**Group Member Details (Name – ID):**

* Kriti Bidani – 0775260

A Systematic Approach for Automated Extraction of Data from Clinical Reports using NLP and Artificial Intelligence Predictive Analytics Tools

# Healthcare Industry Needs

Healthcare industry accumulates enormous data, which is continuously growing, and is largely semi-structured and/or unstructured in nature. This leads to certain potential challenges for the healthcare stakeholders in utilising it efficiently. Medical data that is collected is in various forms

images, prescriptions, signals, discharge summaries, etc. and this huge unstructured data as per the National Center for Biotechnology Information (NCBI) comprises of about 80% of the entire medical records, which in its raw form cannot be efficiently utilised to extract information.

including but not limited to text reports,

The medical knowledge has been drastically increased over the recent years and this has resulted in additional diagnostic and treatment possibilities. But at the same time, it cannot be denied that every patient has their own complex medical needs and for this it is essential for healthcare providers to obtain the entire medical history of the patient in a format which could help in gaining valuable health insights. It is absolutely necessary at the point of care (POC) to gather accurate, complete, and most up-to-date information about patient’s health so that finest quality and safe care could be delivered.

The possible solution to maintain the patients’ data over time is through creating Electronic Health Records (EHR), which has the potential to capture the entire history of the patient’s health. The electronic version works by digitizing the patients’ medical history and further making it consistent and easily accessible to the healthcare providers. Application of machine learning and artificial intelligence (AI) can be served as an important tool in leveraging text data in the form of clinical reports and can further help in better clinical decision making. These techniques can help healthcare provider to better understand the patents’ disease, and thereby effective care could be extended which can improve the outcome.

# NLP as a Solution

In healthcare industry, is being widely deployed to assist clinicians and thereby,

deep leaning

patients through multiple ways. In the past, human involvement was essential to utilise the unstructured data, where clinical narratives and reports were considered as the major form of communication. Recently, artificial intelligence though its powerful tool – natural language processing has been helping the healthcare industry to convert those medical text records to a format which is both useful and easily analyzable.

As a solution, two important techniques of natural language processing have been used to analyze the clinical texts - extraction and summarization. This will open up the pathways for

healthcare industry to make use of analytics driven opportunities.

The clinical

available in the free text form, clearly this lacks uniformity and consistency.

The medical records

reports are usually

include but not limited to doctor’s prescription, discharge summaries, diagnosis reports,

e key information from those text can be potentially taken out using NLP extraction. The optical character recognition (OCR) tool has been deployed to recognize the text from the images which are there in the documents contained in the EHR. This tool is efficient enough to support various file types and hence, will be used in the analysis. Once the key information has been extracted from the large textual data, summarization technique is applied to obtain concise, accurate and meaningful synopsis. In the analysis, the focus was to apply these machine learning techniques efficiently so that better technical solutions can be extended to healthcare industry.

treatment recommendations, etc. Th

# Benchmark

In a study “Automated Methods for the Summarization of Electronic Health Records” which was

published in National Center for Biotechnology Information (NCBI), that the

it was emphasised

revolution of electronic health records has led to an unprecedented increase in the patient’s health information that is stored in an electronic format. As a solution, clinical summarization is adopted for chronically ill patients as they usually have large datasets. They have defined clinical summarization as the act of collecting, describing, and shortening the patient’s information by taking the required points into consideration. This study showcases that for these patients, medical information is difficult to present in a consistent way, which highlights the need for comprehensive EHR summarization. This paper also outlines the techniques and important methodological challenges in the area.

The challenges that were considered in this study are:

* Information redundancy
* Temporality
* Missing data
* Salience detection
* Rules and heuristics
* Deployment of summarization tools

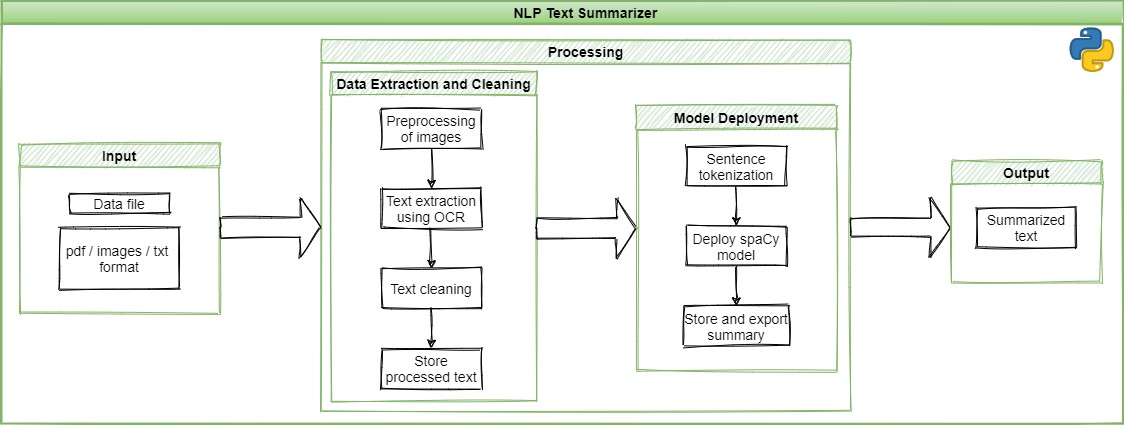
The article defined the summary in four different categories., which are extractive, abstractive,

indicative, and informative. The study that, so far, all the summarization models fall

founds out

under informative and indicated category. But unfortunately, all the summarization models have lacked in evaluation.

# Work-Flow Chart



**Data Collection and Preprocessing**

Data in healthcare industry increases with high velocity and in great volume which makes it big data, which is in various formats such as reports, text records, images, radiation frequencies, audio etc. and is generally unstructured in nature which has to be organized in order to draw information from it. This also creates a challenge for doctors to go through multiple pages for every patient to get to a conclusion.

The data was obtained from the medical professionals, which are in the form of actual patients’ reports from Windsor Regional Hospital. It was in the form of PDF that contained scanned images of the patient’s progress notes, diagnosis, prior and current therapy, imaging investigation results, laboratory reports, so on and so forth.

These images were unclear and contained different font types that made it difficult for the tool to extract the text. So, preprocessing of these images was required so that Optical Character Recognition (OCR) technique could be applied to extract the text.

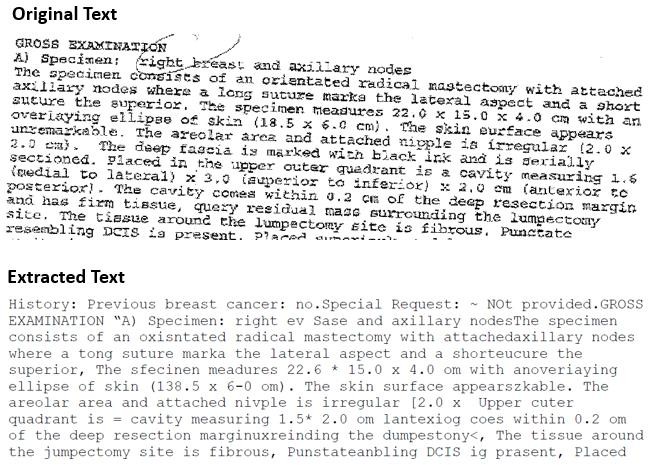
OCR is a technique used to recognize the text which is converted into image. This technology can transform any type of image with text in it into a machine readable format. This technology has been implemented with the help of an OCR engine called Tesseract. This engine looks for models in pixels, letters, words, and sentences. It deploys a two-step method known as adaptive recognition. It needs one data stage for character recognition, followed by a second stage to fill in any missing letters with letters that fit the word or sentence meaning.

To achieve this OCR technique in python, an AWS associated library TEXTRACT has been used. This also works using tesseract OCR. The code used will check if the images are present in the document and if yes, it will use tesseract OCR to convert that images into text. Using encoding the text is converted back into string so that it makes the pre-processing easy. This text extraction process is kept in a python function so that it will be easy to just call and reuse the method.

Pre-processing of the text in an NLP task is very important, because it removes any noise in the data. A library called regex is used to apply the different cleaning techniques. Regex is short form for regular expressions and helps in finding specify patterns and replace them with user specified string. Generally, in text pre-processing, filtering numeric values out is must. But, since our task is summarising the health reports numeric values (Blood Pressure, Heart Rate etc.) are important. The escape sequences were also removed from the extracted text using inbuilt python functions. Further, the special characters and punctuations were also eliminated without affecting the medical terminologies and the sentence formatting.

Next thing in pre-processing is checking the spelling of the words extracted. For this a python library spellchecker is used. This library gives user to use their own word library to verify our text against with. We have found a dictionary with most used medical words since our project is related to medicine. Using this word dictionary in the spellchecker library, we have checked the words for misspelled. Now, we found the misspelled word and its corrected word. To replace the misspelled word with its associated corrected word, we created a dictionary in python which can store misspelled word as key and corrected word as its value. Now a simple for loop will replace the words. This text pre-processing process is kept in a python function so that it will be easy to just call and reuse the method.

Here is an example to illustrate the original image and the text extracted from it:



# Methodologies

NPL summarization which is a technique to prepare a synopsis from huge textual records has been implemented in the analysis. To execute this, two language processing python libraries were used Gensim and spaCy.

Gensim is an open source library in python tool used mainly for topic modelling. In the analysis, genism implementation is done based on TextRank algorithm, which extracts semantic text from documents automatically. This model has the potential to highlight critical information in a large corpus. The summarizer was adjusted to provide apt proportion of the output and it was analysed that 30 percent of the entire medical text record can summarize the medical document most efficiently.

Another model has been built using spaCy package, which is an inbuilt python application that can understand and process large amount of text data. The medical text records were

preprocessed to apply summarizer effectively. For this, the input text was first converted to lower case and tokenized using spaCy’s language model. The loop was created to process each of the tokens and the model was prepared in a way which could potentially ignore the tokens if those were stop-words or punctuations. The keywords were normalised on the basis of their individual weightage. The keywords which were converted to lists were stored in a dictionary using a loop function along with their frequencies.

The results returned the frequency value of 1 for the keywords with highest weightage. Further, the spaCy model has divided the sentences based on full-stop punctuation. The normalized keyword values were added to the key-value pair of the sentence, which denotes the sentence strength and the sentences with more strength were considered in the summary and less important ones were eliminated on the basis of the limits assigned. This limit is basically the number of words to be presented as a summary.

The summaries from both the models, Gensim and spaCy, were compared and because of spaCy’s reliability to obtain better output in the form of summary, it was taken as a final model to be used to prepare Graphical User Interface.

# Graphical User Interface

After successfully completing the NLP model, which can summarize any document to its precise summary, the next phase was to come up a web interface. Since, it is difficult to use the models developed in python tool directly by the healthcare professionals to obtain the summarized reports, it was imperative to build a web interface, where users can directly upload pdf or text and get their summary in just one click.

To execute this, Flask, which is a lightweight web application framework, has been used. It is a third party Python library exclusively used for developing web applications. The Visual Studio code has been used for developing a web app, because of its efficiency to handle multiple programming languages.

To initiate the development of an interface, python environment was created and files in VS-code were set up for this project, after which all the required libraries were installed. To store these libraries, a txt file was created with name “requirements”. Also, a template folder consisting of the html files were created. Similarly, a static folder consisting of the CSS (Cascading Styling Sheets) were created. The extraction, preprocessing, and spacy methods were stored in their respective python files. Now, a python file has been created with name “app.py”, which contained all the codes. This is where when a user makes a request is routed to. Also, extraction, preprocessing, and spacy functions were imported as libraries over here.

In this python file, a routing blocks for text input and pdf's input from the user were created and in pdf input block, the extraction, preprocessing, and spacy functions were called in order. So, when a user uploads a pdf file, at backend it is routed to this block and processed under all three methods and the final summary can be stored in a variable called summary.

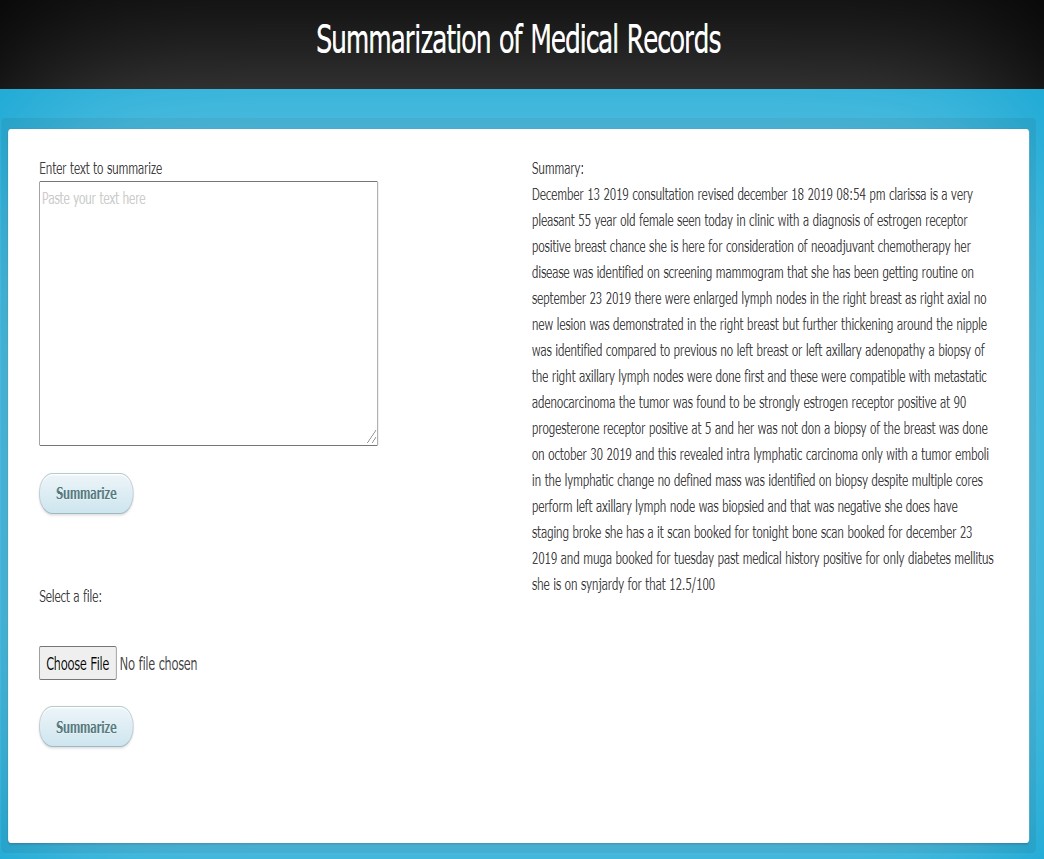
In text input block, extraction was excluded because user will provide the text directly, which will directly go through the preprocessing and spaCy methods and the summary will be stored in a variable called summary.

The backend interface has been developed. Further, a front end interface was needed where users can give and get their requests. For this, an HTML file called “home” was created. To start with, a code with head block was written which had the heading of the web app ("Summarization of Medical Reports").

Next, a body block was created for text input and action command was used to integrate with app.py. This action command takes the routing command of text input from app.py. In the same way another body block was created for pdf input and action command was used which routes the pdf input command from app.py.

In the app.py file render\_template was used to integrate the frontend and backend. Finally, Cascading Styling Sheets (CSS) were used for styling the frontend. This CSS was again integrated with html in the body block using class name. Next, the app.py was executed which has provided a local link that can be opened in a web platform.

Image representing a Web Interface:



# Results

The project was undertaken with an aim to create a graphical user interface based on the model developed through natural language processing to ease the healthcare professionals by automatic extraction and summarization of data from clinical reports. There are several benefits that can be served by the web interface. Firstly, patients’ medical history in a summarised version can be easily obtained with just a single click, which would help in delivering the quality results in healthcare. There is no need for the doctors to go through the report containing multiple pages for every patient to get to a conclusion. The application would potentially save their time in efficient decision-making.

Further, there is a the medical inaccuracies by effective diagnosis, which

potential of reducing

will also allow the doctor to decide the right course of treatment. Overall an effective usage of the interface can enhance the patient and provider interaction through completeness of key information.

# Challenges

There were several challenges while working on this project, which were successfully overcome. Among those, few major challenges are listed below:

* Familiarizing with the terminology used in healthcare industry to efficiently perform extraction and summarization.
* The performance of the final NLP model on the real time data.
* Providing stakeholders, a web interface which will help in utilizing the model and thereby, achieving the desirable output.

# Future Scope

As a future work, the following points can be taken into consideration, which will further enhance the project:

* Further modifying the python codes so that it can work on all the file formats available.
* Include another tab for URL in the existing web interface, so that the text summarization can be done using a web resource.
* Try and test different techniques and use higher processing power to speed up the working of the model.
* Explore various summarization techniques using more and/or different set of data.
* Deploy the model to cloud to enable it to work at different places at a high speed.

# References

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4607324/>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6372467/>

<https://info.nhanow.com/blog/whats-ehr-and-why-is-it-important>

[https://www.healthit.gov/faq/what-are-advantages-electronic-health-](https://www.healthit.gov/faq/what-are-advantages-electronic-health-records#%3A~%3Atext%3DThe%20benefits%20of%20electronic%20health%20records%20include%3A%2Ctimeliness%2C%20efficiency%2C%20and%20equity) [records#:~:text=The%20benefits%20of%20electronic%20health%20records%20include%3A,timeliness%](https://www.healthit.gov/faq/what-are-advantages-electronic-health-records#%3A~%3Atext%3DThe%20benefits%20of%20electronic%20health%20records%20include%3A%2Ctimeliness%2C%20efficiency%2C%20and%20equity) [2C%20efficiency%2C%20and%20equity](https://www.healthit.gov/faq/what-are-advantages-electronic-health-records#%3A~%3Atext%3DThe%20benefits%20of%20electronic%20health%20records%20include%3A%2Ctimeliness%2C%20efficiency%2C%20and%20equity).

[https://hitinfrastructure.com/news/unstructured-healthcare-data-needs-advanced-machine-learning-](https://hitinfrastructure.com/news/unstructured-healthcare-data-needs-advanced-machine-learning-tools#%3A~%3Atext%3DAbout%2080%20percent%20of%20healthcare%2Cof%20Things%20(IoT)%20devices) [tools#:~:text=About%2080%20percent%20of%20healthcare,of%20Things%20(IoT)%20devices](https://hitinfrastructure.com/news/unstructured-healthcare-data-needs-advanced-machine-learning-tools#%3A~%3Atext%3DAbout%2080%20percent%20of%20healthcare%2Cof%20Things%20(IoT)%20devices).

[https://towardsdatascience.com/natural-language-processing-nlp-for-electronic-health-record-ehr-part-](https://towardsdatascience.com/natural-language-processing-nlp-for-electronic-health-record-ehr-part-ii-1b2a024ea03a) [ii-1b2a024ea03a](https://towardsdatascience.com/natural-language-processing-nlp-for-electronic-health-record-ehr-part-ii-1b2a024ea03a)

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7157505/>

[https://towardsdatascience.com/how-to-extract-text-from-images-with-python-](https://towardsdatascience.com/how-to-extract-text-from-images-with-python-db9b87fe432b#%3A~%3Atext%3DThe%20Python%20Library%26text%3DPython%2Dtesseract%20is%20an%20optical%2Cfor%20Google%27s%20Tesseract%2DOCR%20Engine) [db9b87fe432b#:~:text=The%20Python%20Library&text=Python%2Dtesseract%20is%20an%20optical,for](https://towardsdatascience.com/how-to-extract-text-from-images-with-python-db9b87fe432b#%3A~%3Atext%3DThe%20Python%20Library%26text%3DPython%2Dtesseract%20is%20an%20optical%2Cfor%20Google%27s%20Tesseract%2DOCR%20Engine)

[%20Google's%20Tesseract%2DOCR%20Engine.](https://towardsdatascience.com/how-to-extract-text-from-images-with-python-db9b87fe432b#%3A~%3Atext%3DThe%20Python%20Library%26text%3DPython%2Dtesseract%20is%20an%20optical%2Cfor%20Google%27s%20Tesseract%2DOCR%20Engine)

<https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>

[https://medium.com/@oyewusiwuraola/simple-natural-language-processing-projects-for-health-](https://medium.com/%40oyewusiwuraola/simple-natural-language-processing-projects-for-health-sciences-how-to-extract-and-summarize-53ea2e7b3f1d) [sciences-how-to-extract-and-summarize-53ea2e7b3f1d](https://medium.com/%40oyewusiwuraola/simple-natural-language-processing-projects-for-health-sciences-how-to-extract-and-summarize-53ea2e7b3f1d)

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4986665/>