NLP to summarize the protocol document and ML to predict the trial's completion

Rutuja Rajendra Bhuwad
School of Computing
Dublin City University
Dublin, Ireland
rutuja.bhuwad2@mail.dcu.ie
21264255

Shreya Ashok Bangera
School of Computing
Dublin City University
Dublin, Ireland
shreya.bangera2@mail.dcu.ie
20210943

***Abstract*—Clinical trials are one of the most discussed topics in the global pandemic crisis as they are the first step towards finding new approaches to diagnosing and treating diseases. However, a single clinical study often takes six to seven years to complete and costs millions. Therefore, it is essential to simplify this lengthy procedure and make it more efficient. This paper proposes a model to predict a clinical trial's likelihood of being 'Completed' or 'Not Completed' based on its study protocol document. A study protocol document is a complete description of the research before the trial begins. The dataset considered is available on clinicaltrials.gov, a clinical trial registry managed by the United States National Library. First, the critical information from the protocol document is extracted and later summarized using a longformer. After that, Doc2Vec is used to transform the summary and remaining metadata into a feature vector. Machine learning predicts a clinical trial's completion status using the feature vector in the second part. In this paper, four different machine learning classification models Logistic Regression, Decision Tree Classifier, Support Vector Classifier and Random Forest Classifier are tried for comparison. The Random Forest Classifier showed the highest accuracy among these four models, at 64.38 percent.**

***Index Terms*—Natural language Processing, Transformers, Clinical Trials, Diabetes, longformer, BERT-base-uncased pre-trained model, Doc2Vec, Gensim.**

I. Introduction

Medical research has evolved from being conducted through trial and error to the current practices of legal restrictions. Clinical research has had a prolonged and intriguing history. Clinical trials have a documented history that dates back to descriptions in 500 BC. The path leads from dietary therapy legumes and lemons to medication [1]. Efforts were undertaken to improve the design and statistical components of clinical trials after the fundamental methodology was outlined in the 18th century. Scientific challenges were overcome using experimental controls, randomness, treatment comparisons, placebo tests, and double-blind experiments. New methods for assessing study designs have been used over the years, highlighting the significant changes in medical practice [2]. Additionally, the need for standardized techniques to evaluate the efficacy of treatments and rules for patient protection followed the creation of evidence-based practice. However, constant improvements to research methods are required to achieve the best clinical results.

In short, clinical trials are research projects that study novel techniques, put them to the test, and evaluate how they affect people's health. Innovative diagnostic, therapeutic, and preventative approaches to treating diseases are developed with the help of clinical trials, as they allow researchers to discover what works and doesn't in humans, which cannot be investigated in a single step in a lab setting.

Clinical studies are carried out in stages. Each phase is meant to address specific concerns while taking the necessary precautions to protect participants. Once the studies are approved by regulatory agencies, trials are conducted in four different phases. Phase 1 checks if the treatment is safe and identifies the side effects. Phase 2 checks the effectiveness of the test and further evaluates its safety. Phase 3 confirms the effectiveness, monitors the side effects and compares the treatment with other treatments. Lastly, Phase 4 provides additional information, including risk, benefits and best use after approval. Before a trial starts, thorough and in-depth secondary research is carried out and based on the findings, research planning begins for further execution. This planning is documented for reference and designated as a protocol document, which is a document that outlines the aim, study design, technique, statistical considerations, and organizational structure of a clinical study. Additionally, it guarantees the validity of the information acquired and the safety of trial participants.

Recently, COVID-19 has surfaced and exploded as a primary clinical trial subject. The vaccine's development proved challenging for a variety of reasons. The COVID-19 vaccine's acceptability is the main problem for developed countries, whereas the availability and cost of the vaccine are the primary issues for underdeveloped countries [3]. All vaccines take a long time to develop due to the considerable steps involved in the production. The successful completion of the most significant milestone in vaccine development. Numerous investigations and research activities are being carried out in this field to accelerate and optimize the procedure of vaccine development and to evaluate the data to produce more accurate results. Therefore, developing a project that could anticipate clinical research outcomes would be beneficial. The pharmaceutical industry spends more than 180 billion dollars annually on research and development activities yet regularly experiences failure rates of approximately 50 percent or more [4].

Clinical trials are inherently vexing due to their complexity, regulation and therefore serve as a roadblock to more rapid and effective findings. The article explains the factors leading to the clinical trial failure which include, lack of efficacy, safety concerns, a lack of funds to finish a study, failing to uphold proper manufacturing processes, disobeying FDA directives, having difficulty with patient recruiting, enrollment, and retention [5].

Science and technology have made significant changes in many sectors. Using them in clinical trials will be one of the leading developments for humankind. It hasn't been an oblivious fact that ML has turned out to be a boon to multiple industries and conglomerates. Similarly, It could significantly impact the pharmaceutical and Biomedical sectors. ML is leaps ahead in terms of accuracy over long-practised discriminant analysis. Historical data can be analysed using advanced technology to neutralise any potential for mistakes in clinical trials going forward. Natural language processing can combine humans and machines to enhance the machine learning procedure for better results on analysis. Different aspects of machines can be used in the varied areas of clinical trials. Involving science and technology will also develop a feeling of trust in participants to get involved in clinical trials and provide a more varied population for trials. A high volume of data is generated every second, which can be processed and used for prediction and analysis using appropriate machine learning models and techniques.

We intend to optimize the clinical trial study by accelerating the process by creating a machine learning model to resolve the problem statement: Whether the trial will be completed or it will be incomplete? The ML pipeline could help to predict the completion of the trial and minimise the cost and allocate the funds in the right direction. It would give a chance to small-scale companies to cope up with market leaders. Significant reductions in failures and accurate predictions could help these companies to avert losses and develop a new drug successfully. This in turn, would bring competition to the existing market; it would bring considerable benefits to consumers by giving them more and cheaper options to choose from.

So, to optimize clinical trial studies, the proposed method will be implemented in two stages. In stage one, we will use the study protocol document which is created before the trials start, and summarize the document using a longformer. Furthermore, we employed a Doc2Vec pre-trained gensim to tokenize the summarised text data and then converted text into a fixed-size vector. In stage two, passed the summarised text vector along with the metadata to the machine learning model for training and prediction purposes. Finally, we compared the accuracy for three machine learning models to evaluate the results.

II. Literature Review

In this section, we studied the previously proposed methods for extracting information from documents, text summarization, and training machine learning classifier models on text data.

A. Literature Review on Information Extraction

Text extraction from text requires word representation for a machine to interpret the text and be able to do word-based searches for extracting information. The language model is often built to translate from text to numbers. In the past, NLP developed its models using a wide range of techniques; these models were used to assign probabilities, frequencies, or other cryptic values to specific words, word groups, word sequences, paragraphs of text, or entire texts.

The statistical language modelling method has been employed to learn the joint probability function of word sequences. The curse of dimensionality created a problem because a word sequence on which the model will be tested will likely differ from all the word sequences seen during training. This problem is solved by learning a distributed representation of words that allows the model to learn about an exponential number of semantically related phrases from each training phrase [6]. Further, more research was done to comprehend language's syntactic and semantic regularities [7].

Initially, the question-answering models could answer approximately 40 per cent of answers correctly by using these techniques where the process allows vector-oriented reasoning based on the offsets between words. Additionally, the author mentioned how the word vectors' size and the training data volume affect training time and accuracy. Evaluated the quality of these vector representations by comparing the word similarity between the word vectors [8]. Also compared the results with several alternative methods based on different kinds of neural networks. Later, the BoW model was employed to extract information based on the frequency of words. It is a fundamental model where Each element represents the normalized number of occurrences of a basic term in the document. Despite being widely utilized, BoW has two disadvantages they ignore word semantics and lose the word order [9].

Researchers have developed an unsupervised algorithm that automatically creates vector representations of phrases and text documents. The dense vector used to represent each page is trained using this technique to anticipate the words it will include. Results demonstrated that this method of text representation outperformed bag-of-words models and was capable of performing new tasks, including text categorization and sentiment analysis. Alternatively, due to its inherent severe sparsity, large dimensionality, and inability to capture high-level semantic meanings behind text data, BoW representation suffers [10]. Researchers proposed fuzzy Bag-of-Words (FBoW), a document representation technique to get around the challenges. The backbone of FBoW's fuzzy mapping is the cosine similarity between word embeddings, which assesses the semantic link between words.

Deep models' introduction in this area offered a new language representation paradigm, particularly context-dependent encoders like BERT (Bidirectional Encoder Representations from Transformers), which were aimed to pre-train deep bidirectional representations from the unlabeled text [11]. Therefore, the pre-trained BERT model can be improved by adding one more output layer to produce models that can perform diverse NLP tasks. To create a summary of earlier work, surveys were conducted with practitioners who wanted to know more about using transformers to solve text ranking problems [12]. They gave a general review of text ranking using the most well-known transformer neural network design, the BERT. The combination of transformers and self-supervised pretraining has resulted in a paradigm change in several domains, including natural language processing and information retrieval. Most of the strategies they covered can be generally classified as dense retrieval techniques, which directly do ranking, and second transformer models, which perform reranking in multi-stage structures. Information extraction in clinical settings was conducted through building databases [13]. Data from 184,634 clinical trials were extracted and built into a protocol retrieval system from individual protocols to give a more thorough search.

Since clinical studies take a while to complete, it is advantageous to study similar earlier clinical trials while creating a new clinical study. The study was conducted to examine Trial similarity search. The length of the trial documents and the lack of labelled data presented a challenge for this investigation [14]. The researcher then suggested the Trial2Vec technique, which allows self-supervision without annotating related clinical trials. Trial2Vec additionally encrypts documents while considering meta-structure, resulting in compact embeddings that assemble multi-aspect data from the entire document.

B. Literature Review on Machine learning models on Clinical data

In pharmaceutical research, clinical trial success or failure probabilities are frequently calculated using straightforward algorithms based on historical data. Tree-based classification techniques like the decision tree, boosted decision trees, random forest algorithm, and bayesian additive regression tree was appropriate because it is a classification problem, and non-linearities and interactions between features are expected [15]. So, each observation is given a success or failure status at the terminal nodes. The limitations and potential benefits of applying machine learning to clinical trials were also discussed, along with the difficulties (philosophical and practical barriers) that may arise [16]. Additionally, it offers suggestions for overcoming some of the most typical challenges encountered throughout this process and advice on the finest machine learning approaches. The author emphasized the value of machine learning for pre-trial clinical trials, cohort selection, participant management, and data collecting and analysis. Research is conducted for diseases, particularly breast cancer the number of deaths is high as the symptoms are not detected until the last stage is reached [17]. The research was conducted on the usage of Machine Learning to predict Breast Cancer symptoms in earlier stages to start the treatment on time by Clinicians to save the patient's life. A relative study between three algorithms (Logistic Regression, Support Vector Machine, Naive Bayes) proves the hypothesis considered and answers the research question which states that different ML algorithms have a distintion in their performance and accuracy, among all SVM giving the most accurate results for the experiment.

The recent COVID-19 pandemic has also unlocked further the opportunities of technology in the field of Clinical trials. Identifies and elaborates some of the challenges faced in Clinical trials during the pandemic and suggests ways of using ML to assist in solving such problems [18]. The areas for challenges taken into account include – Clinical trials being conducted for non-COVID-19 related drugs, Clinical trials to use existing available drugs to treat COVID-19, and Clinical Trials to produce new drugs for the treatment of COVID-19.

C. Literature Review on Text summarisation

Clinical trial descriptions can be reduced into short summaries using text summarization, which can cut down on time needed to become familiar with the topic of the study [19].

Following techniques were discussed, the PageRank algorithms such as LexRank use cosine similarity over the TF-IDF-normalized word vectors, whereas TextRank uses word co-occurrences [20] [21]. Latent semantic analysis technique uses to convey the significance of each phrase in text summary determined by the presence of word combination patterns. The Luhn technique assigns a significant factor to each sentence [22]. The number represents the total of the keyword's position in the sentence, and its related relevance is indicated by its frequency. Contrarily, the SumBasic method merely gives each phrase a weight depending on the percentage of times a word appears in sentences compared to all other terms in the text. The summaries produced using these techniques were evaluated using ROUGE-Scores, and it was found that the TextRank method produced the best results for the summarizing task. Text summarization is predominant in the research field as it helps reduce the time consumed by going through lengthy documents or articles. A comparative study between extractive (TF-IDF, TextRank) and abstractive (Seq2Seq, Pointer-Generator) techniques using datasets of CNN/Daily Mail and Gigaword is evaluated on a score from Rogue metrics [23]. The results show extractive summarization techniques score well perhaps they sound more artificially generated by robots whereas abstractive techniques generate human-like summaries but with a greater cost of processing.

In recent times transformers have been used to summarize text [24]. Still, Long sequences can't be handled well by

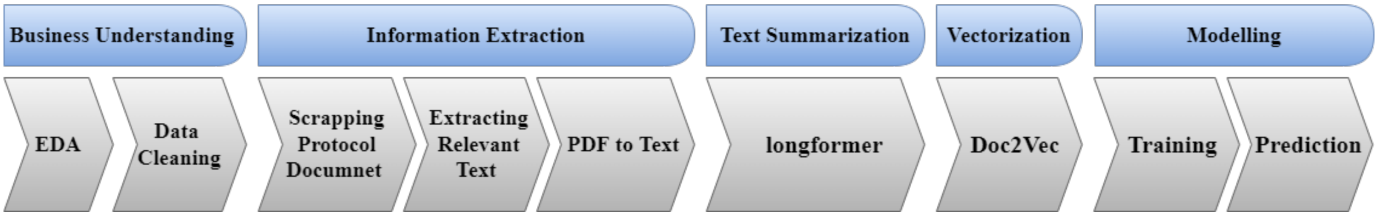


Fig. 1. Pipeline

simple transformer-based models because their self-attention operation scales quadratically with length. As a result, the Longformer transformer, which features an attention mechanism that increases linearly with sequence length, was proposed it replaces the typical self-attention by combining local windowed attention with task-driven global attention and seen that pretrained Longformer frequently outperforms RoBERTa on a large document.

Therefore, in this research, we tried to implement a longformer to summarize the protocol document and utilize the abbreviated text to predict completion status using a classification machine learning algorithm.

III. Methodology

Considering the in-depth investigation and analysis from the previous part, we created the necessary procedures for our project and organized them into a pipeline. The pipeline for our project is shown in Fig. 1, and each segment of the pipeline is covered in the section following.

A. Dataset

The data is collected from https://clinicaltrials.gov/. ClinicalTrials.gov is a massive database of privately and publicly financed clinical trials from across the world. Discover over 4 lakh research studies globally from 50 states and 221 countries. ClinicalTrials.gov is a service of the National Library of Medicine in the United States. Clinicaltrials.gov has grown at an astounding rate since its establishment in 2000. Many clinical studies are reviewed, monitored, and approved by an Institutional Review Board (IRB). It is an impartial group comprising physicians and statisticians. Their responsibility is to ensure that the research is ethical. A clinical trial in the United States must have an IRB if it is examining a medicine, biological product or medical device regulated by the Food and Drug Administration (FDA), or if it is financed or carried out by the federal government.

In the database, 327 tables use nct id as a unique identifier. The total data available is 4,23,077. Out of 4,23,077 entries, only 23,282 had Study Protocols, Statistical Analysis Plans (SAPs) and Informed Consent Forms (ICFs) documents. As the research is focusing on protocol documents, the database contains in total 21, 745 records of study protocol documents for all diseases. The following step was to investigate each disease and choose one for the project.

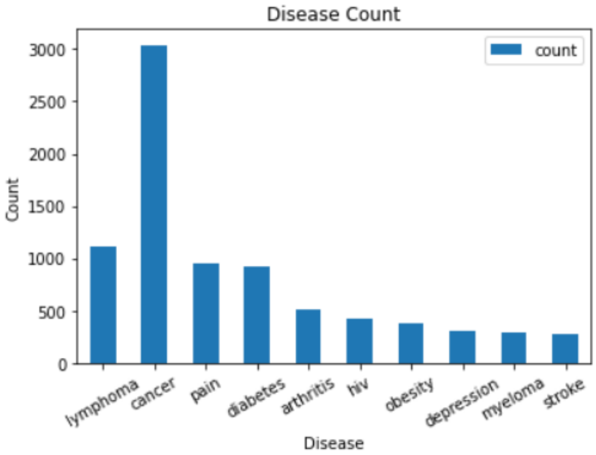


Fig. 2. Document count of disease

The platform used in this project is the Google Colab Cloud-Based platform with Tensor Processing units and python as the computational language. The process begins with understanding, cleaning, exploring, tokenizing, summarizing and transforming the dataset

B. Exploratory Data Analysis

While performing Exploratory analysis initially created a dataframe of diseases and the count of protocol documents for those specific diseases. It was found that there were 13706 unique diseases present in the database. Later grouped the diseases into one meaning, and combined all the derivatives of cancer into one single word cancer.

The top 10 unique diseases were considered for further analysis. Fig 2 shows a clear description of different diseases and their number of documents. Since the clinical trial data is heterogeneous, hence it was decided to focus on diabetes as it has a maximum number of protocol documents. Diabetes is examined globally and is a well known disease with a mixed population all over the world. Out of 21, 745 records which contained protocol documents only 970 diabetes had the protocol document.

While cleaning the data we initially focused on creating target vectors. As the research classifies whether the trial will be completed or not completed, the status column is the best fit for creating target vectors. Fig 3 shows distribution of the trial in different status such as ’Completed’, ’Recruiting’, ’Active, not recruiting’, ’Terminated’,’Unknown status’, ’Not yet recruiting’, ’Enrolling by invitation’, ’Withdrawn’, ’Suspended’ unique values. The ’completed’ values were kept as it is and changed the remaining values were changed to ’Not completed’ status.

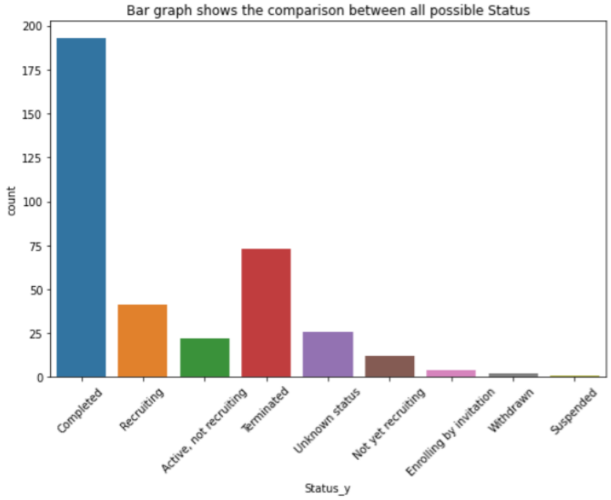


Fig. 3. Status attribute values

Basic text cleaning steps were implemented on protocol documents as well as metadata. Extracted the PDF link from the document column and stored in the dataframe. At last created a dataframe consisting of ‘NCT numbet’, ‘Status’, ‘Conditions’, ‘Sponsor/Collaborators’, ‘Age’, ‘Sponsor/Collaborators’, ‘Study Design’, ‘Location’ and ‘’Links.

C. Information Extraction

EDA supported in correctly analysing the data, and the data cleaning technique yielded solid 970 records of data for further pipeline processing. The pipeline will then go on to protocol papers since the next phase will be to extract meaningful content from each of these 970 records. Fig 4 shows the frequency distribution of page numbers since the pdf pages range from 2 to 250 pages, not every page is required. Sending all text would cause an imbalance while giving input for the machine learning model. To ensure that each pdf delivers stable and relevant data, text extraction is an essential aspect of the pipeline. As all the protocol documents did not follow a particular format also the page number varies thus it was difficult to extract specific and same content from the pdf.

To obtain the protocol pdf from the URL in the metadata, the protocol document is parsed by hitting each URL from the HTTP request for each trial. Then, because the protocol papers were in pdf format, the PyPDF2 library is used to execute operations on them. First, is used PdfFileReader, a method called getFormTextFields(), to extract text data from the PDF. This method retrieves text data and displays it in dictionary format. Then there was the pdf document, from which using the pdf document the pages were extracted with the highest number of the requisite terms.

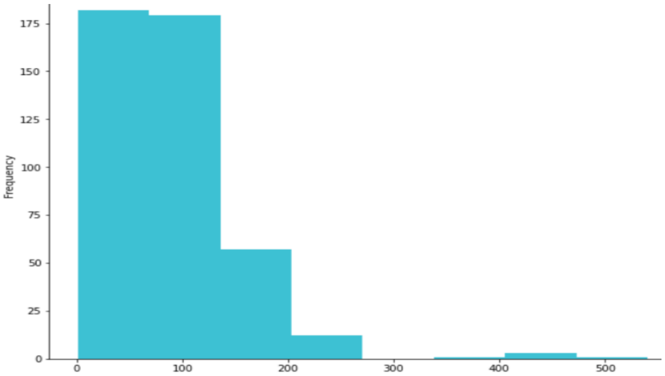


Fig. 4. Page length distribution

Algorithm 1 describes the logic followed in this paper to extract information. According to the logic first, created a list of thirteen critical terms from the clinical trial webpage [4]. Next, a list of page number was created and calculated the total count on that page for each terminology using the findall() method, which searches the string from left to right to find all matches of the pattern in the string. Finally, selected a single page for each terminology with the highest count for that keyword and extracted the text using the extractText() function, which is used to extract text from PDF files. After that, saved collected text in a data frame for further processing for each terminology.

Algorithm 1 An algorithm for Information Extraction

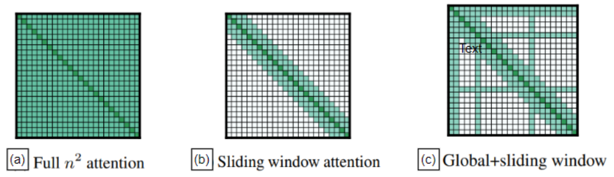
```
Require: Study Protocol pdf and Keyword list
  initialize list = []
  for keyword in keyword list do
    for page in pdf do
      current_page = pdf.getPage(page)
      text = current_page.extractText()
      # keyword search using findall() function keyword is
      in page
      count = len(findall(keyword,text)
      list.append(count,page) keyword not found
    end for
  end for
```

D. Summarize Text

Once the dataframe of relevant text is ready, the first phase began of our research by summarizing the text, which will subsequently be utilized for prediction.Here, different text summarization methods were tried: BERT (Bidirectional Encoder Representations from Transformers), GPT-2(Generative Pretrained Transformer 2) and longformer. Classic transformers divide the document into chunks to process and then aggregate it. Thus, the disadvantage of the classic transformer is that it cannot connect the words of different chunks on a neural level. In contrast, longformer aims to accept the document without divide into chunks.

Thus, the length of text that the traditional transformers can process at once is limited, but the longformer can handle lengthy documents; therefore, it was incorporated it into phase one [25]. Moreover, the longformer can handle up to 4096 tokens, eight times BERT and four times GPT-2 models.

The longformer [24] has introduced a sparse attention mechanism which reduces the computational time and memory requirements. Longformer is a BERT-like model built from the RoBERTa(Robustly Optimized BERT Pre-training Approach) checkpoint and pre-trained on lengthy texts for MLM (multilevel modelling). Transformer-based models scale quadratically with the sequence length, whereas longformers scale linearly with the sequence length. So longformer is basically what CNN does for MLP(Multi-Layer Perceptron) same it does for transformers. Thus, instead of O(n*n), longformer has O(n*w) where w is the window size.



(a) Full n^2 attention (b) Sliding window attention (c) Global+sliding window

Fig. 5. Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer

To illustrate further, the comparison of the composition of attention patterns in longformer with the full self-attention pattern and sliding window pattern is shown in Fig 5. The classic transformer with full self-attention shows that all n units in the sequence can attend to all other units in the sequence, so the complexity of this pattern will be O(n*n). Whereas Fig 5(b) shows the sliding window pattern, there is a window of fixed size. So, in this case, the n unit can attend only the units present adjacent to it. It cannot attend to the unit outside the window slide. In addition to the pattern of the sliding window the longforemer follows the global attention-based pattern. There are some special tokens involved that can attend any unit at every layer; this enables route to process the information to any of the other units as seen in Fig 5(c). These special tokens have global attention. So the memory requirement we have for n units and w window size will be O(n*w) which is less than the classic transformers.

In this study, we employed the Longformer2RoBERTa architecture, which uses longformer as the encoder and RoBERTa as the decoder. Therefore, two classes are required, LongformerTokenizer class for tokenization, while the EncoderDecoderModel class is needed for Seq2Seq(Sequence to Sequence). We have used the ”patrickvonplaten/longformer2roberta-cnn-dailymail-fp16” instance for our model as it contains the Seq2Seq model. Longformer was used for the encoder segment, so longformer tokenization was used for it. For the longformer tokenizer, we used the ”allenai/longformer-base-4096” pretrained model. When the text from the data frame was passed to the tokenizer, it created input ids using PyTorch-based Tensors. The model used these input ids to generate the summary. Once the summary is ready, it is decoded into the readable form using the tokenizer again. A summary of four lines was created for each page.

one terminology → page → 4 lines summary.

Later, all of the summaries were combined into one. As a result, a single summary of a single trial is created.

E. Doc2Vec

Passing the summarised text of the protocol document along with metadata as it is to the machine learning model for prediction purposes is impossible. This is because the machine learning models process only numeric data. As machines cannot understand human language, the text needs to be converted into numbers. There are various methods for translating text to numbers, including MCA multiple correspondence analysis, ordinal encoding, and one-hot encoding. However, each technique supports a small amount of categorical data; thus, we must apply NLP to embed long texts into numbers. This process is referred to as text embedding.

Many different approaches were tried to convert the data into embedding forms with Doc2Vec, BERT, RoBERTa, and neural network models (LSTM, CNN). We implemented Doc2Vec because the BERT model has a constraint in the form of a maximum number of input tokens compared to the Doc2Vec model [26] [27]. The Doc2Vec model uses either the Word tokenizer or the Sentence tokenizer [28] [29]. For each word in a text document, the doc2vec model creates a word vector and a document vector, denoted by W and D, respectively. Moreover, the model assigns the weights for softmax in the hidden layer during training so that it can produce a document vector for new documents.

Hence, we found the Doc2Vec pre-trained model suitable and relevant for our modelling compared to other methods. Here, gensim stands for Generating Similar it determines the semantic structure of text data. It is a Python package used for tasks related to natural language processing, including unsupervised topic modelling, text embedding, and text similarity analysis. Fig 6 depicts the framework of the gensim Doc2Vec model. Word tokens are generated for the text, and tags are assigned to each document to create vectors. The tokenized sentences were passed into the Doc2Vec pretrained model to provide a fixed-size vector. As a result, the model produces a (374, 5)-dimensional output vector. The vectors obtained from Doc2Vec will be passed to the next stage as an input feature to the modelling stage.

Loaded the genism Doc2Vec pre-trained model. The tokenized sentences were passed into the pre-trained model for

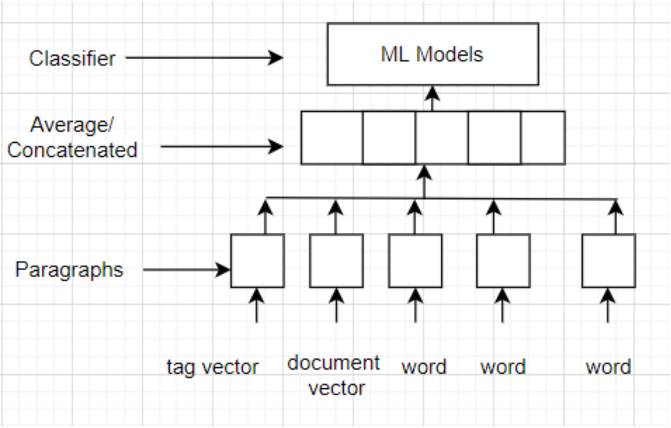


Fig. 6. Framework for Learning Paragraph Vector

providing a fixed-size vector. After applying the genism model to the tokenized text, it provided the (374, 5) shape of the vector.

F. Modelling

Machine learning describes a computer algorithm that ”learns” to perform better in a specific activity. Machine learning algorithms can be broadly classified into supervised and unsupervised methods. Unsupervised learning is used to examine and cluster sets of unlabeled data. These algorithms discover hidden patterns in the data, whereas supervised machine learning methods are applied for labelled datasets. In this study, the machine learning model’s objective is to forecast the target attribute’s values. Our target attribute is status, which has two possible values: completed and not completed, denoted by binary numbers 0 and 1, respectively. It is a machine learning classification problem because we want to classify the target vector as either 0 or 1.

The three steps of the supervised learning process that we followed are shown in Fig 7. Initially, the dataset must be randomly split into training and validation datasets. The first set serves as a training set that includes 80 percent of the data, and the second serves as a validation set that consists of the remaining 20 percent. Training sets are used to estimate parameters or compare other models’ performances.

In the learning stage, the machine learning algorithm continuously assesses input and target pairs from the training set, further calculates a target value for each pair, and evaluates it against the actual value. A resampling technique called cross-validation is used to assess machine learning models on sample data. The k parameter in this technique specifies how many groups should be formed from the given data. For example, when the value of k is assigned as ten, ten samples are produced from the training data. Among the ten samples, nine serve as training sets and one as validation. Consequently, ten groups are formed. For each group, the training set is used to fit a model, while the validation set is used to evaluate it. Finally, save the evaluation result, then select the model with the best result.

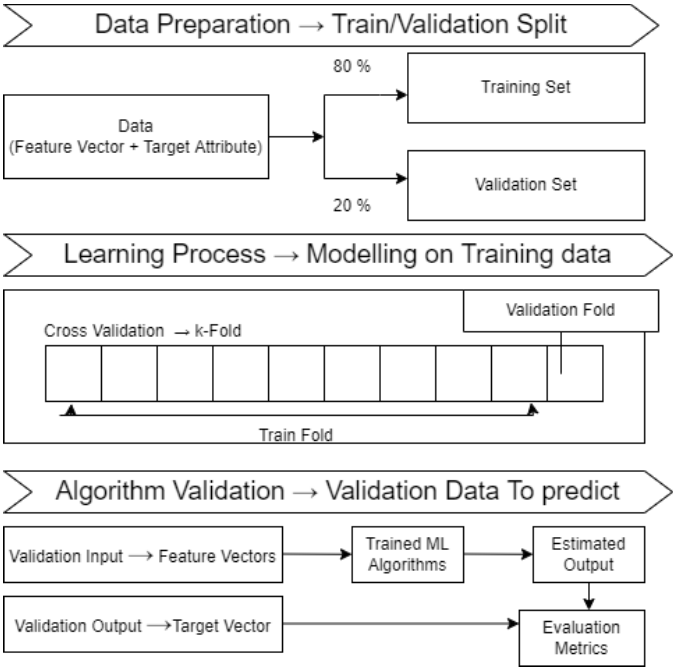

Data Preparation → Train/Validation Split
80 %
Training Set
Data
(Feature Vector + Target Attribute)
20 %
Validation Set
Learning Process → Modelling on Training data
Validation Fold
Cross Validation → k-Fold
Train Fold
Algorithm Validation → Validation Data To predict
Validation Input → Feature Vectors
Trained ML Algorithms
Estimated Output
Validation Output →Target Vector
Evaluation Metrics

Fig. 7. Result table

After the learning process, the model is created to test the model validation dataset. In the third stage, validation data is passed to the machine learning model as an input, and the trained machine learning model will predict the output. Later, this output is compared with the actual output of the validation set. It must reflect the output values of the validation set with enough accuracy for its intended use to be verified successfully.

We can build numerous machine learning models for clustering, classification, and regression using libraries from Sklearn, and we can utilize statistical techniques to assess these models. We implemented four machine learning models for classification using the sklearn library. The models used in this study are Logistic Regression, Decision Tree Classifier, Random Forest Classifier and Support Vector Classifier.

A decision tree is a rule-based classification technique that uses non-parametric supervised learning. In other words, it makes decisions based on a set of rules. The logic is to build yes or no questions using the dataset’s input attributes, split the dataset repeatedly, and then isolate all the data points corresponding to each class. Finally, the data is arranged in a tree-like layout using this process.

A random forest classifier is an ensemble supervised learning algorithm that uses multiple learning algorithms to obtain better predictive performance. A Random Forest Classifier uses numerous independent decision trees that perform together. Each tree in the random forest produces a class forecast, and the classification that receives the most votes becomes the prediction made by that model.

An algorithm for predictive analysis called logistic regression is based on the idea of probability. It uses the sigmoid function to convert its output and produce a probability value. Like linear regression, logistic regression represents data using an equation. To forecast an output value, input values are combined linearly with weights or coefficient values. Any real-valued number can be mapped onto a value between 0 and 1 using this S shaped curve.

In Support Vector Classification, each data point is represented as a point in n-dimensional space, where n is the number of attributes. The algorithm then carries out classification by locating the line that successfully differentiates the class labels. This line, also known as the hype plane, serves as the decision boundary; anything falling on one side will be labelled as ”Completed,” while anything falling on the other will be labelled as ”Not Completed.” Furthermore, after executing the above four models, we attempted to find the model that produces the best outcomes for our scenario.

IV. Results

Two different methods were used for evaluation. First conducted k-fold cross validation on the dataset to obtain the accuracy, and then we obtained the accuracy without doing so. The random forest had the highest accuracy among both methods compared to the other three models. However, accuracy was higher when using k-fold cross validation compared with the accuracy without k-fold cross validation. In k-fold cross-validation, the model is trained with different chunks of data, making the model more accurate when a new chunk is executed.

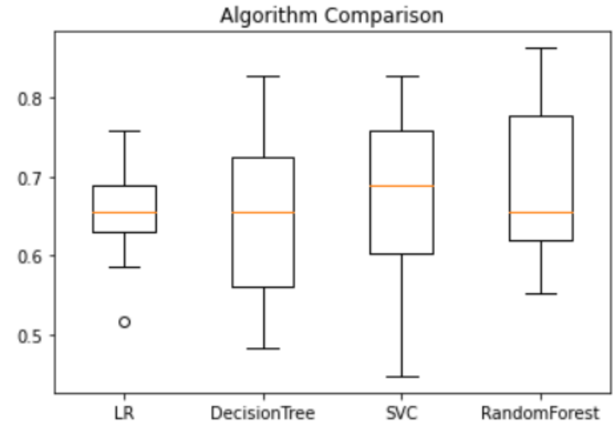
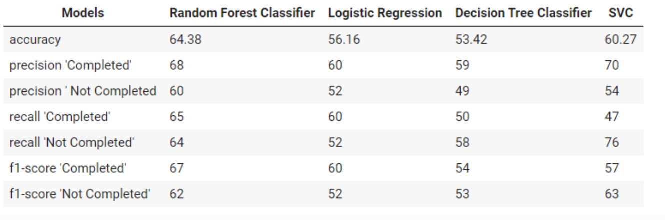


Fig. 8. ML model Comparison boxplot

A summary of the evaluation using k-fold cross validation is shown in Fig 8, utilizing a box plot, whereas the assessment without k-fold is shown in Fig 9. The Random Forest Classifier has the highest accuracy (with cross-validation as 68.96 and without cross-validation 64.38), and the Decision Tree has the lowest accuracy (with cross-validation as 64.82 and without cross-validation 53.42), in both the methods among all the four different models. We aimed to predict whether the trial would be ’Completed’ or ’Not Completed’. The training was done using a balanced dataset, which helped provide a decent number of results.

Fig 9 illustrates that SVC performs better with a precision metric of 70 percent while identifying actually ’Completed’ trials among the predicted ’Completed’ trials. On the other hand, RFC performs better with a precision metric of 60 percent while identifying actual ’Not Completed’ trials among the predicted ’Not Completed’ trials. This categorisation is crucial, as trials that tend to be unsuccessful in the future should not be funded in the first place, as they might incur a loss of funds.

Fig 9 illustrates that RFC performs better with a recall metric of 65 percent while identifying ’Completed’ trials among all the ’Completed’ trials. In contrast, SVC performs better with a recall metric of 60 percent while identifying “Not Completed” trials among all the ’Not Completed’ trials. This categorization is crucial, so we don’t tag a ’Completed’ trial as ’Not Completed’, resulting in missing the perfect cure for the disease.



Models	Random Forest Classifier	Logistic Regression	Decision Tree Classifier	SVC
accuracy	64.38	56.16	53.42	60.27
precision 'Completed'	68	60	59	70
precision ' Not Completed	60	52	49	54
recall 'Completed'	65	60	50	47
recall 'Not Completed'	64	52	58	76
f1-score 'Completed'	67	60	54	57
f1-score 'Not Completed'	62	52	53	63

Fig. 9. Result table

V. Conclusion

The problem statement stated above aimed to predict the trial’s success and failure achieved by implementing the pipeline mentioned in fig 1. There are very few research papers available on this topic of predicting the completion of the trial. Those papers did not use a protocol document summary for prediction in the manner we performed the implementation.

We successfully implemented the pipeline on the collected dataset. Extracting the important parts from the protocol document was one of the toughest challenges we faced, as the documents were not in the same format. Clinical trial boards should provide the same and strict format for every protocol document, which should be mandatory. Fixing the input will surely yield better results in the output. Extracted information was passed to the longformer for summarization, as finding results from the whole document was challenging and lengthy. This summary, along with the metadata, was not in the machine-readable format; therefore, we vectorized it so that we could pass those vectors to machine learning models. Compared four different ML models results.

We have received remarkable results, which can be further improved with ongoing trials and research on the same subject. This idea can be implemented and brought into action using more extensive datasets, which will surely change the clinical trial industry.

VI. Future-work

Clinical trials are a vast area for research and development. The proposed model will surely boom the clinical trial industry. Apart from the methods mentioned earlier, techniques and models, different approaches related to machine learning, deep learning and neural networks can be used to evaluate the best method for accurate results.

We have used protocol documents to predict the success and failure of the trial. Instead, metadata for the website can be used to indicate trial completion. A comparative study can be performed on these two methods to identify which procedure gives better results. In addition, research on information extraction can be conducted to extract essential paragraphs from the whole protocol document instead of the whole page from the pdf.

We have used an abstractive text summarization method where the summary is generated by extracting important sentences and then manipulating the text to form the summary, where there is a chance of loss of information. Thus extractive text summarization can be used in further studies where important sentences are not manipulated but kept as it is; therefore, there is no chance of information loss. Evaluation of text summarization could be performed to check the quality of the summary and whether it makes the same sense as the text or not. In-depth research could be conducted on techniques to evaluate text summaries.

Some clinical trial documents were in pdf format, but the search function was not working. After digging around into it, we found that it was an image in .pdf format. Thus finall() function was not working. Alternative techniques could be found to tackle this problem.

VII. Acknowledgement

Professor Tomas Ward gave us an opportunity to work in an area which is not much explored and has the potential to grow, which resulted in extensive exposure to clinical studies and to learning the practical application of the Machine Learning. Himanshu Vashisht industrial supervisor had offered direction throughout the task and answered all of the questions thoroughly. Brainstorming on various ideas and techniques with my peer resulted in a terrific exploration and information transfer.

References

[1] Arun Bhatt. Evolution of clinical research: a history before and beyond james lind. *Perspectives in clinical research*, 1(1):6, 2010.

[2] Emma M Nellhaus and Todd H Davies. Evolution of clinical trials throughout history. *Marshall Journal of Medicine*, 3(1):41, 2017.

[3] Harshani Yarlagadda, Meet A Patel, Vasu Gupta, Toram Bansal, Shubekshya Upadhyay, Nour Shaheen, and Rohit Jain. Covid-19 vaccine challenges in developing and developed countries. *Cureus*, 14(4), 2022.

[4] World preview 2019, outlook to 2024. Accessed 10 December 2019.

[5] David B Fogel. Factors associated with clinical trials that fail and opportunities for improving the likelihood of success: a review. *Contemporary clinical trials communications*, 11:156–164, 2018.

[6] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, mar 2003.

[7] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[10] Rui Zhao and Kezhi Mao. Fuzzy bag-of-words model for document representation. *IEEE Transactions on Fuzzy Systems*, 26(2):794–804, 2018.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021.

[13] Junseok Park, Seongkuk Park, Kwangmin Kim, Woochang Hwang, Sunyong Yoo, Gwan-su Yi, and Doheon Lee. An interactive retrieval system for clinical trial studies with context-dependent protocol elements. *PloS one*, 15(9):e0238290, 2020.

[14] Zifeng Wang and Jimeng Sun. Trial2vec: Zero-shot clinical trial document similarity search using self-supervision. *arXiv preprint arXiv:2206.14719*, 2022.

[15] Bernard Munos, Jan Niederreiter, and Massimo Riccaboni. Improving the prediction of clinical success using machine learning. *medRxiv*, 2021.

[16] E Hope Weissler, Tristan Naumann, Tomas Andersson, Rajesh Ranganath, Olivier Elemento, Yuan Luo, Daniel F Freitag, James Benoit, Michael C Hughes, Faisal Khan, et al. The role of machine learning in clinical research: transforming the future of evidence generation. *Trials*, 22(1):1–15, 2021.

[17] Neetu Sangari and Yanzhen Qu. A comparative study on machine learning algorithms for predicting breast cancer prognosis in improving clinical trials. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 813–818. IEEE, 2020.

[18] William R Zame, Ioana Bica, Cong Shen, Alicia Curth, Hyun-Suk Lee, Stuart Bailey, James Weatherall, David Wright, Frank Bretz, and Mihaela van der Schaar. Machine learning for clinical trials in the era of covid-19. *Statistics in Biopharmaceutical Research*, 12(4):506–517, 2020.

[19] Christian Gulden, Melanie Kirchner, Christina Schüttler, Marc Hinderer, Marvin Kampf, Hans-Ulrich Prokosch, and Dennis Toddenroth. Extractive summarization of clinical trial descriptions. *International journal of medical informatics*, 129:114–121, 2019.

[20] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[21] Rada Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL interactive poster and demonstration sessions*, pages 170–173, 2004.

[22] Josef Steinberger, Karel Jezek, et al. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4(93-100):8, 2004.

[23] Pooja Raundale and Himanshu Shekhar. Analytical study of text summarization techniques. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–4. IEEE, 2021.

[24] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[25] Théo Ding, Walter Vermeiren, Sylvie Ranwez, and Binbin Xu. Improving patent mining and relevance classification using transformers. *arXiv preprint arXiv:2105.03979*, 2021.

[26] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[27] Marko Pranjić, Marko Robnik-Šikonja, and Senja Pollak. An evaluation of bert and doc2vec model on the iptc subject codes prediction dataset. 2020.

[28] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[29] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.