PDF To Speech using TexParsing and Tokenization

[1]Anup A Saleem, [2]Rathila Ramesh, [3]Abhishek B.K ,[4]Niveditha.S
[1]Student, [2]Student, [3]Student, [4]Asst.Professor
[1]Department of Computer Science,
[1]SRM Institute of Science and Technology, Chennai, India

Abstract : There are many areas in computer science which have not used the resources around them optimally to their fullest potential. Such is the case of PDF to speech conversion in today's perspective. PDF to speech conversion is a problem which can be solved by dividing it in two halves. One half being PDF text extraction and the second one being text to speech conversion. PDF text extraction can be done using TexParsers, Although all the PDF files cannot be handled by just one Parser as the contents of a file may vary from texts to photos and other media. Also in text to speech conversion there are various locales for different regions for the English language, which can help increase the understand-ability of the converted speech. Using the fore mentioned tools and various other tools to handle exceptional cases, this project aims to fuse the functionality of these tools and produce an accurate PDF to Speech conversion system. We have compared our systems with other systems based on various criteria, such as, Processing speed, Accuracy and CPU usage. In accordance with processing speed the tool which we used, computed the same test cases in just 25% of the time used by the nearest fastest tool tested. In terms of CPU usage, our system uses around 10% less CPU Space than the next least space occupying tool tested. The OCR tool we integrated was compared to be the most accurate among two other competitors with the accuracy rate of 80%

IndexTerms- PDF, TexParsers, Text to Speech

I. INTRODUCTION

Technological advancement happens every day and one such advancement is text to speech. There are many systems that converts text to speech but the question is how efficient it is and can it convert text stored in any format to speech. Data can be stored in various formats like .TXT or PDF or as an image file. Our aim is to present a tool that would treat data stored in these formats differently to extract the text and convert it to speech . If the data is stored as .TXT file then we will extract the text using java file reader and if the text is stored as a PDF then we will extract it using TexParser. Now if there is an image which contains text then we extract it using OCR(Optical Character Recognition). The extracted text will be fed to the Google TTS system which will produce the output in the form of speech. The Google TTS system produces sound that resembles human speech. To make the speech more understandable we provide different locales from which the user can choose. Along with that the user will also have the freedom of choosing the gender of the voice reading the text ,its speed and its pitch. We have chosen android platform for the application development since it is the widely used mobile platform which has scope for future developments.

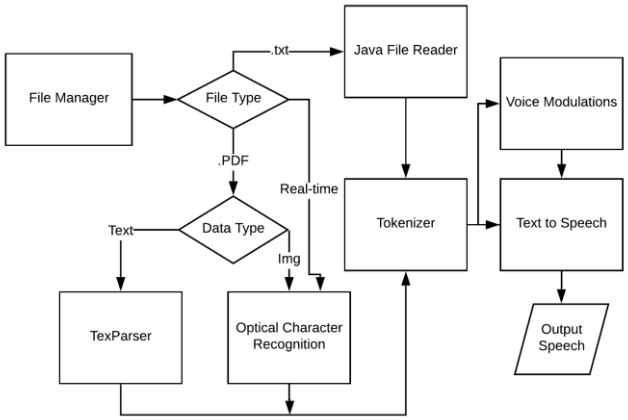

Figure 1 - System Architecture

II. LITERATURE REVIEW

In this section we will give a Literature review on the existing use of technology in the fields of text extraction from PDF and Text to speech. Earlier the extraction of text out of documents other than .txt was a tedious task and the results were not accurate. Using OCR for text extraction from file types such as PDF yielded in less accurate outputs and sequential recognition of word blocks was a tedious task.

R.Shantha Selvi Kumari etc.[7] proposed a system which recognized text using OCR inside documents and store the output text in a .txt file. This system basically recognized text blocks and tried to sequentially deliver them inside an output file. It was also noted that the system worked only for text with constant font styles which were loaded into the system manually. The text extraction from PDF is not possible using this system.

Ramiah, S. etc.[8] also developed a system which could recognize text in real-time with the use of OCR in Android phones. This system used a much more advanced tool than the system proposed in [7]. Although, this tool is not used to recognize texts out of any kind of document.

So,it is possible to extract text out of PDF using OCR, and that is by converting the PDF pages into individual image files then reading the image files using OCR. But, this type of working won't be optimal for a system for various reasons such as the storage of image files which will be yielded by conversion of PDF pages. Not only that, the accuracy of OCR systems not reliable in current scenario. Therefore, there is a need of an alternative which extracts text out of documents in a much more efficient manner.

Hannah Bast and Claudius Korzen[1], defined such an alternative for PDF text extraction. The system they proposed consisted of parsing the Tex files in which PDF files are formed in raw form. The sole purpose of this system was accuracy so they used various evaluation algorithms such as doc-diff, rearr-diff and word-diff to evaluate the output file and making corrections in order to achieve higher accuracy in output file. This was a successful attempt at extracting text out of PDF with high accuracy. Although this system could only read the PDF file in raw text form therefore, the PDF consisting of image data cannot be processed by this system.

Now that we have looked into PDF text extraction lets move on to the next phase which is Text to speech. The extracted text out of the PDF will be converted to speech. This will give us a system which will convert PDF documents into speech. Text to speech technology emerged in the early 2000s and it has evolved a lot since then.

Alias, F etc.[9] presented a paper in 2008 which introduced a new advanced way of converting Text to Speech. They developed a system which tokenizes the text into recognizable words by the use of natural language processing.

Hussain Rangoonwala etc. [4], proposed a better algorithm which even breaks down the words down to syllables and the generated the speech is even able to imitate voices.

In the current scenario Text to speech conversion is a well researched field and advanced logical approach towards it has led to creation of highly accurate engines which converts speech without error.

To use these system there is a requirement of GUI for human interaction, P. M., Santra etc.[6], developed a system which provided GUI for enabling human interaction with the Text to Speech engine. The proposed system was a desktop application which requires a user to copy and paste the text into a text area in order for the text to get converted into speech. Moreover, there was an option for the user to select any .txt files from local storage for speech conversion.This system required a lot of human

interaction than needed optimally. Also there is no provision given for reading any other file types other than .txt. This system also didn't have any variation in the voice of the speech generated.

Deepshikha Mahanta etc.[5], developed a system which provided an option for converted speech to be in Indian English. This was done by recording various words said by a human in Indian accent and training the system to recognize each recording using the spelling of the words.

These systems didn't explore the automation of text extraction out of document. Also, there are technologies involved which can be applied in a much vast and optimal manner than presented. Using the technologies present in the current scenario. A system can be implemented which directly converts the speech from a PDF file into speech form. This can be done by using Google Text to speech engine for speech conversion, iText PDF library and Google Vision API for PDF text extraction. By using OCR and TexParsing together an optimal system can be built which divides responsibilities among these platforms optimally in order to efficiently convert PDF text to speech.The use of mentioned tools will enable us to produce speech in various different locales which will give the converted speech better understand-ability in all regions.

III. THE COMBINATION OF VARIOUS PLATFORMS

Google Text to Speech engine

Google has provided an open source text to speech API, which works with an astounding accuracy. This engine uses an algorithm known as syllabification. The major advantage of this algorithm is that it parses the words down to the point of syllables, by doing this even words which don't have meanings also are recognized by the engine.

I-text PDF parser

For extracting text out of PDF files which are stored in raw text format a stable and quick TexParser is required. Itext provides a library which is perfect for the mentioned use case. TexParsing is the benchmark of PDF text extraction. It uses various mathematical algorithms to recognize the string stored as text inside the PDF. These algorithms are known as rear-diff, doc-diff, word-diff. The accuracy of these systems are very reliable.

Google Vision API

Although OCR is considered as a developing technology and an outdated one in terms of PDF text extraction due to its low accuracy, it has few advantages over the TexParsing. Such as, TexParsing can only parse through the raw string form of data, if the data is contained in an image format then TexParser will not be able to read the data. This is where OCR plays an important role as it complements the working of TexParser.The Vision API uses Google's internal machine learning framework which uses image training sets to automatically develop image classification algorithms.

IV. PROPOSED SYSTEM

There are many algorithms working simultaneously to achieve a uniform task, which is PDF to speech conversion.

Syllabification

First lets discuss the algorithm which enables the conversion of text to speech in an accurate manner. For this we need to look at typical syllable model in Fig 2.

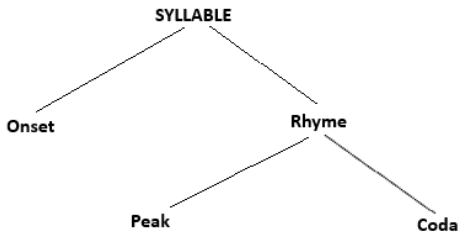

Figure 2 - Syllable Model

A syllable can be broken down into two parts -:

- Onset- This consists of Consonants or consonant clusters, its obligatory in some languages and optional or even prohibited in some languages.
- Rhyme- Rhyme is broken down further into two parts -:
  1. Peak- It consists of a vowel or syllabic consonant, its obligatory in most languages.
  2. Coda- It consists of consonant, its optional in some languages and highly restricted in many.

This model of syllable can be used to depict syllables in all languages. Different languages have different rules for their syllables but the basic model remains the same.If C stands for consonant and V stands for vowel then in English language the syllables can be formed using just V, but other syllables can also be formed such as CV, VC, CVC, VCV.Syllables can be classified into monosyllable, disyllable, trisyllable and polysyllable. So once the language and its rules are set as a key, it recognizes syllables to the most minimal point .

Rearr-diff, doc-diff and word-diff

Parsing Tex files is a very tedious task and it requires few sequential heuristic algorithms to work together in order for the parsing to be efficient

First lets look into how the system interprets the file and evaluates it in a mathematical form. After the parsing is done the extracted document can be called D_{out} and the actual document can be called ground truth document D_{gnd}. Its important to recognize the difference between D_{out} and D_{gnd} exactly in order to produce an output file with no difference from the ground truth file. So, for evaluation of the files mathematically there are three groups of criteria presented :-

- **Newline Differences** measures the quality of paragraph boundary detection it is broken down into -:
  - NL+: the number of extra newlines in the output.
  - NL−: the number of missing newlines in the output
- **Paragraph Differences** measures the quality of difference in body and non-body text blocks, also the reading order. They are broken down into:
  - P+ : the number of extra paragraphs in the output
  - P− : the number of missing paragraphs in the output
  - P : the number of rearranged paragraphs in the output
- **Word Differences** measures the quality of difference in individual words and their boundaries. They are broken down into:
  - W+ : the number of extra words in the output
  - W− : the number of missing words in the output
  - W : the number of misspelled words in the output

These criteria can be determined by the system so using these we can evaluate the accuracy of the output file and use it to repair the output files into more accurate form.

As shown in Equation. 1, variable Z which denotes the overall distinction in output file and ground truth file.

$$Z = ((NL+) + (NL-)) + c.((P+) + (P-) + (P)) + ((W+) + (W-) + (W-))$$

Equation 1 -Mathematical Evaluation of output document

Now, a heuristic algorithm known as doc-diff compares the documents word by word. It converts the document's words into string array, lets consider them to be Wo and Wg and then compare them one by one. Then the doc-diff algorithm develops phrases of different types -:

- **Common phrase** ([word 1, …, word i]): a sequence of i consecutive words which are common to Wo and Wg .
- **Differing phrase** ([word 1, …, word j], [word 1, …, word k]): a sequence of j extra words, which occur in Wg but not in Wg ; and of k missing words, which occur in Wg but not in Wo.
- **Rearranged phrase** ([word 1, …, word m], [word 1, …, word n]): a sequence of m words in Wo and n words in Wg , which are (almost) equal ($m \approx n$), but their positions in Wo and Wg are different.

Now these phrases are computed by two other algorithms knows as rearr-diff and word-diff. First word-diff computes the phrases in paragraphs. Rearr-diff computes the rearranged phrases and identifies similar spaces between missing and extra words and then wraps these phrases with related differing phrases.

Tokenization of extracted data

After extracting the data in string format we need to convert these strings into sentences in order to feed into Text to speech engine. This enables to maintain an optimal working load on TTS engine.

As shown in Fig 3. The tokenizer creates a sub-string of the string with indices I and j as the range then the token waits to be loaded into the text to speech engine. As soon as the instance of text to speech stops talking the token is loaded into the TTS engine. While this is happening simultaneously a call is given to a method sentenceSwitcher() which change the indices I and j for the next sentence this function calls the tokenizer after the end of its operation and the cycle continues until the whole document is finished reading.

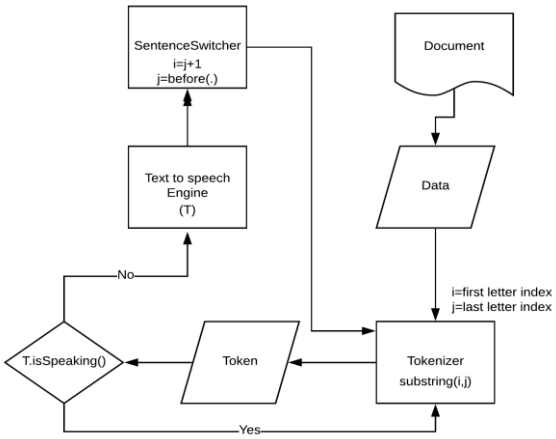

Figure 3 - Tokenization of extracted string

Design and implementation

A file manager module was implemented to handle and store the resources used by the application. This helps determine whether the file in processing is of what type and then sends the file to a class which has the appropriate combo of tools to process the data consisted inside the file into speech.

Voice modulations module enables the system to produce speech in various different voices which are computed by the combination of pitch,speech rate, locale and gender of the voice. These modulations are done by creating a Input GUI for changing the parameters of the Text to speech instance and restarting it. The GUI is shown in Fig 4.

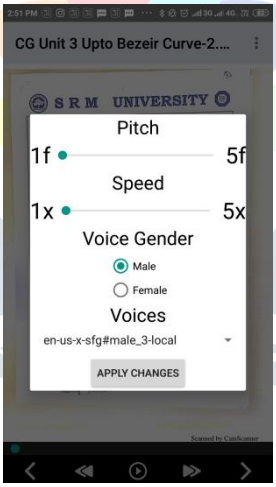

Figure 4 - GUI for voice modulation

Image data stored inside PDF were extracted using I-text library by extracting the encoded RAW format and then converting it back into bitmap.Also, a real-time module was added which enabled users to convert physically present text into speech in real-time.

By using, the mentioned algorithms and techniques we develop a system which serves the purpose of PDF text to speech with high accuracy. This system automatically extracts data out of PDF files and feeds it into Text to Speech engines. This requires minimal human interaction and most of it is automated.The converted speech can be configured into various types by altering the locale, speed, pitch and gender of the voice.This helps in increasing the understand-ability of the produced speech. Text extraction

is done using both the technologies in the scenario in such a manner both of them complement each other and have divided work load.This project has scope for helping to create a document reader for visually impaired people.

V. EXPERIMENT RESULT

Image recognition and classification algorithms

Google cloud vision API uses Google's internal framework for machine learning to create systems which automatically develops algorithms for image classification. Some image recognition systems are trained already by the creators but in future the system will incorporate deep learning systems which collects and analyzes the data and trains the system automatically to develop image recognition algorithms.

Right now, its a developing technology and has a lot of scope for improvement, but this is the best version of OCR in the current technological scenario. As shown below in Table 1 the testing of various providers yielded these results, which clearly shows Google vision API is the best available option and Is promising for the future.

Provider	Correct	Wrong	No Result	Precision	Recall
Microsoft Cognitive Services	142	76	283	65%	44%
Google Cloud Vision	322	80	99	80%	80%
AWS rekognition	58	213	230	21%	54%

Precision=Correct/(Correct+Wrong)

Recall=(Correct+Wrong)/Total

Table 1 - A comparison between OCR providers

Existing TexParsing tools

We compared some existing TexParsing PDF text extraction tools namely Apache FOP, Apache PDFBox, iText, JPDFWriter

As shown in Table 2 in existing tools Itext performs much faster in generating 2000 homogeneous PDF files.

TexParsers	Time(in ms)
Apache PDFBox	061781
Apache FOP	094424
iText	015537
JPDFWriter	123236

Table 2 - Comparison of PDF Text Extraction tools in terms of speed

Also, As shown in Table. 3, iText is optimized in such a manner that it requires much lower CPU usage than other tools.

TexParsers	CPU Usage
Apache PDFBox	58%
Apache FOP	61%
iText	38%
JPDFWriter	47%

Table 3 - Comparison of PDF Text Extraction tools in terms of CPU Usage

VI. CONCLUSION AND FUTURE SCOPE

This paper achieved, text extraction out of PDF using TexParsers with high accuracy, the extracted text was also successfully converted to speech. Earlier, The extraction of Text out of PDF had a big run-time and wasn't optimal for usage. We solved this issue by processing one page at a time of the PDF file. The voice modulations were included and working perfectly. The images were extracted out of the PDF files and converted to speech successfully.A real-time enables the user to extract physically present text in real-time There is a lot of scope for improvement for OCR processing.The extraction of text using OCR has a lot of scope for improvement as the strings are extracted but the words are not sequentially recognized. In future an algorithm can be created which arranges the OCR output data in a sequential manner. In future this paper paves way for mobile applications which will be useful to visually impaired people if the GUI can be optimized for the visually impaired users.

REFERENCES

[1] Hannah Bast and Claudius Korzen. 2017. A Benchmark and Evaluation for Text Extraction from PDF. In Proceedings of Joint Conference On Digital Libraries, Toronto, Ontario, Canada, June 2017 (JCDL'17), 10 pages. DOI: 10.1145/nnnnnnn.nnnnnnn

[2] Jisha Gopinath1, Aravind S2, Pooja Chandran3, Saranya S S4"Text to Speech Conversion System using OCR",International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 5, Issue 1, January 2015) 389

[3] Najib Ali Mohamed Isheawy And Habibul Hasan "Optical Character Recognition (OCR) System" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. II (Mar – Apr. 2015), PP 22-26 www.iosrjournals.org DOI: 10.9790/0661-17222226 www.iosrjournals.org

[4] Hussain Rangoonwala, Vishal Kaushik, P Mohith and Dhanalakshmi Samiappan "TEXT TO SPEECH CONVERSION MODULE" International Journal of Pure and Applied Mathematics Volume 115 No. 6 2017, 389-395 ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)url: http://www.ijpam.eu Special Issue

[5] Deepshikha Mahanta, Bidisha Sharma, Priyankoo Sarmah, S R Mahadeva Prasanna* "Text to Speech Synthesis System in Indian English" Indian Institute of Technology Guwahati, India {deepshikha.m, s.bidisha, prasanna, priyankoo}@iitg.ernet.in 978-1-5090-2597-8/16/$31.00 c 2016 IEEE

[6] P. M., Santra, S., Bhowmick, S., Paul, A., Chatterjee, P., & Deyasi, A. (2018). *Development of GUI for Text-to-Speech Recognition using Natural Language Processing. 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech).*doi:10.1109/iementech.2018.8465238

[7] R.Shantha Selvi Kumari, R. Sangeetha.Optical Character Recognition for Document and Newspaper,International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.20 (2015)© Research India Publications; http://www.ripublication.com/ijaer.htm

[8] Ramiah, S., Liong, T. Y., & Jayabalan, M. (2015). *Detecting text based image with optical character recognition for English translation and speech using Android. 2015 IEEE Student Conference on Research and Development (SCOReD).*

[9] Alias, F., Sevillano, X., Socoro, J. C., & Gonzalvo, X. (2008). Towards High-Quality Next-Generation Text-to-Speech Synthesis: A Multidomain Approach by Automatic Domain Classification. IEEE Transactions on Audio, Speech, and Language Processing, 16(7), 1340–1354. doi:10.1109/tasl.2008.925145