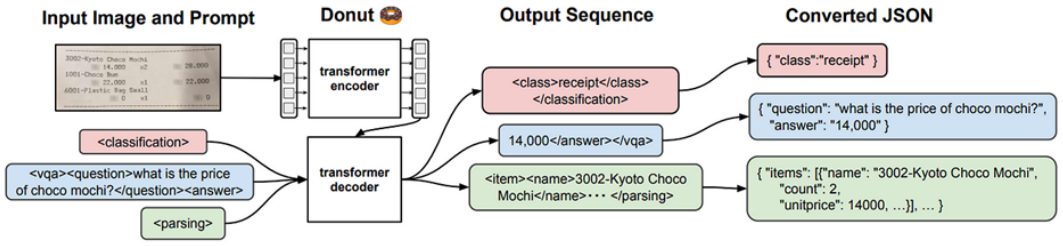


FIGURE 3.1: Donut model pipeline

To achieve this translation, Donut relies on a specific architecture called BART (Bidirectional and Auto-Regressive Transformers). BART is like a pre-trained translator that has already learned the intricacies of many languages. Here, Donut leverages a publicly available, pre-trained multilingual BART model to initialize the decoder's internal workings, giving it a strong foundation for understanding different languages that might appear in documents.

During training, Donut utilizes a "teacher-forcing scheme" to guide the decoder's learning. This means the decoder is fed the correct answer (ground truth) at each step, like a student being shown the solution while learning. However, during the actual usage (test phase), Donut takes inspiration from GPT-3, a powerful language model. It provides the decoder with a starting point called a "prompt," similar to giving a translator a first sentence or two to get them going. Additionally, for each specific task Donut is tackling (like document classification or question answering), new specialized tokens are incorporated into the prompt to steer the decoder towards the desired outcome. This combination of pre-trained knowledge, teacher-forced learning, and well-crafted prompts empowers Donut's decoder to translate the visual representation of documents into meaningful text.

3.3 Proposed Architecture

The figure above depicts the proposed system architecture. Here is a brief description of the design-

1. **Input Check Image**

   The first step involves feeding a digital image of the cheque into the system. This image can be captured from a variety of sources, including a check scanner at a physical bank branch or a mobile banking app.

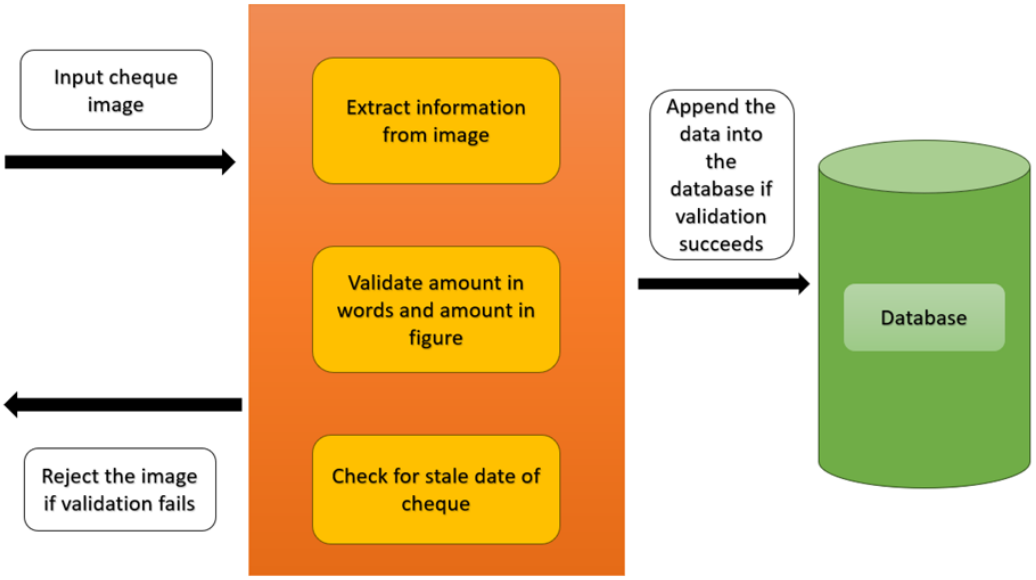


FIGURE 3.2: Proposed architecture diagram

2. **Extract Information from Image**

   Once the image is captured, the system employs optical character recognition (OCR) to extract relevant information from the cheque. OCR is a technology that enables a computer to recognize and interpret text characters from an image. In the case of a cheque, the OCR engine would be designed to identify and extract the following data points:

   * Payee name
   * Date
   * Amount (written in words)
   * Amount (written in numerals)
   * Bank name

3. **Validate Extracted Data**

   After the information is extracted, the system performs a series of validations to verify its accuracy and authenticity. Here are some common validation checks:
   Verification of amount in words and figures: The system checks if the amount written in words matches the amount written in numerals. Any discrepancies between these two values would flag an exception.
   Date validation: The system verifies if the date on the cheque is current and not stale dated.

4. **Append Data to Database**

   If all the validation checks are successful, the extracted data is appended to the

bank's database. This data is typically stored in a secure electronic record that is linked to the digital image of the cheque.

5. **Rejection of Cheque Image**
   If any of the validation steps fail, the cheque image is rejected by the system. The reason for rejection would be communicated back to the user, allowing them to take corrective measures.

Chapter 4

Results and Discussions

Standard machine learning performance metrics can not be used to find efficacy of Text extraction models. The major reasons for this are -

1. **Textual Complexity**
    - NLP deals with textual data, which is inherently more complex than structured numerical data.
    - Standard ML metrics (e.g., accuracy, precision, recall) are designed for numerical predictions and may not directly apply to text.
2. **Variable-Length Sequences**
    - NLP tasks often involve variable-length sequences (sentences, paragraphs, documents).
    - Metrics like accuracy assume fixed-length predictions, which doesn't align with NLP's dynamic nature.
3. **Semantic Understanding**
    - NLP models aim to understand meaning and context in text.
    - Standard metrics don't capture semantic nuances, sentiment, or coherence.

The metrics used to evaluate this model are -

- BLEU Score
- ROUGE Score

4.1 BLEU (Bilingual Evaluation Understudy)

A quantitative indicator called Bilingual Evaluation Understudy (BLEU) is used to rate the quality of text produced by machines. BLEU, which was first created for machine translation, is now a popular tool for many different Natural Language Processing (NLP) applications. In particular, it tackles the requirement for a trustworthy method to gauge how effectively a system performs in picture-related natural language processing (NLP) tasks including chatbots, question answering, and image summarization. In order to do this, BLEU compares the text produced by the machine to several reference texts that were written by humans. Several factors are taken into account while calculating a BLEU score. The word overlap between the generated text and the references is measured by precision. By limiting the amount of matching words in the output to the maximum discovered in each reference, clipped precision overcomes problems like repetition and makes use of the existence of numerous references. To ensure a fair comparison between outputs of different lengths, a shortness penalty is finally given to account for scenarios when the generated text is shorter than the references. The brevity penalty term is included in the final BLEU score, which is calculated by a geometric mean combining clipped n-gram precisions (usually up to 4-grams). Despite being a widely used metric, BLEU has many drawbacks, such as its incapacity to accurately represent the semantic meaning of text. However, BLEU is still a useful metric for assessing NLP tasks that require producing text of human caliber.

4.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE stands out as a suite of metrics specifically designed to evaluate the quality of automatically generated summaries. Its primary focus lies in assessing how well a Natural Language Processing (NLP) model can grasp critical information and translate it into summaries resembling human-written outlines. This evaluation becomes crucial in tasks like automatic news summarization or machine translation output condensation.

ROUGE achieves this assessment through a variety of metrics, each targeting a different aspect of similarity between the machine-generated summary and the reference summaries crafted by human experts. ROUGE-N, for instance, measures the overlap between n-grams (sequences of n words) in both the output and reference summaries.

This can range from unigrams (single words) to bigrams (two-word sequences) and beyond.

On the other hand, ROUGE-L focuses on identifying the longest common subsequences – the longest stretches of words that appear in the same order in both summaries. ROUGE-W delves deeper by assigning weights to n-grams, placing more emphasis on informative words that contribute more to the overall meaning. Finally, ROUGE-S incorporates the concept of skip-gram concurrence, allowing for more flexibility in matching n-grams even if they don't appear consecutively in the summaries.

By calculating these various ROUGE scores, researchers and developers can gain valuable insights into the effectiveness of NLP models at generating summaries that are not only factually accurate but also structurally similar to those written by humans.

4.3 BLEU Score

The table contains BLEU scores of 3 parameters over 4 sample sizes. It also contains weighted average for each parameter. The weighted average of these parameters range between 0.4 – 0.65.

No of Samples	50	250	500	1000	Weighted average
Payee Name	0.61	0.57	0.64	0.61	0.61
Amount in words	0.42	0.45	0.41	0.43	0.42
Amount in fig.	0.72	0.53	0.58	0.55	0.56

Table 4.1: BLEU score values

4.4 Recall Score

The table contains Recall scores of 3 parameters over 4 sample sizes. It also contains weighted average for each parameter. The weighted average of these parameters range between 0.35 – 0.55

No of Samples	50	250	500	1000	Weighted average
Payee Name	0.48	0.45	0.53	0.50	0.50
Amount in words	0.49	0.50	0.54	0.52	0.52
Amount in fig.	0.66	0.35	0.38	0.37	0.37

TABLE 4.2: Recall score values

4.5 F1 Score

The table contains F1 scores of 3 parameters over 4 sample sizes. It also contains weighted average for each parameter. The weighted average of these parameters range between 0.40 – 0.65

No of Samples	50	250	500	1000	Weighted average
Payee Name	0.46	0.40	0.48	0.46	0.45
Amount in words	0.59	0.62	0.66	0.61	0.62
Amount in fig.	0.67	0.39	0.41	0.40	0.40

TABLE 4.3: F1 score values

4.6 Visualizations

The provided graphs illustrate the performance metrics for various parameters. Specifically:

Figure 4.1 displays the BLEU score for the payee name across a sample size ranging from 50 to 1000. **Figure 4.2** represents the BLEU score for the amount in words within the same sample range. **Figure 4.3** showcases the BLEU score for the amount in figures. **Figure 4.4** combines BLEU scores for all parameters (payee name, amount in words, and amount in figures). **Figure 4.5** focuses on the Recall score for the payee name. **Figure 4.6** presents the Recall score for the amount in words. **Figure 4.7** depicts Recall scores across all parameters. Lastly, **Figure 4.8** highlights the F1 score for the payee name, while **Figure 4.9** shows F1 scores for all parameters.

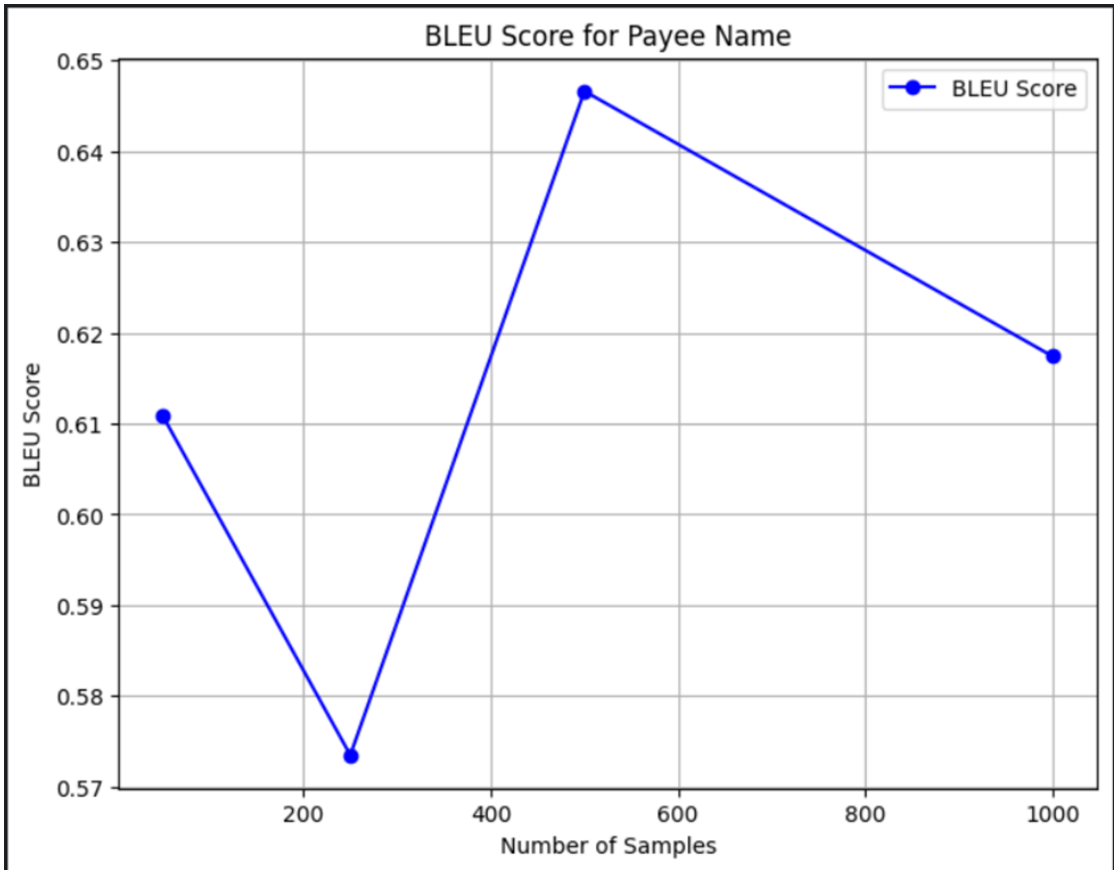


FIGURE 4.1: BLEU scores for payee name

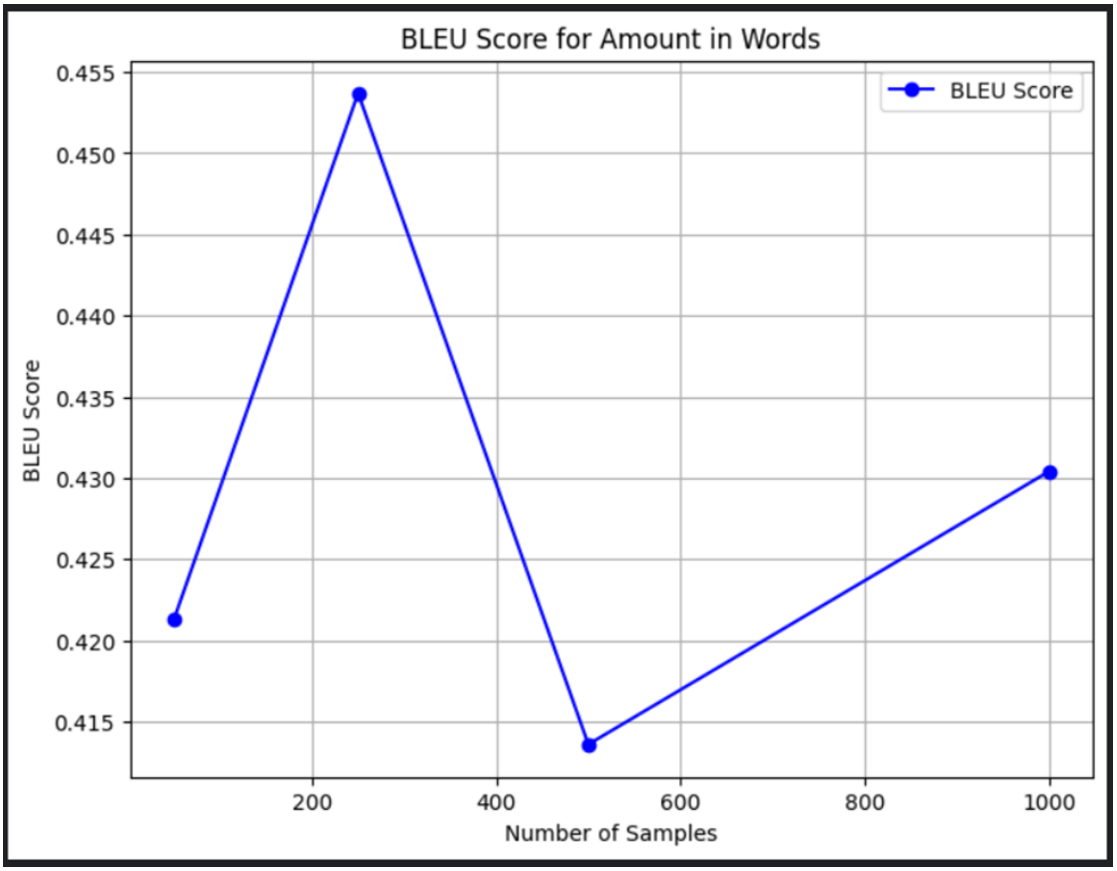


FIGURE 4.2: BLEU scores for amount in words

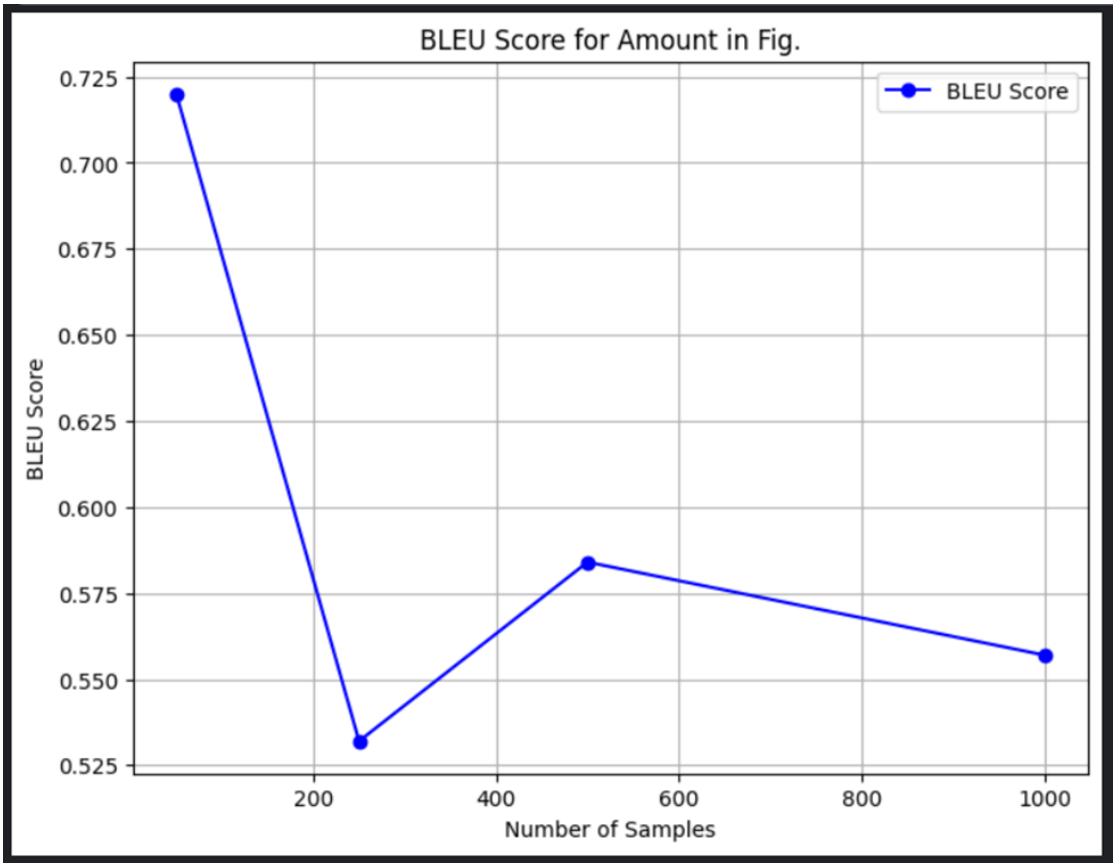


FIGURE 4.3: BLEU scores for amount in fig.

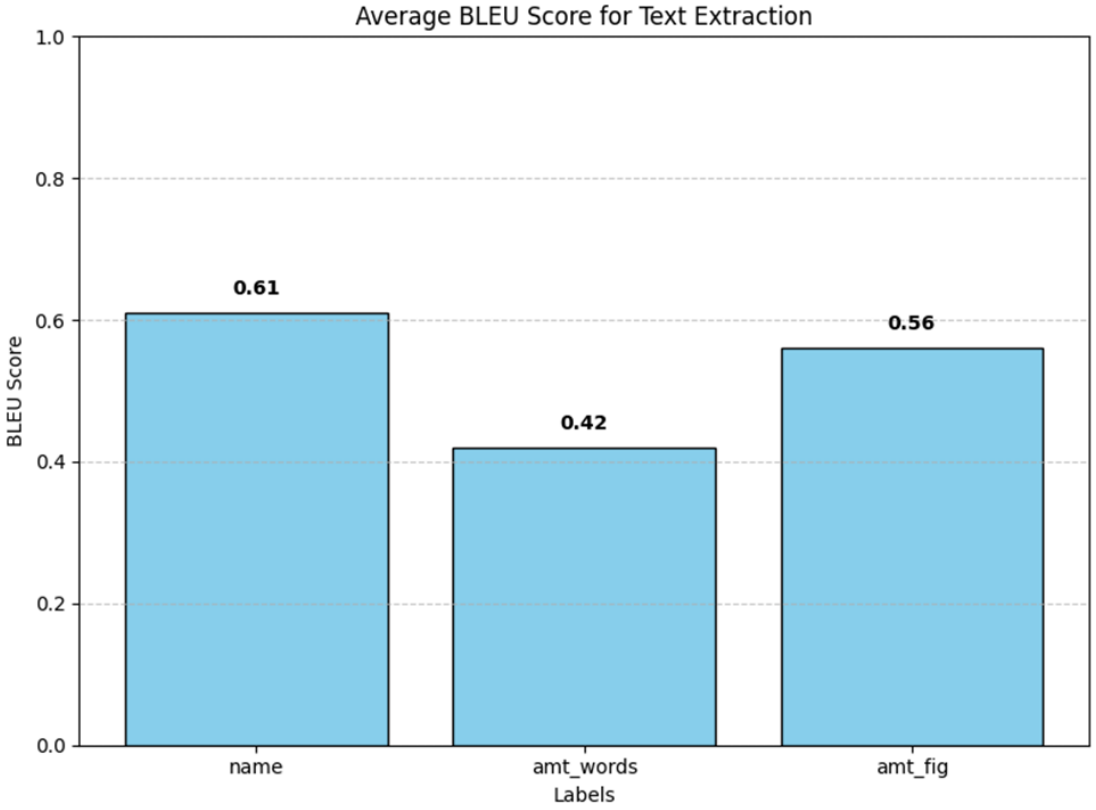


FIGURE 4.4: BLEU scores for all parameters

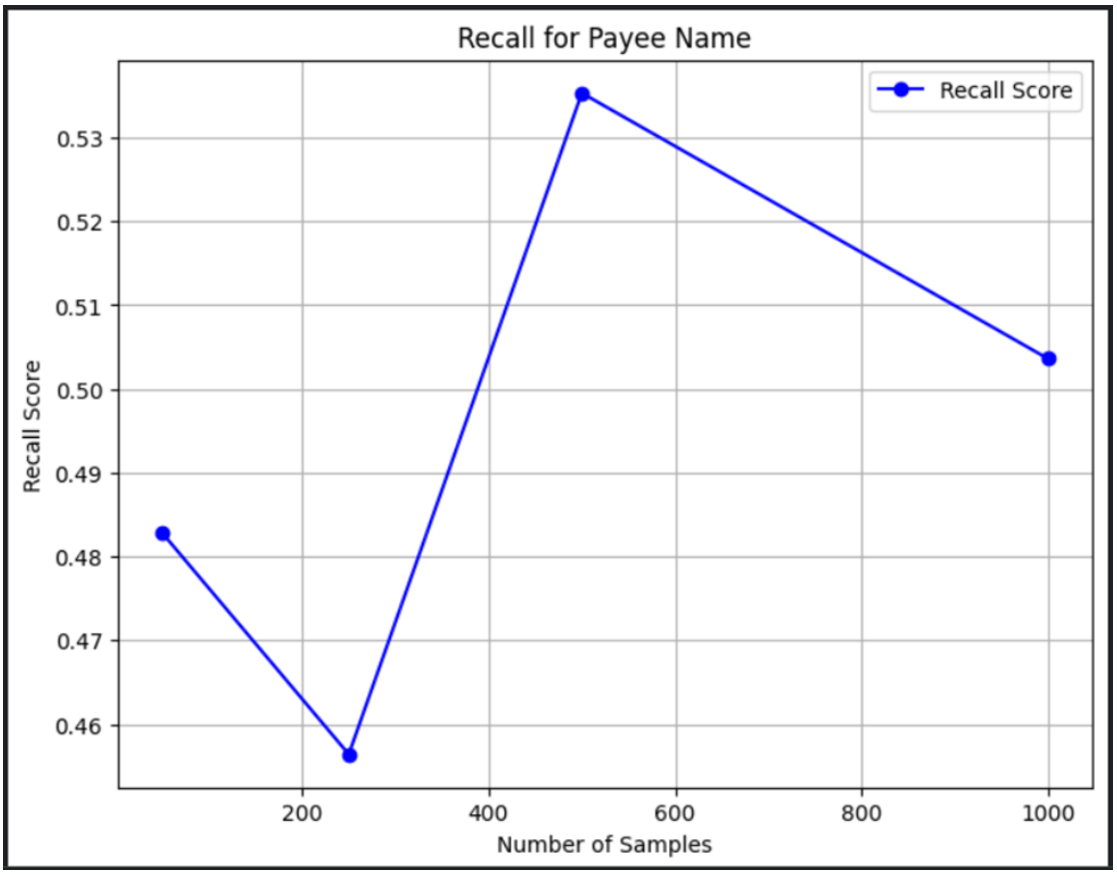


FIGURE 4.5: Recall scores for payee name

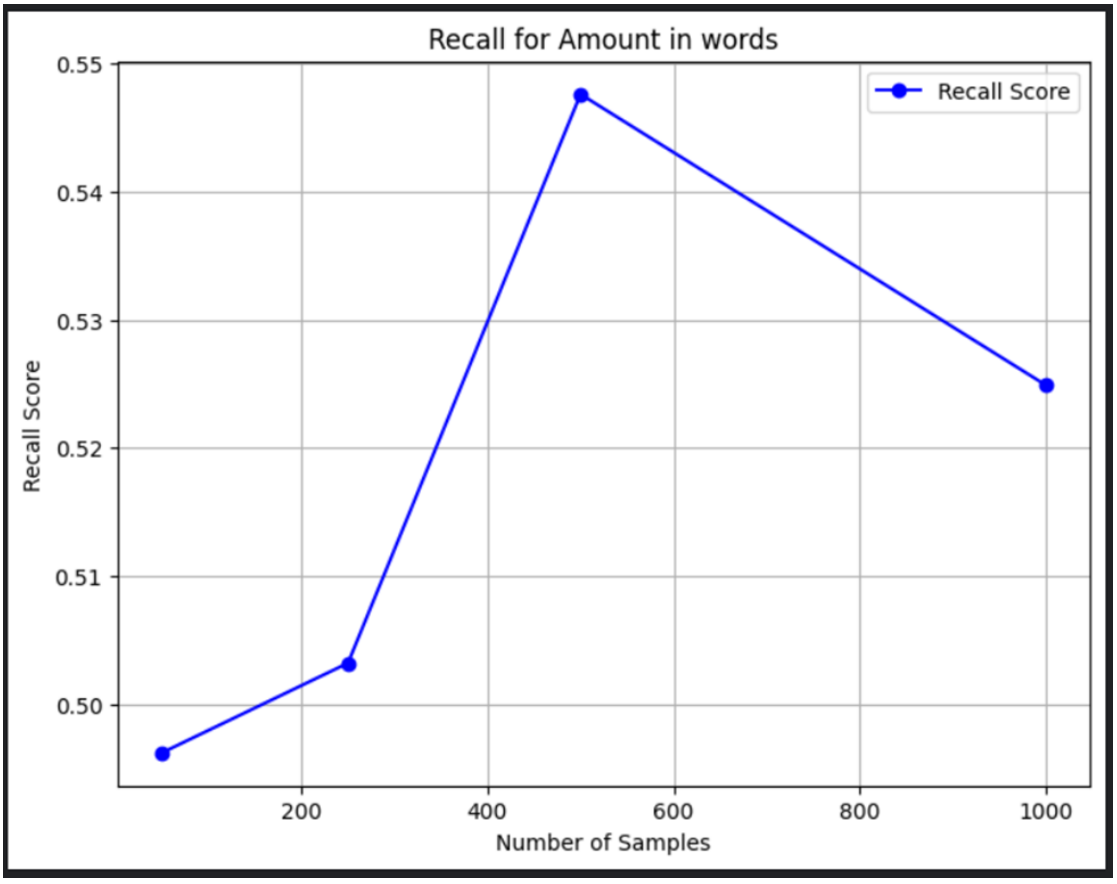


FIGURE 4.6: Recall scores for amount in words

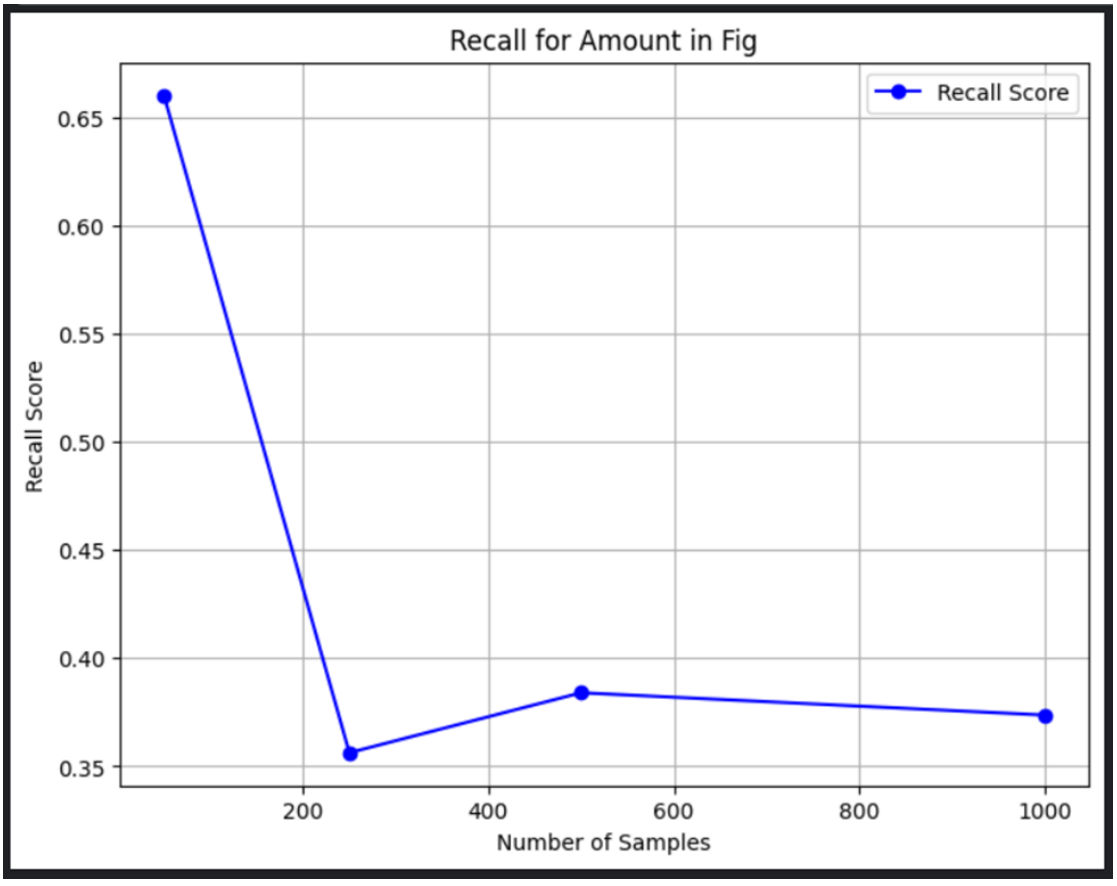


FIGURE 4.7: Recall scores for amount in figure

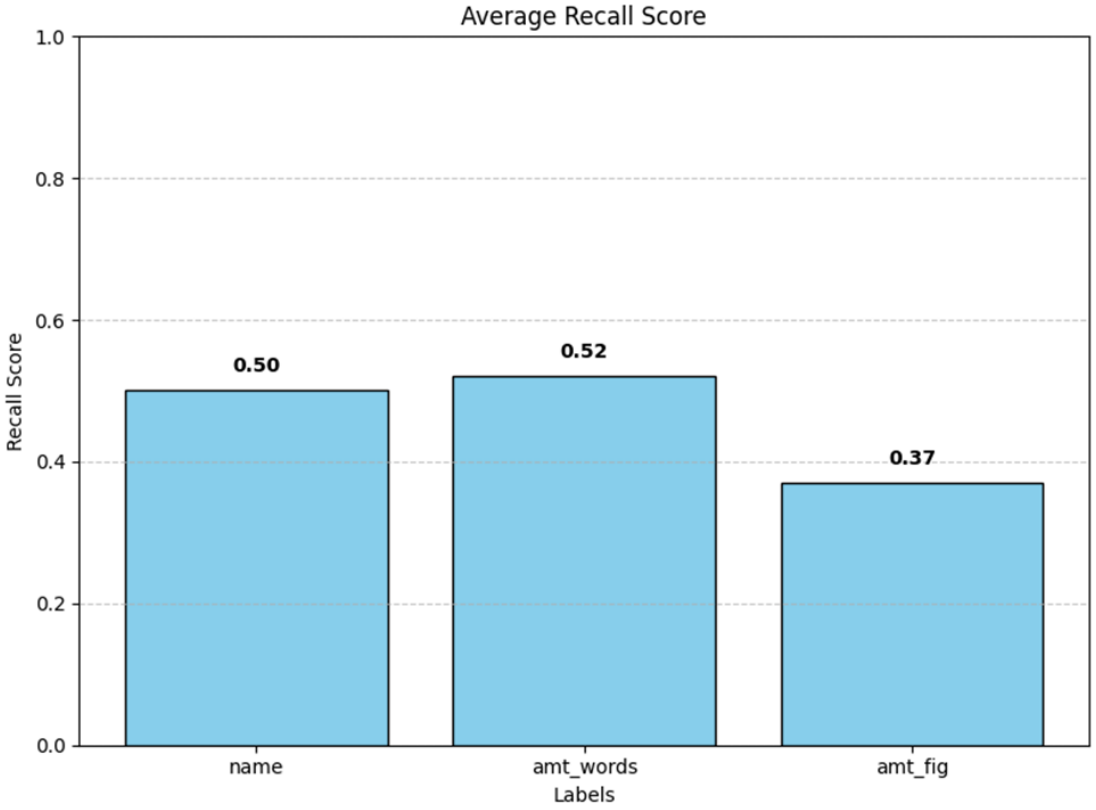


FIGURE 4.8: Recall scores for all parameters

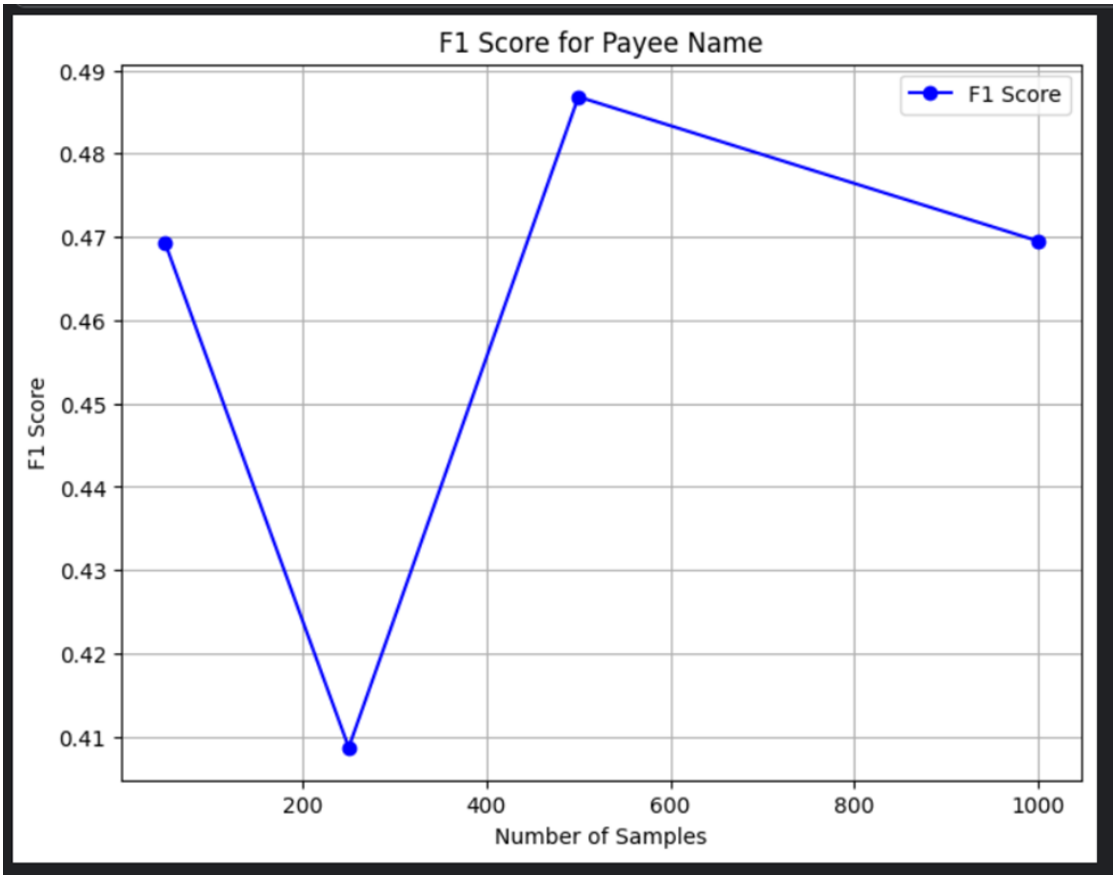


FIGURE 4.9: F1 scores for payee name

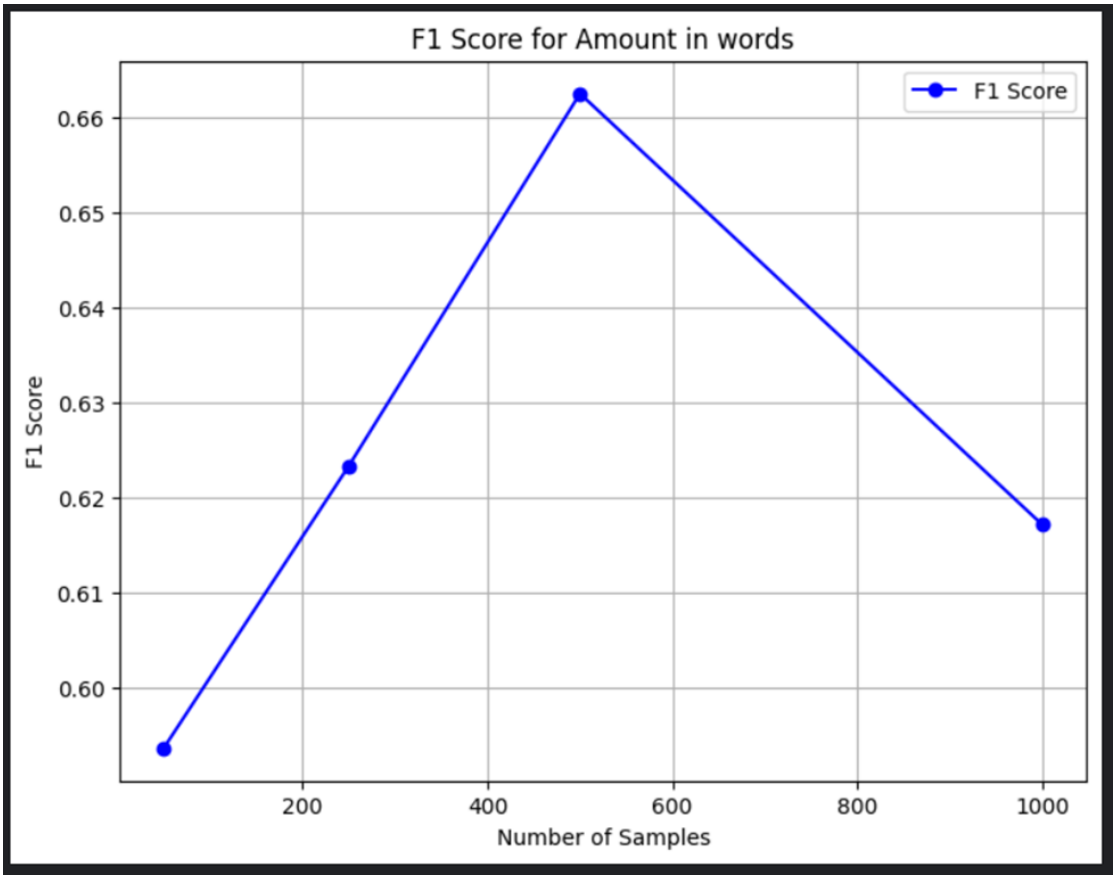


FIGURE 4.10: F1 scores for amount in words

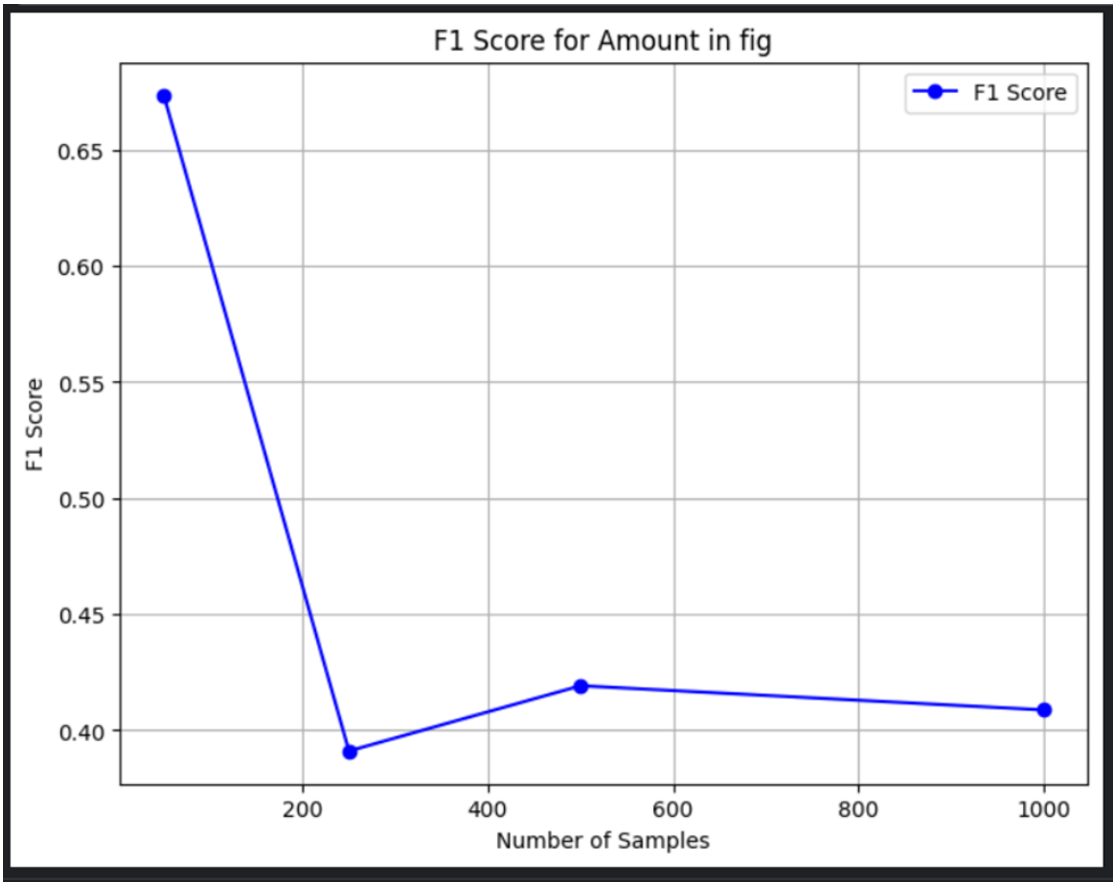


FIGURE 4.11: F1 scores for amount in figure

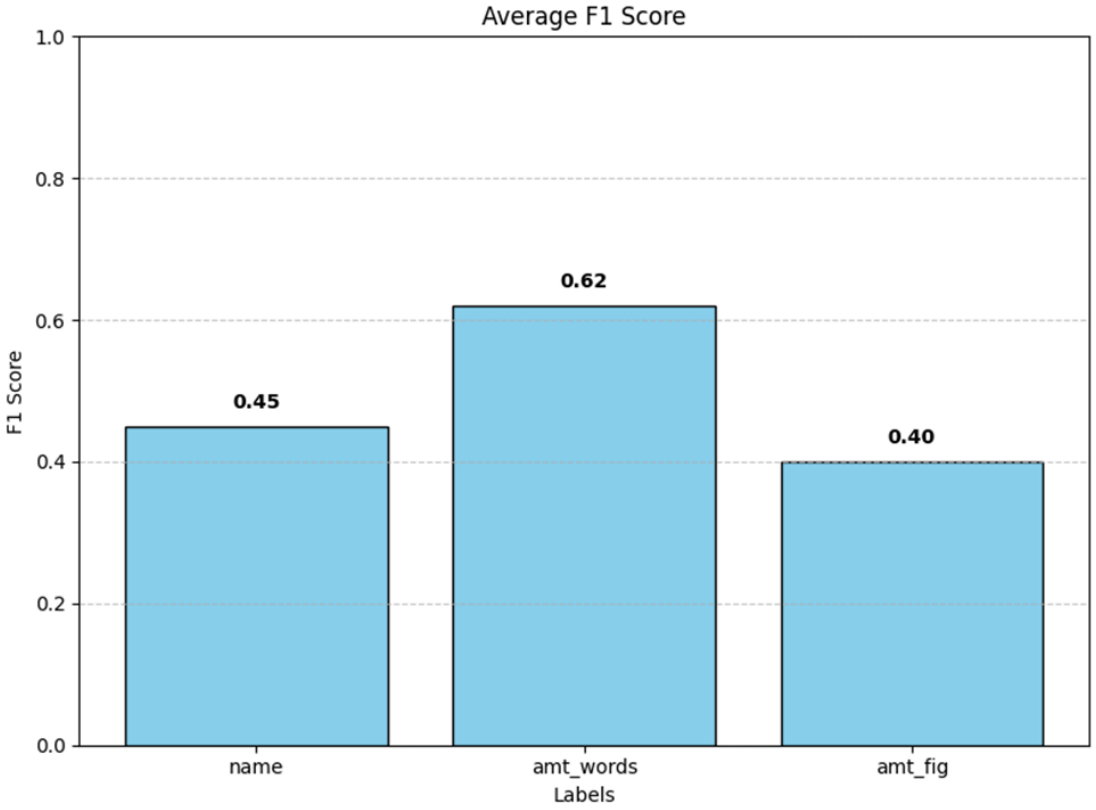


FIGURE 4.12: F1 scores for all parameters

Chapter 5

Conclusion and Future Scope

Donut offers a revolutionary end-to-end framework that transforms the interpretation of visual documents. Donut immediately analyzes document photos and extracts structured information, in contrast to conventional approaches which rely on optical character recognition (OCR) as an intermediate step. This novel method makes it unnecessary to have different OCR engines, which streamlines training and increases productivity. Moreover, during training, Donut generates synthetic document images using a special tool known as the Synthetic Document Image Generator (SynthDoG). As a result, the framework is more flexible and less dependent on enormous volumes of actual document data. Donut can handle documents in many languages and circumstances because its architecture is built for multilingual support. Donut has proven through extensive testing to perform better on datasets that are available to the public as well as on collections from the private sector. Furthermore, it is a very appealing option for practical uses due to its affordability. Although the pre-training target can still be improved, Donut's adaptability makes it a valuable tool for upcoming developments in visual document interpretation in a variety of activities and domains.

Chapter 6

Appendix

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)


import pkg_resources
import num2words
from symspellpy import SymSpell
from word2number import w2n

from dateutil import relativedelta
from datetime import datetime
import random
import re

import torch
import re
import pandas as pd
device = "cuda" if torch.cuda.is_available() else "cpu"

from sklearn.metrics import accuracy_score, precision_score, recall_score,
import nltk
from nltk.translate.bleu_score import sentence_bleu
import itertools

def BLEU(original_names, extracted_names):
```

```
    # Tokenize and preprocess names
    original_tokens = [nltk.word_tokenize(name.lower()) for name in origina
    extracted_tokens = [nltk.word_tokenize(name.lower()) for name in extrac

    # Initialize BLEU score
    bleu_score = 0

    # Compute BLEU score for each pair of original and extracted names
    for original, extracted in zip(original_tokens, extracted_tokens):
        # Compute BLEU score for the current pair
        current_score = sentence_bleu([original], extracted)
        # Add to the total BLEU score
        bleu_score += current_score

    # Calculate the average BLEU score
    average_bleu_score = bleu_score / len(original_tokens)

    average_bleu_score = round(average_bleu_score, 4)

    return average_bleu_score

def _split_into_words(sentences):
    return list(itertools.chain(*[_.split("-") for _ in sentences]))

def _get_word_ngrams(n, sentences):
    assert len(sentences) > 0
    assert n > 0

    words = _split_into_words(sentences)
    return _get_ngrams(n, words)

def _get_ngrams(n, text):
    ngram_set = set()
    text_length = len(text)
    max_index_ngram_start = text_length - n
    for i in range(max_index_ngram_start + 1):
        ngram_set.add(tuple(text[i:i + n]))
    return ngram_set
```

```
def rouge_n(reference_sentences, evaluated_sentences, n=2):
    if len(evaluated_sentences) == 0 or len(reference_sentences) == 0:
        raise ValueError("Collections must contain at least 1 sentence.")

    evaluated_ngrams = _get_word_ngrams(n, evaluated_sentences)
    reference_ngrams = _get_word_ngrams(n, reference_sentences)
    reference_count = len(reference_ngrams)
    evaluated_count = len(evaluated_ngrams)

    overlapping_ngrams = evaluated_ngrams.intersection(reference_ngrams)
    overlapping_count = len(overlapping_ngrams)

    if evaluated_count == 0:
        precision = 0.0
    else:
        precision = overlapping_count / evaluated_count

    if reference_count == 0:
        recall = 0.0
    else:
        recall = overlapping_count / reference_count

    f1_score = 2.0 * ((precision * recall) / (precision + recall + 1e-8))

    # Round the values to 4 decimal places
    recall = round(recall, 4)
    f1_score = round(f1_score, 4)

    return recall, f1_score


count=0

pred_name = []
org_name=[]

pred_amt_words=[]
org_amt_words=[]
```

```
pred_amt_fig=[]
org_amt_fig=[]

pred_bank=[]
org_bank=[]

random_indices = random.sample(range(1, 10001), 50)

# Iterate over the random indices
for i in random_indices:
    count += 1
    image_path = f"/kaggle/input/cheque-images/yolo-x-image/yolo-x-image/{i
    print("Iteration Count:",count)

    payee_name,amt_in_words,amt_in_figures = parse_cheque_with_donut(image_

    pred_name.append(payee_name)
    pred_amt_words.append(amt_in_words)
    pred_amt_fig.append(amt_in_figures)
    #pred_bank.append(bank_name)

    org_name.append(df_name.iloc[i-1])
    org_amt_words.append(df_amt_words.iloc[i-1])
    org_amt_fig.append(df_amt_fig.iloc[i-1])
    #org_bank.append(df_bank.iloc[i-1])

print("BLEU Scores")

print("Payee name:", BLEU(org_name,pred_name))
print("Amount in words:", BLEU(org_amt_words,pred_amt_words))
print("Amount in fig:", BLEU(org_amt_fig,pred_amt_fig))
#print("Bank name:", BLEU(org_bank,pred_bank))

n_recall,n_f1 = rouge_n(org_name,pred_name)
aw_recall,aw_f1 = rouge_n(org_amt_words,pred_amt_words)
af_recall,af_f1 = rouge_n(org_amt_fig,pred_amt_fig)
#b_recall,b_f1 = rouge_n(org_bank,pred_bank)

print("Recall")
```

```
print("Payee name:", n_recall)
print("Amount in words:",aw_recall)
print("Amount in fig:",af_recall)
#print("Bank name:", b_recall)

print("F1 score")

print("Payee name:", n_f1)
print("Amount in words:",aw_f1)
print("Amount in fig:",af_f1)
#print("Bank name:", b_f1)
```

Bibliography

[1] Agrawal, Prateek, et al. "Automated bank cheque verification using image processing and deep learning methods." Multimedia Tools and Applications 80 (2021): 5319-5350.

[2] Jha, Mukesh, et al. "Automation of cheque transaction using deep learning and optical character recognition." 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, 2019.

[3] Thorat, Chhanam, et al. "A detailed review on text extraction using optical character recognition." ICT Analysis and Applications (2022): 719-728.

[4] Kaundilya, Chandni, Diksha Chawla, and Yatin Chopra. "Automated text extraction from images using OCR system." 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2019.

[5] Jiju, Alan, Shaun Tuscano, and Chetana Badgujar. "OCR text extraction." International Journal of Engineering and Management Research 11.2 (2021): 83-86.

[6] Sidhwa, Harshit, et al. "Text extraction from bills and invoices." 2018 international conference on advances in computing, communication control and networking (ICACCCN). IEEE, 2018.

[7] 7) Huang, Zheng, et al. "Icdar2019 competition on scanned receipt ocr and information extraction." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

[8] 8) Xu, Yang, et al. "Layoutlmv2: Multi-modal pre-training for visually-rich document understanding." arXiv preprint arXiv:2012.14740 (2020).

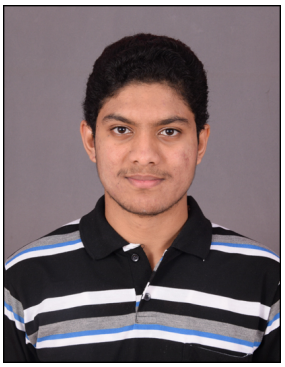
[9] Kim, Geewook, et al. "Donut: Document understanding transformer without ocr." arXiv preprint arXiv:2111.15664 7.15 (2021): 2.

[10] Bajrami, Merxhan, et al. "A Comprehensive Analysis of LayoutLM and Donut for Document Classification." (2023).

[11] Kim, Geewook, et al. "Ocr-free document understanding transformer." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.

[12] Kim, Geewook, et al. "On text localization in end-to-end OCR-Free document understanding transformer without text localization supervision." International Conference on Document Analysis and Recognition. Cham: Springer Nature Switzerland, 2023.

Biodata



Name: Darshan N Shenoy
Mobile No.: 8792461421
E-mail: darshanshenoy.n2020@vitstudent.ac.in
Permanent Address: Banashankari 3rd stage, Bengaluru-85