TEXT MINING PROJECT

Text Classification & Summarization on medical transcriptions

Barbera Valentina, 856780
Sinanaj Luca, 844540

01/2024

Department of Computer Science, Faculty of Data Science, Università degli Studi di Milano-Bicocca

ABSTRACT

The goal of this project is to apply Natural Language Processing (NLP) techniques to a dataset of Medical Transcriptions, which contains transcripts of medical visits across various medical specialties.

The tasks applied to this dataset include **text preprocessing, text representation, text classification, and text summarization.** The models used for text classification are Logistic Regression, Naive Bayes, and Random Forest using TF-IDF, as well as the pre-trained BERT model. For the text summarization part, pre-trained models such as t5 and GPT, along with models provided by the SUMY library (Luhn, Lexrank, and Textrank), have been employed.

This report will present the performance of the aforementioned models along with their respective evaluation metrics.

1.DATASET

The dataset used was the Medical Transcriptions dataset obtained from Kaggle (https://www.kaggle.com/datasets/tboyle10/medicaltranscriptions) to perform NLP tasks.

The dataset consists of 4000 records containing transcriptions of medical visits categorized by medical specialty. It comprises 6 columns:

1. **Description**: A brief description of the respective transcription.
2. **Medical specialty.**
3. **Sample name**: Title of the transcription.
4. **Transcription**.
5. **Keywords**: Relevant keywords present in the transcription.

The data is provided in a CSV file where each individual record represents a medical visit transcription.

2.EXPLORATORY ANALYSIS

The initial phase of the project involved cleaning the dataset, where records containing *NaN/Null* values were removed. Specifically, there were 33 missing values in the "Transcriptions" column and 1068 null values in the "Keywords" column, as reported in Figure 1.

Figure 1

```
data.isna().sum()

Unnamed: 0              0
description             0
medical_specialty       0
sample_name             0
transcription          33
keywords             1068
```

Later, we aimed to understand the distribution of data within each medical category ("Medical Specialty").

Figure 2

```
data['medical_specialty'].value_counts().plot(kind='bar')
```

```
<Axes: >
```

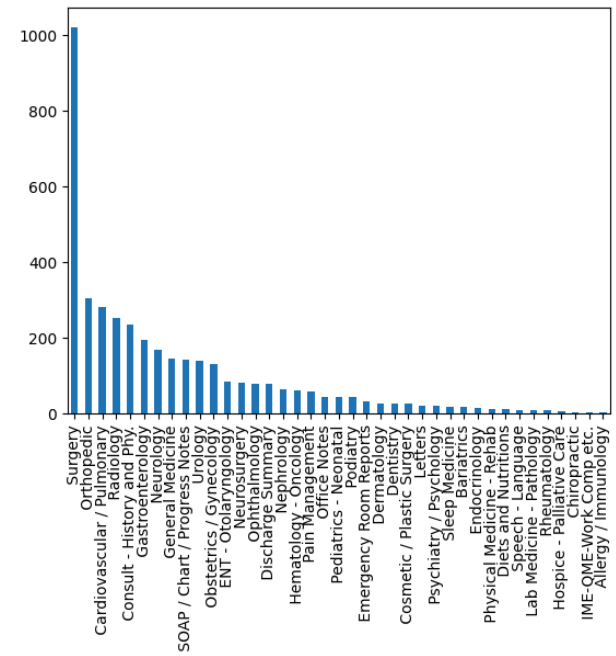


Figure 3

```
Surgery                          1021
Orthopedic                        303
Cardiovascular / Pulmonary        280
Radiology                         251
Consult - History and Phy.        234
Gastroenterology                  195
Neurology                         168
General Medicine                  146
SOAP / Chart / Progress Notes     142
Urology                           140
Obstetrics / Gynecology           130
ENT - Otolaryngology               84
Neurosurgery                       81
Ophthalmology                      79
Discharge Summary                  77
Nephrology                         63
Hematology - Oncology              62
Pain Management                    58
Office Notes                       44
Pediatrics - Neonatal              42
Podiatry                           42
Emergency Room Reports             31
Dermatology                        25
Dentistry                          25
Cosmetic / Plastic Surgery         25
Name: medical_specialty, dtype: int64
```

From *Figure 2*, we observe that out of nearly 4000 observations and 39 Medical Specialties, over 1000 observations belong to the "Surgery" category, while the remaining observations are distributed among the top 10 categories.

Based on the results presented in *Figure 3,* which display the top 25 Medical Specialties by the number of observations, we decided to select 5 medical categories for text classification, specifically 'Orthopedic', 'Cardiovascular/Pulmonary', 'Consult - History and Phy.', 'Gastroenterology', and 'Neurology'.

Of the 5 chosen categories, "Surgery" was not considered due to its perceived generality, and including it would have resulted in an excessively imbalanced dataset. After this initial data exploration phase, we significantly reduced the dataset from almost 4000 observations to just over 1000. Consequently, to avoid further reduction in the dataset size, we opted to retain the 5 medical categories despite their imbalanced representation, as the difference in the number of observations among them was minimal.

Following the selection of these categories for text classification, we decided to reshape the dataset, considering only the two columns relevant for this initial task: the "Transcription" column and the "Medical Specialty" column.

3.TEXT PREPROCESSING

Subsequently, we performed text preprocessing by removing punctuation, numbers, repetitions, and converting the entire text to lowercase, as shown in *Figure 4.*

Figure 4

	medical_specialty	transcription
3	Cardiovascular / Pulmonary	2-D M-MODE: , ,1. Left atrial enlargement wit...
4	Cardiovascular / Pulmonary	1. The left ventricular cavity size and wall ...
7	Cardiovascular / Pulmonary	2-D ECHOCARDIOGRAM,Multiple views of the heart...
9	Cardiovascular / Pulmonary	DESCRIPTION:,1. Normal cardiac chambers size....
11	Cardiovascular / Pulmonary	2-D STUDY,1. Mild aortic stenosis, widely calc...
	medical_specialty	transcription
3	Cardiovascular / Pulmonary	d mmode left atrial enlargement with left a...
4	Cardiovascular / Pulmonary	the left ventricular cavity size and wall th...
7	Cardiovascular / Pulmonary	d echocardiogrammultiple views of the heart an...
9	Cardiovascular / Pulmonary	description normal cardiac chambers size nor...
11	Cardiovascular / Pulmonary	d study mild aortic stenosis widely calcified ...

We applied tokenization to the processed text from the previous step.

This process involved breaking down the text into smaller units called "tokens," which are essential for natural language preprocessing.

Following tokenization, we removed the "stop words" — common words that frequently occur within the text — using the NLTK library.

Next, we implemented "stemming," a normalization process that aims to reduce a word to its root or base form. This helps treat words with similar or related meanings in a consistent manner.

After this initial text preprocessing phase, we decided to present some graphical representations through the analysis of unigrams and trigrams.

Figure 5

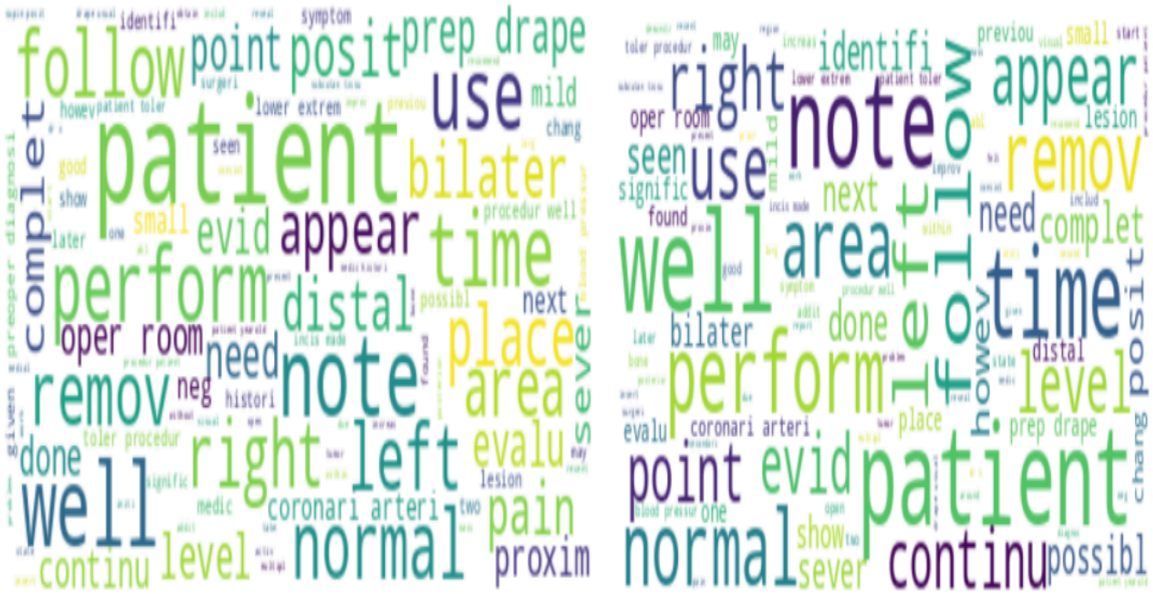
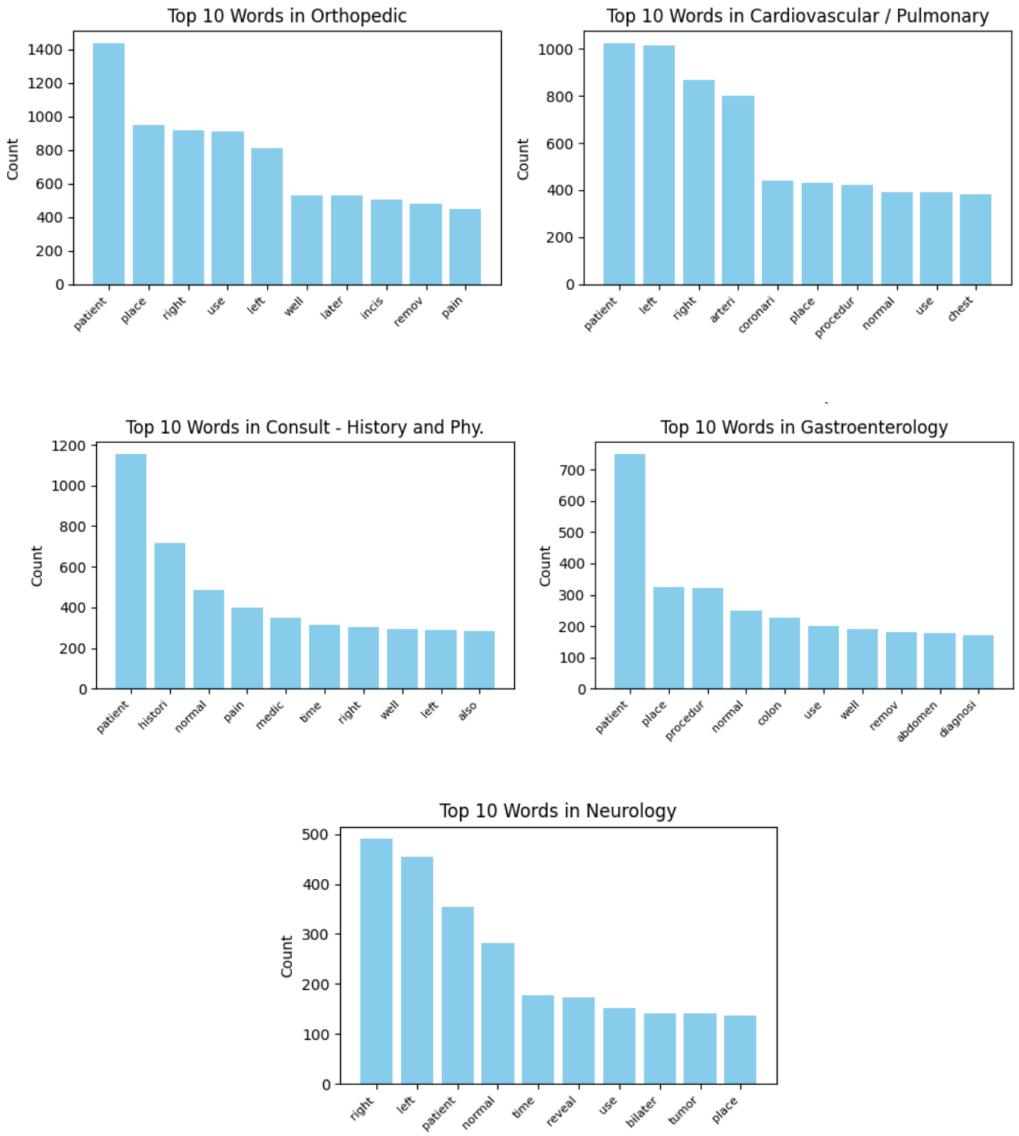


Figure 5 displays a word cloud composed of unigrams, while the second one exhibits a word cloud composed of trigrams. Upon careful observation, recurring words like "patient," "well," "right," "left," or "note" are noticeable.

Based on this initial outcome, we chose to identify the 10 most frequent words within each of the 5 considered categories and remove them, similar to the process used for eliminating stop words earlier.

Figure 6



Based on the results shown in *Figure 7*, we decided to eliminate the following words:

Figure 7

```
{'well', 'arteri', 'also', 'pain', 'perform', 'left', 'incis', 'normal', 'note', 'use', 'procedur', 'time', 'place', 'histori', 'remov', 'posit', 'patient', 'right'}
```

However, among these words, we opted to retain "history" as it is representative of the "Consult-History" category. This word is capable of aiding the classification model in

distinguishing this category from the others, unlike the remaining words that appear more generic.

4.TEXT REPRESENTATION

For text representation, the choice fell on the TF-IDF (Term Frequency - Inverse Document Frequency) technique, commonly used in information retrieval and text analysis to evaluate the importance of a word within a document. This measure consists of two main components:

TF (Term-Frequency): Measures the frequency of a word in a specific document (important words appear frequently in a document).

IDF (Inverse Document Frequency): Measures how rare or common a word is across all documents in the corpus. Consequently, words common in many documents have a lower IDF, while rare words have a higher IDF.

The use of this technique for subsequent Text Classification is crucial because it allows the selection of the most important words in each record (medical transcription). This helps in identifying relevant terms that aid in accurately classifying the transcription with the correct Medical Specialty.

5.TEXT CLASSIFICATION

Text classification is a Natural Language Processing (NLP) technique that involves automatically categorizing text documents, in our case, each content within the "Transcription" column, into different predefined classes or categories, namely within their respective medical specialties ("Medical Transcription").

For this project phase, we initially defined specific labels for each of the five Medical Specialty categories, as follows:
Figure 8

medical_specialty	label	data_type	
Cardiovascular / Pulmonary	0	train	224
		val	56
Consult - History and Phy.	4	train	187
		val	47
Gastroenterology	3	train	156
		val	39
Neurology	2	train	135
		val	33
Orthopedic	1	train	242
		val	61

The dataset was divided into a training set and a test set. Specifically, 944 observations were used for training the model, and 236 for testing purposes to assess the model's performance.

5.1 LOGISTIC REGRESSION

The first classification model we applied is the logistic regression model combined with the previously explained TF-IDF technique. TF-IDF is essential for converting each text document into a numerical vector used during the training phase.

After training the model on the training data, we proceeded to evaluate its performance

on the test set, as depicted in *Figure 9*.

Figure 9

	precision	recall	f1-score	support
0	0.85	0.91	0.88	57
1	0.59	0.66	0.62	50
2	0.97	0.73	0.83	48
3	0.48	0.44	0.46	32
4	0.74	0.82	0.78	49
accuracy			0.74	236
macro avg	0.73	0.71	0.71	236
weighted avg	0.75	0.74	0.74	236

The classification report displayed in Figure 8 provides a detailed evaluation of the logistic regression model's performance. Each row in the report corresponds to a specific class, while the columns present different evaluation metrics for each class.

Among the metrics reported are:

Precision: Measures the model's accuracy when predicting a specific class. It's calculated as the ratio of true positives (correctly predicted positive observations) to the sum of true positives and false positives (observations wrongly predicted as positive). For instance, for class 0 (Cardiovascular/Pulmonary), 85% of positive predictions are correct. The class with the highest precision is class 2 (Neurology), whereas classes 1 and 3 (Orthopedic and Gastroenterology, respectively) show relatively lower precision values.

Recall: Evaluates how well the classification model identifies all positive examples in the dataset. It's calculated as the ratio of true positives (correctly identified positive examples) to the sum of true positives and false negatives (positive examples wrongly classified as negative). The class with the highest recall is class 0, where 91% of examples are correctly predicted, followed by 82% for class 4 (Consult-History and Phy).

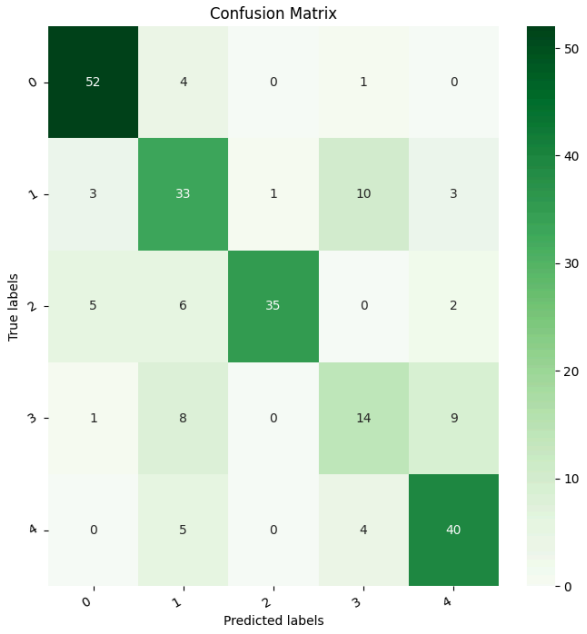
F1-score: Represents the harmonic mean of Precision and Recall. In this case, the best performance is observed for class 0, with an F1-score of 0.88.

Accuracy: Measures the percentage of correct predictions made by the model compared to the total number of observations. For the Logistic Regression model, the accuracy is 0.74, indicating that 74% of total predictions were correctly classified.

Macro avg and Macro weighted are respectively the average of metrics (precision, recall, and F1-score) for each class and the weighted average of metrics, where each class contributes equally based on its presence in the test set.

In addition to the classification report, the confusion matrix displayed in *Figure 10* provides further details about the model's performance by showing the distribution of correct and incorrect predictions for each class.

Figure 10

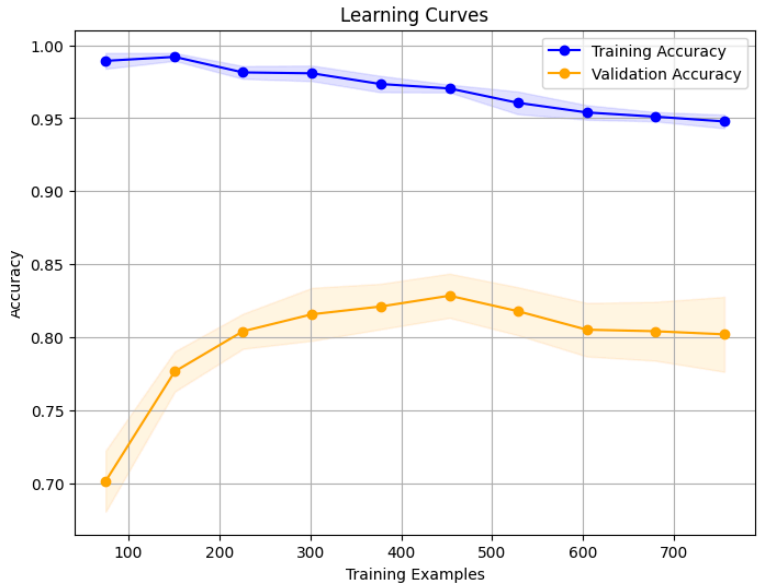


From the above figure, it's evident that the best-predicted class is class 0, where out of a total of 57 examples, 51 are correctly classified. Conversely, the class with the poorest classification performance is class 3, where out of 32 observations, only 14 are correctly classified.

Finally, *Figure 11* displays the learning curves of the model. A decreasing trend is

observed for the training accuracy, which tends to plateau around 70 training examples. As for the validation curve, there's an ascending trend up to just over 400 training examples, followed by stabilization around 700 training examples. The validation curve showcases a wider range of values compared to the training curve, which also leads to a higher standard deviation.

Figure 11



5.2 NAIVE BAYES

The second classification model we chose to use is the Naive Bayes probabilistic model, again implemented using the TF-IDF method. This model describes how to update the probabilities of a hypothesis given the observed evidence. In terms of classification, it computes the conditional probability that an instance belongs to a specific class given the observation of certain features. It is based on the assumption that variables are independent of each other, simplifying the calculation of conditional probabilities.

Figure 12 displays the classification report, similar to the explanation provided earlier for

Logistic Regression.

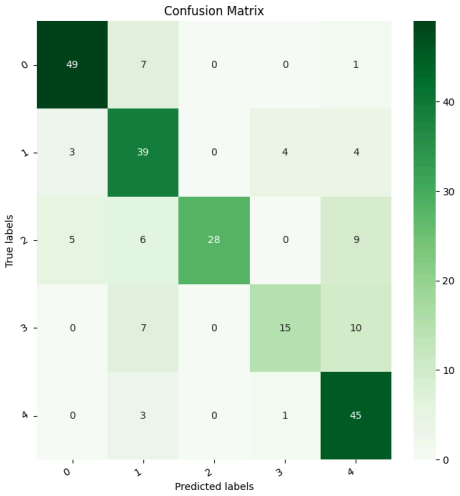
Figure 12

	precision	recall	f1-score	support
0	0.86	0.86	0.86	57
1	0.63	0.78	0.70	50
2	1.00	0.58	0.74	48
3	0.75	0.47	0.58	32
4	0.65	0.92	0.76	49
accuracy			0.75	236
macro avg	0.78	0.72	0.73	236
weighted avg	0.78	0.75	0.74	236

From the reported classification, it's observed that class 2 shows a precision of 1, meaning 100% of the positive predictions are correct. This result is unusual and might indicate a potential overfitting issue with the model. Regarding Recall, the best results are reported for class 0, where 86% of the examples are correctly predicted, and for class 4, where 92% of the examples are correctly predicted. Classes 0 and 4 also demonstrate the highest F1-score values.

Compared to the previous model, there's a slight increase in accuracy, reaching 0.75, indicating that 75% of the total predictions were classified correctly.

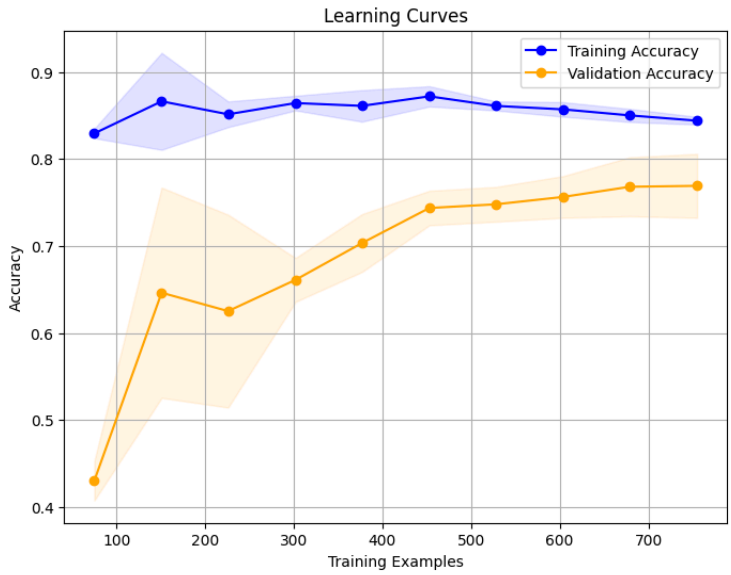
Figure 13



In this case as well, from *Figure 13* displaying the confusion matrix, we notice poor performance for category number 3.

Figure 14 represents the learning curves, showing an increasing trend in validation accuracy with a wider range of values compared to the previous model. This suggests a potentially higher standard deviation (std deviation). As for the training accuracy, the curve displays a consistent trend over time.

Figure 14



5.3 RANDOM FOREST

The third classification model used is the Random Forest, also implemented through the TF-IDF method. Random Forest is a machine learning model based on ensemble learning, combining predictions from various base models to improve overall performance and model generalization. It relies on using random decision trees as the base models. Decision trees are machine learning tools that make sequential decisions based on if-then-else rules, executing a condition based on another condition.

Figure 15

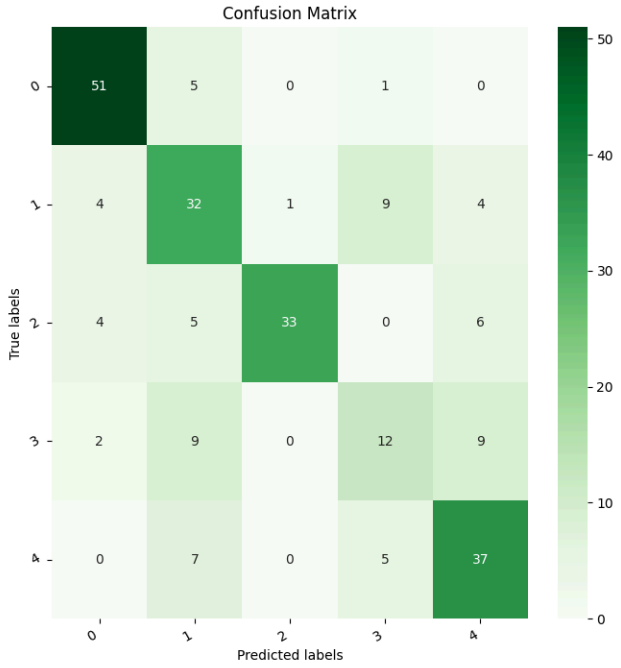
```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.89      0.86        57
           1       0.55      0.64      0.59        50
           2       0.97      0.69      0.80        48
           3       0.44      0.38      0.41        32
           4       0.66      0.76      0.70        49

    accuracy                           0.70       236
   macro avg       0.69      0.67      0.67       236
weighted avg       0.71      0.70      0.70       236
```

Figure 15 presents the classification report, indicating that the model correctly predicted 70% of the total observations. The class with the highest precision is number 2, where 97% of positive predictions are correct. Class 0 exhibits the highest recall value of 0.89. However, once again, class number 3 performs poorly, a result also evident in Figure 15, where it's noticeable that class 0 shows the best performance.

Figure 16



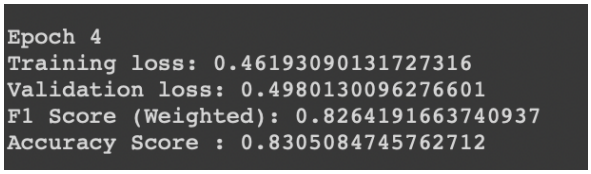
5.4 BERT

In this configuration, we used the pre-trained BERT model with the "bert-base-uncased" version (https://huggingface.co/bert-base-uncased). This model has been fine-tuned for a binary classification task, with five labels to predict. The model is implemented using the Hugging Face's ClassificationModel class, which allows an easy fine-tuning of the pre-trained BERT model on a given dataset.

The maximum sequence length for input data is set to 128, and the batch size for

training is 32. The model will be trained for 5 epochs. The best model chosen was the following (*figure 17*):

Figure 17



This BERT model, after Epoch 4, shows promising performance with an overall accuracy of 83.05%. The F1-score (Weighted) of 0.8264 also indicates good overall model performance. The classification report (*figure 18*) further details the precision, recall, and F1-score for each class, demonstrating varying degrees of accuracy and performance across different medical specialties. The validation loss (0.4980) is slightly higher than the training loss (0.4619), indicating a minor level of overfitting. However, both loss values are relatively low, indicating that the model is learning well during training and performing reasonably well on unseen validation data. The model seems to have acceptable generalization capabilities, but a more balanced performance on unseen data could potentially be achieved by further optimization or regularization techniques to reduce overfitting.

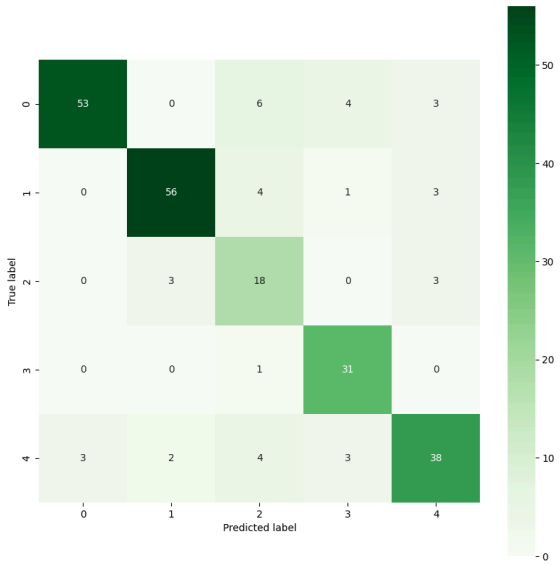
Figure 18

	precision	recall	f1-score	support
0	0.9464	0.8030	0.8689	66
1	0.9180	0.8750	0.8960	64
2	0.5455	0.7500	0.6316	24
3	0.7949	0.9688	0.8732	32
4	0.8085	0.7600	0.7835	50
accuracy			0.8305	236
macro avg	0.8027	0.8314	0.8106	236
weighted avg	0.8482	0.8305	0.8346	236

The accuracy breakdown for individual classes (*figure 19*) provides insights into how well the model performs for each specific category. Here's a closer look at the accuracy for each class:

- Cardiovascular / Pulmonary: 53 correct predictions out of 56 samples (94.6% accuracy)
- Orthopedic: 56 correct predictions out of 61 samples (91.8% accuracy)
- Neurology: 18 correct predictions out of 33 samples (54.5% accuracy)
- Gastroenterology: 31 correct predictions out of 39 samples (79.5% accuracy)
- Consult - History and Phy.: 38 correct predictions out of 47 samples (80.9% accuracy)

Figure 19



6.TEXT SUMMARIZATION

Text summarization is an NLP technique that aims to extract the most relevant or essential information from a text document, creating a shorter and more concise version. Its objective is to allow users to obtain important information without reading the entire document.

There are two main approaches to text summarization:

1. **Extraction-based Summarization:**

This approach extracts sentences or portions of text directly from the original document to create the summary. The extracted sentences are selected based on criteria such as relevance, keyword frequency, and coherence with the main theme.

2. **Abstraction-based Summarization:**

This approach generates summaries by rephrasing or abstracting information from the source text. It involves understanding the context and generating new sentences that convey the key points while being coherent.

For text summarization, the entire dataset was considered instead of specific categories used in classification. The aim was to obtain summaries for each entry in the "Transcription" column and apply evaluation metrics like Rouge to assess the quality of the generated summaries. Additionally, comparisons were made between the content of the "Description" column (containing brief descriptions of the "Transcription" column) and the obtained summaries to assess the difference between automatically generated summaries using pre-trained ML models and those created by humans.

In this final phase of the project, text preprocessing techniques used previously, such as lowercasing, punctuation removal, and duplicate removal, were applied. Unlike the classification part, numbers were not removed in this case, as they are crucial for better understanding medical diagnoses, indicating, for example, the patient's age.

6.1 ABSTRACTION-BASED SUMMARIZATION

6.1.1 T5 Small

The first model used for text summarization is T5 (Text to Text Transfer Transformer). The small version was employed due to computational limitations. This model's distinctive feature is its "text-to-text" approach, unifying the formulation of various tasks as text conversion problems. It enables the model to address different tasks with a single base architecture, simplifying the training process and knowledge transfer

between tasks.

6.1.2 GPT

The second text summarization model applied is GPT (Generative Pre Trained Transformer), based on transformer architecture capable of capturing long-range relationships in text.

6.1.3 TXTAI

Executes machine-learning workflows to transform data and build AI-powered semantic search applications. This pipeline runs a text2text model that abstractively creates a summary of the input text.

6.2 EXTRACTION-BASED SUMMARIZATION

6.2.3 LexRank

Unlike TextRank, LexRank uses cosine similarity between vectors of sentences based on keywords extracted from the text. More similar sentences receive higher scores and are selected for the summary.

6.2.4 Luhn

It's based on identifying and extracting relevant key sentences from the text. Sentences containing the most important keywords are then used to create the summary.

6.2.5 TextRank

It is a summarization extraction algorithm based on the concept of graph and node weighting. The main idea is to treat sentences as nodes in a graph, with the weights of the connections between nodes representing the semantic relationship between them. The algorithm assigns a score to each sentence based on its relevance to other

sentences in the text.

6.3 EVALUATION METRICS

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

It evaluates the quality of a generated summary concerning one or more reference summaries (gold standard). It measures the overlap of n-grams (consecutive word sequences of length n) between the generated summary and the reference summaries.

ROUGE-1 (unigram overlap)

Focuses on the overlap of single words between the generated summary and the reference summaries. A high score indicates a good match of individual words, suggesting accurate capturing of key information.

ROUGE-2 (bigram overlap)

Assesses the overlap of bigrams (consecutive pairs of words) between the generated summary and the reference summaries. A higher score indicates better matching between the bigrams in the generated summary and those in the references.

ROUGE-L (Longest Common Subsequence)

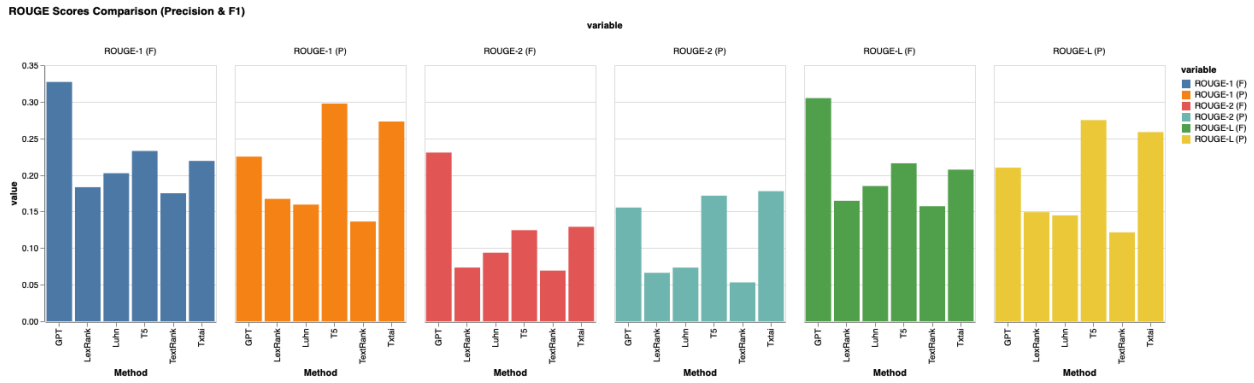
Measures the length of the longest sequence of common words between the generated summary and the references, considering the order of words in the longest common sequence.

However, it's important to note that Rouge-1 and Rouge-2 do not consider aspects such as semantics, coherence, and text fluency, while Rouge-L might be influenced by the length of the longest common sequence and does not consider synonyms or similar words that could be equally informative.

RESULTS

Figure 20

Method	ROUGE-1 (P)	ROUGE-1 (R)	ROUGE-1 (F)	ROUGE-2 (P)	ROUGE-2 (R)	ROUGE-2 (F)	ROUGE-L (P)	ROUGE-L (R)	ROUGE-L (F)
LexRank	0.1673	0.2734	0.1832	0.0662	0.1198	0.0734	0.1491	0.2491	0.1646
TextRank	0.1363	0.3186	0.1750	0.0531	0.1391	0.0692	0.1215	0.2914	0.1572
Luhn	0.1595	0.3551	0.2023	0.0733	0.1791	0.0937	0.1447	0.3291	0.1848
GPT	0.2250	0.7471	0.3271	0.1554	0.5914	0.2306	0.2099	0.6954	0.3049
Txtai	0.2730	0.2087	0.2191	0.1778	0.1176	0.1291	0.2585	0.1973	0.2073
T5	0.2975	0.2207	0.2327	0.1716	0.1164	0.1244	0.2749	0.2056	0.2159



Among the reported metrics, we can observe that the F1 value greater than Rouge-1 is reported by the GPT model. This indicates a good match compared to other models for individual words. GPT is also the model that shows the best performance in Rouge-2 and Rouge-L. We can notice that the F1 score values in all three scenarios are not particularly high. This is because the Rouge metric does not take into account synonyms and semantics, and the Description column, being manually created, contains the presence of synonyms, which in this case are penalized, leading to an imperfect match between terms.

CONCLUSIONS AND FUTURE DEVELOPMENTS

Overall, the achieved results are quite satisfactory for both tasks. Regarding the classification part, it was observed that the "Neurology" category performs less effectively compared to others, likely due to a smaller number of observations. For

future work, one could consider applying a larger dataset and a more powerful GPU to have the opportunity to implement more complex and efficient models.

The Text Summarization part yielded less promising results compared to the classification part, and a potential improvement could involve the application of more advanced and computationally expensive models. Additionally, incorporating evaluation metrics that consider semantics and the presence of synonyms could further enhance the performance.