PDF PROOFING REPORT

Research Project: Pdf Proofing

Author: Sufyan Qadir

Table of Content

1. Abstract **2**

2. Introduction **2**

3. Problem Statement **2**

4. Methods and Tools **2**

4.1. Initial Approach 2

4.2. Translation Correction 2

4.3. Image Correction 3

5. Tests and Results **4**

6. Discussion **4**

7. Reference **4**

1. Abstract

Madcap Flare is used to create the PDF files for our documents. Despite the fact that we have translated PDF instruction scripts into numerous languages, issues nevertheless arise from time to time, such as language errors and missing images. Each document must be proofread, which takes time. This problem has to be automated using Python so that it can be examined more quickly and efficiently.

2. Introduction

The HTML files (in English) from Madcap Flare are used to create the pdf files. Simply put, PDF files are instructions that specify where each component should be placed on a screen or piece of paper. It frequently lacks logical organization, such as sentences or paragraphs, and is unable to adjust to changes in paper size. It takes some time to find these issues manually; each document takes about 10 minutes. Given that we translate documents into fifteen different languages and take into account human mistakes, proofreading would take around two hours and thirty minutes. **In order to save time and money, it would be beneficial to automate this task and hasten the procedure.**

3. Problem Statement

There are missing images and incorrect language translations in our translated documents.

4. Methods and Tools

4.1. Initial Approach

Possible Solution	A simple possible solution would be to read all the paragraphs and detect the language. ● Select directory ● Loop through all the htm files ● cycle through each paragraph. ● Detect the language of each paragraph ● If the language is not what is expected, show the file name, language, and text.
Tools	● Python (fasttext, os, re) ● Gitlab

4.2. Translation Correction

Possible Solution	Reading blocks of text and identifying the language is another option. ● Select file ● cycle through each block of text. ● Detect the language of each block ● If the language is not what is expected, show the file name, language, and text.
Tools	● Python (fasttext, os, re, pdfminer) ● Gitlab

4.3. Image Correction

Possible Solution	The number of images in the translated PDF file should be compared to the number of images in the original English PDF file. ● Select two files ● cycle through each and search for any images. ● Count them ● If the number of images is not the same for both files then flag the file.
Tools	● Python (pdfminer) ● Gitlab

5. Tests and Results

When copied into the input terminal, testing the file path was successful. Text can be detected fairly accurately if the proper language of the document is selected. The image counts of each file can be compared; however, before it can be used again, the imageDetector script should be improved.

6. Reference

- *Cavdar, Z. (n.d.). Fasttext-langdetect: 80x faster and 95% accurate language identification with fasttext [Python]. Retrieved 13 January 2023, from https://github.com/zafercavdar/fasttext-langdetect.git*
- *pnj. (2014, April 6). Answer to 'How to extract text and text coordinates from a PDF file?' Stack Overflow. https://stackoverflow.com/a/22898159*