# BERT-based Extractive Text Summarization using Unsupervised Machine Learning in the biomedical domain

Yufei Gu
Department of Computer Science
University College London
London, UK
yufei.gu.20@ucl.ac.uk

Sahan Bulathwela
Department of Computer Science
University College London
London, UK
m.bulathwela@ucl.ac.uk

David Cox
NHS
London, UK
david.cox@hee.nhs.uk

Chaudhary Siddharth
Department of Computer Science
University College London
London, UK
siddharth.chaudhary.20@ucl.ac.uk

Zihao Feng
Department of Computer Science
University College London
London, UK
zihao.feng.20@ucl.ac.uk

## ABSTRACT

During the pandemic, NHS with all countries' medical systems are under pressure and there is an increasing demand for efficient information capture and transmission. In this paper, we introduces our unsupervised extractive text summarization system in the biomedical domain and evaluates different machine learning algorithms' performance on a sample data set.

## KEYWORDS

Extractive text summarization; Unsupervised machine learning; Biomedical text summarization

## 1. INTRODUCTION

The demand for remote consultations is growing exponentially these days, however, the documentation of remote consultations is very inefficient and the medical information appears rarely gets shared with patients and other medical teams. The information loss between conversations is another worrying topic. Difficulties often occur when patients try to understand the full details of the treatment and actions they are recommending. Human memory is not always reliable, and additional manual documentation will be a huge cost. To improve the quality and efficiency of clinical consultations, an efficient method of information retrieval and documentation is urgently needed.

Automatic text summarization is one method that has great potential in meeting such demand. *A summary* is defined as "a text that is produced from one or more texts, that contains important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that" according to Radef et al. [1] Summary helps people quickly understand the core concepts and key information contained in the text. *Automatic text summarization* is the task of automatically producing a concise and fluent summary. [2] It allows both patients and medical workers to save time on processing redundant information and can significantly improve reading efficiency. Automatic text summarization can also transform informal medical documents like doctor notes so they can be compatible with existing electronic medical record (EMR) systems.

Nowadays, many mature automatic text summarization techniques have been applied to various domains, in particular, snippets generated by search engines as web page previews and benefiting our lives. However, still not many automatic text summarization techniques are used in hospitals and medical services nowadays. This paper aims to: (1) Introduce the proposed method of our extractive text summarization system; (2) Evaluate different machine learning algorithms' performance on a sample dataset of biomedical documents.

The rest of the paper are organized as follows. A overview of related works is presented in Section 2. The introduction of our proposed method is presented in Section 3. The experiment design and evaluation methods are introduced in Section 4. The results of the experiment and limitations of our system are presented in Section 5. Finally, a conclusion of this research and future exceptions of our project are presented in Section 6.

## 2. RELATED WORK

A huge amount of attempts and methods have been developed in the text summarization field since the early 1950s and these studies can be divided into different categories in many different ways.

*Extraction* and *Abstraction* are the two main types of automatic text summarization methods. Extractive Summarization works by identifying key tokens and sentences from the text source and reorganizing them into the summary, while Abstractive Summarization generates summaries in human language which contains the key information through 'understanding' the original text.'A summary is reliable only if it is true to the original. Abstractive summarizers are considered to be less reliable despite their impressive performance on benchmark datasets because they can hallucinate facts and struggle to keep the original meanings intact.' [8] [9] Because a production system should be highly reliable and fluent language is not our priority in medical information summarization, we focused on extractive text summarization in this research and for our project.

Regarding the number of input documents being processed at the same time, text summarization methods can be single-document or multi-document. [14] In this research, we mainly focused on single-document summarization.

According to the summarisation algorithm, the text summarization technique can be supervised or unsupervised. The supervised algorithm needs a training phase which requires labeled data, while unsupervised algorithm needs no additional training phase nor labeled data. Because we didn't have labeled biomedical text data for this research and are not able to collect private data, we focused on unsupervised text summarization techniques in this research.

There are other classification methods based on the nature of the summary, the summary language, summary content and summary type. [15] The txt summarization method presented in this method is overall single-document, extractive, generic, monolingual, unsupervised, informative, generic, and specific for biomedical domain.

Most of the existing extractive text summarization systems use statistical, probabilistic, concept-based, topic-based, graph-based, machine learning, and hybrid approaches. [15] [16] Statistical methods extracts important sentences from the paragraph based on statistical analysis on a set of features. Probabilistic methods leverage the probability distribution of words, concepts, and topics within the text to approximate new probability distributions for potential summaries. The probability distribution of the final summary is adjusted to the original text or follow the distribution of essential concepts and topics. [17] [18] Concept-based and Topic-based methods extracts biomedical concepts from the input documents and employs an itemset mining algorithm to discover main topics. [19] Graph-based Methods construct a weighted directional on representing the source text and use transfer probability to identify key sentences. [20] Many machine learning techniques have been developed in the context of automatic text summarization system including classification [18], clustering [19] [21], neural network [22] [23], and optimization approaches [24] [25]. Hybrid methods with multiple approaches being combined and used have also being developed. [26]

Much effort has been made toward developing biomedical text summarizers [27] [28]. The dominant approach in biomedical text summarizers focused on extractive methods, but there are also a growing interest in abstractive text summarization and graph-based methods. Recent research has focused on a hybrid technique comprising statistical, language processing and machine learning techniques. [27]

Various datasets have been developed to evaluate text summarization systems, DUC-2004 and CNN/Daily Mail dataset are the two most popular dataset. State-of-the-art methods obtained ROUGE-1 scores around 0.33 and ROUGE-2 scores around 0.12 on DUC-2004 [29] while ROUGE-1 scores achieved around 0.44 and ROUGE-2 scores around 0.20 on CNN/Daily Mail [30]. However, it is important to mention that the ROUGE scores achieved by text summarization system largely depends on the size of the summary relative to the source text (compression rate) for generic text summarization systems. Comparison between different text summarization system is a difficult problem with different experiment parameters being set.

## 3. PROPOSED METHOD

In this section, we described the proposed framework of extractive text summarization and the various machine learning algorithms we tested in this research. Four main steps are included in our summarization process: (1) Text Preprocessing; (2) Feature Extraction; (3) Machine Learning; (4) Summary Generation. A detailed introduction of each step is provided below.

### 3.1. Text Preprocessing

Preprocess is usually an essential step to clean the input data before performing machine learning techniques. In this method, the goal of preprocessing is to get sentences and tokens for pre-trained BERT models to perform feature extraction. In order to complete this task, we used the Natural Language Toolkit (NLTK) library to separate sentences and split tokens from each sentence. The source text is stored in a temporal file with separate sentences and another temporal file stores every sentence from the source text in the format of a number of split tokens. The two temporal files are required by the BERT feature extraction script in the next step.

### 3.2 Feature Extraction

As mentioned before, extractive text summarization generates summaries by selecting important sentences from the original text input. So preprocessing on the original text and sentences such as feature extraction is required to get a mathematical representation of sentences so machine learning algorithms can be performed on the representations.

In this research, we use pertained BioBERT models to extract features from the original text. BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right contexts in all layers. [12] Pre-trained BERT models are conceptually simple and outperform previous methods for pre-training NLP tasks. BERT is an unsupervised system which means that it was trained using only a plain text corpus and this is extremely important because this means that BERT can make use of the enormous amount of plain text being spread on the internet.

In certain cases, rather than fine-tuning the entire pre-trained model end-to-end, it can be beneficial to obtain *pre-trained contextual embeddings*, which are fixed contextual representations of each input token generated from the hidden layers of the pre-trained model. This should also mitigate most of the out-of-memory issues. [13] We use the script extract_features.py developed by J. Devlin et al.[12] to get BERT activations from each Transformer layer. The vector representation of each sentence is computed by calculating the sum of the weight list of all tokens in this sentence. After we convert each sentence into vector format, we can perform machine learning on the mathematical datasets.

## 3.3 Machine Learning

There are three main types of machine learning algorithms for extractive text summarization: cluster analysis, graph-based algorithms, and deep learning approaches. In this research, due to the lack of labeled data for supervised learning, we mainly focused on unsupervised machine learning algorithms between clustering and TextRank (graph-based).

### 3.3.1. Cluster Analysis

Cluster analysis or clustering is the task of grouping a set of objects or data so that objects in one cluster are more similar to each other than those in other clusters. [3] Cluster analysis can be used to analyze the relationship of sentences and organize related sentences in a cluster. Many clustering algorithms have been developed over the last century designed for different kinds of models and we evaluate some of them in this paper.

### 3.3.2. Graph-based approach

We can use a graph-based approach to rank each sentence according to its importance in the text using the TextRank algorithms. TextRank originates from PageRank, an algorithm used by Google Search to rank web pages in their search engine results. PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites. [11]

## 3.4 Summary Generation

We selected sentences to be part of the final summary based on the machine learning algorithm outcome on the mathematical representations.

We first rank every sentence. TextRank algorithm directly returns a list of sentence scores, so no additional procedure is needed. For clustering algorithms, an additional step is required. We consider each cluster as a group of sentences with the same topic, and the sentence closer to the center of the cluster is considered the topic sentence of this cluster which contains the key information of this cluster. All sentences are sorted and ranked according to their distance to the center of the cluster it is assigned.

After sentences are ranked, the top sentences are selected to be the final summary. In the experiment, we set a compression rate to support adjustment of the length of summaries and the amount of information aimed to remain in the summary. The integer multiple of the number of sentences and the compression rate is the number of top sentences to be selected. After sentences are selected, they are outputted in the order of their original order in the source text. This is because keeping compliance with the logical order of the original article is the easiest way of ensuring the readability of the auto-generated summary.

# 4. EXPERIMENT DESIGN

In this section, we described our experimental methodology and evaluation methods. We also explained the system parameters we set for our experiments so the results of our experiments can be reproduced if anyone is interested in our research.

## 4.1 Datasets

Large and precise datasets have a significant impact on the training and evaluation of machine learning algorithms. Due to the sensitivity problem of patients' privacy, it is relatively hard to collect data in the clinical field. This results in a limited number of Clinical and Biomedical datasets for natural language processing tasks usually with a compliance audit, training requirement, and usage restrictions. Since we are not allowed to collect data for our research, we randomly select public unlabeled biomedical text from PubMed as our dataset. The PubMed database contains more than 33 million citations and abstracts of biomedical literature. [7] The corresponding full text is downloaded from the publisher's website or PubMed Central (PMC). We consider the abstracts from PubMed as the reference summary, which means the standard summary to its text source. A comparison between this reference summary and the machine-generated summary from the source text is being made and the evaluation method is introduced in the next session.

## 4.2 Machine Learning Algorithms

4 Classes of Machine Learning Algorithms and a total of 18 algorithms are selected for testing.

### 4.2.1 Centroid-based clustering

In centroid clustering, each member is assigned to the cluster with the smallest distance between its central vector and the vector representation of this member. In this research, 3 Variant of K-Clustering is selected and implemented as a representative of Centroid-Based Clustering.

- **K-medoids & K-means Clustering:** K-means algorithm might be the most popular machine learning classification algorithm. The classic k-means algorithm works in this way: First, k members are selected as the initial centers. Next, each member is assigned to the cluster with the closest center. After every member is assigned, the center of each cluster is re-calculated. If the clustering change, repeat to assign each member to the clusters with the new centers. If the clustering doesn't change, the algorithm will exit the loop. In this research, we implement k-medoids as it selects a certain sentence as the center of the cluster while k-means computes the mean mathematical vector as the center of the cluster. The iteration limit is set to 50, which means if re-clustering loops 50 times, it will exit the loop and return the result of the last clustering attempt.

- **Bi-k-means clustering:** Bi-k-means clustering is a variant of the k-means clustering algorithm. It is driven by the aim of minimizing the Sum of Squared Error (SSE) of each cluster. It first assigns all members in the same cluster, then recursively applies binary clustering to the cluster that can reduce the overall

SSE by a maximum scale until the number of clusters meets the number k.

There are many methods to compute the distance between two vectors. In this research, we evaluate three of them: Euclidean distance, Manhattan distance, and Cosine Similarity. Each centroid-based clustering is tested with all three distance measurement methods and their results are evaluated and discussed in the Results.

K-means clustering algorithm is an efficient clustering algorithm. However, it has several drawbacks. Firstly, the given k value will directly affect the clustering outcome. Secondly, because k-means is a non-convex optimization algorithm, it will converge to a local optimum. So it is very sensitive to the initial k centers. In this research, the first k sentences are selected as initial centers according to the lead-3 principle of text summarization. Finally, it is sensitive to noise points and abnormal data.

The code of three k-clustering algorithms is implemented according to their maximum and three distance measurements are chosen by the distance_num parameter.

Due to time constraints, we didn't review further centroid-based clustering methods like k-means++ in this research.

### 4.2.2 Connectivity-based clustering (hierarchical clustering)

Connectivity-based clustering is based on the core idea of objects being more related to nearby objects than to objects farther away. [3] In connectivity-based clustering, a cluster can be defined as the maximum distance needed to connect parts of the cluster. In this research, we mainly tested the agglomerative clustering algorithm.

- **Agglomerative Clustering:** Agglomerative Clustering is one of the most general clustering algorithms of connectivity-based clustering. It follows a bottom-up procedure. It first considers every data sample as a cluster. Then in each iteration, it merges two clusters with the closest distance (there are many ways to compute the distance between two clusters, and they are discussed in the next paragraph) until the number of clusters reduces to the requirement. Agglomerative Clustering can significantly reduce the chain effect but usually takes more time to process. Because it iterates through every member in every cluster in each iteration, it has a time complexity of O(N^3).

There are mainly three different methods to measure the distance between two clusters: Single-link, Complete-link, and UPGMA-link. Single-link computes the minimum distance between all members of two clusters. Complete-link computes the maximum distance between all members of two clusters. UPGMA computes the average distance between every member of two clusters. Three distance methods are implemented and separately tested in this research.

Due to time constraints, we didn't review other connectivity-based clustering methods like divisive clustering in this research.

The agglomerative algorithm is implemented with three measures of computing the distance between clusters: Single link, Complete link, and UPGMA.

### 4.2.3 Density-based clustering

In density-based clustering, clusters are defined as the space of high density of its members compared to the remainder of the given dataset. In this research, DBSCAN, OPTICS, and Mean-Shift are chosen among the Density-Based Clustering methods.

- **DBSCAN:** DBSCAN is an advanced clustering method. It marks all points as unvisited in the beginning. While there are unvisited points, it will randomly pick an unvisited point p and compute the number of points in its neighborhood. If there are at least M points, a new cluster will be created with the point p. All points in p's neighborhood are defined as a set N. For every point in N, if there are at least M points in its neighborhood, these M points are also added to N. Every point that are not assigned to a cluster in N will then be added to the cluster of p. If the number of points in p's neighborhood is less than M, p will be marked as a noise point.

- **OPTICS (Ordering Points To Identify the Clustering Structure):** OPTICS is an extended algorithm of DBSCAN to solve its problem on the sensitivity of parameters. Based on DBSCAN, OPTICS returns an ordered list of reachability-distance of every point in the given dataset.

- **Mean-Shift:** Mean-Shift algorithm is a density-based clustering approach based on kernel functions. It picks a random point as the initial center. It uses a kernel function to determine the weight of points in its neighborhood for re-estimation of the cluster's mean. At every iteration, the kernel is shifted to the centroid or the mean of the points within it. [10] At convergence, every visited point is considered as one cluster.

Due to the curse of dimensionality, it is hard to choose an appropriate eps for the dataset we used in this research for the DBSCAN algorithm, so it is only tested on Cosine Similarity (only good results are achieved when using cosine similarity) while OPTICS are tested on all three distance methods. These three algorithms use the implementation from the clustering library of the Python sklearn package. Due to time constraints, we didn't review other density-based clustering methods in this research.

### 4.2.4 TextRank

As introduced in section 3.3.2, TextRank is a graph-based algorithm that can rank sentences according to their importance in the article. It assumes the source text is a directed graph. Each node is a representation of a sentence and the weight is the probability of transfer between nodes. The similarity of the two sentences is initialized as the transfer probability and stored in a square matrix. The program will randomly transfer through an outbound link to the next sentence and the probability of its destination being chosen is equally

| Type | Algorithm | Distance | ROUGE 1-R | ROUGE 1-P | ROUGE 1-F1 | ROUGE 2-R | ROUGE 2-P | ROUGE 2-F1 |
|---|---|---|---|---|---|---|---|---|
| Centroid-Based Clustering | K-Medoids | Euclidean | 0.1276 | 0.2330 | 0.1547 | 0.0247 | 0.0543 | 0.0324 |
| | | Manhattan | 0.1315 | **0.2373** | 0.1589 | 0.0261 | 0.0548 | 0.0334 |
| | | Cosine | 0.3022 | 0.2235 | 0.2312 | 0.1001 | **0.0715** | 0.0757 |
| | K-Means | Euclidean | 0.1171 | 0.2280 | 0.1431 | 0.0195 | 0.0490 | 0.0268 |
| | | Manhattan | 0.1370 | **0.2450** | 0.1610 | 0.0237 | 0.0521 | 0.0307 |
| | | Cosine | 0.3451 | 0.2143 | **0.2452** | 0.1181 | 0.0707 | **0.0820** |
| | Bi-K-Means | Euclidean | 0.1372 | 0.2144 | 0.1531 | 0.0241 | 0.1531 | 0.0241 |
| | | Manhattan | 0.1823 | 0.2223 | 0.1838 | 0.0397 | 0.0545 | 0.0435 |
| | | Cosine | 0.3158 | 0.2148 | 0.2361 | 0.1076 | 0.0709 | 0.0796 |
| Hierarchical Clustering | Single Agglo-merative | Euclidean | 0.3538 | 0.1736 | 0.2206 | 0.1078 | 0.0495 | 0.0638 |
| | Complete Agglo-merative | Euclidean | 0.2450 | 0.2217 | 0.2138 | 0.0555 | 0.0569 | 0.0521 |
| | UPGMA Agglo-merative | Euclidean | 0.3244 | 0.2031 | 0.2350 | 0.0925 | 0.0609 | 0.0693 |
| Density-Based Clustering | DBSCAN | Cosine eps=0.05* | 0.1789 | 0.1946 | 0.1725 | 0.0418 | 0.0453 | 0.0406 |
| | OPTICS | Euclidean | 0.0747 | 0.2149 | 0.1032 | 0.0155 | 0.0493 | 0.0227 |
| | | Manhattan | 0.0719 | 0.2086 | 0.0995 | 0.0153 | 0.0494 | 0.0225 |
| | | Cosine | 0.1889 | 0.1946 | 0.1734 | 0.0567 | 0.0560 | 0.0520 |
| | Mean-Shift | Euclidean | **0.3677** | 0.1607 | 0.2078 | 0.1039 | 0.0452 | 0.0580 |
| Graph-Based | Text-Rank | \ | **0.3817** | 0.2360 | **0.2748** | **0.1320** | **0.0790** | **0.0937** |

Table 1. Comparative Analysis of ROUGE scores of different unsupervised ML algorithms on clinical documents

distributed among all of its outbound links. This process forms a first-order Markov chain. At the convergence of this network, a stable probability distribution table will be returned and the sentence with a higher probability of being transferred will be considered a key sentence of the source text.

In this experiment, we use the TextRank function from Python networkx package to prevent re-build the wheel.

## 4.3 Experiment setup

Any pre-trained BERT models can be used for feature extraction and we choose BioBERT-Base v1.1 (+ PubMed 1M) based on BERT-base-Cased (same vocabulary) from J. Lee's works. [6]

During the whole experiment, the number of clusters is set to 6 for all clustering algorithms and the compression rate is set as 0.05 to match the portion of reference abstracts to the source text.

To reduce runtime and efficiently use the computational power, preprocessing with feature extraction, text summarization, and evaluation are carried out in three independent processes. Any text file will only be preprocessed and perform feature extraction once before 18 different summaries are generated for them. Rouge evaluations are carried out separately on summaries produced by each algorithm.

For further information, please check our git repository.

## 4.4 Evaluation

Summary evaluation is a very difficult task. Unlike many other machine learning tasks such as image recognition or speech recognition, there is no standard answer or fix results for summaries. Different humans will likely write a different summary of the same text source based on their understandings and ways of presentation. The definition of a good summary is an

open question to a large extent. [4] The lack of a standard summary evaluation metric is one reason for the slow progress of usable automatic text summarization techniques. In this research, we mainly use automatic evaluation methods to evaluate our results and introduced human evaluation of the outcome we produced.

### 4.4.1 Automatic Evaluation

Several automatic summary evaluation metrics have been raised since the early 2000s and ROUGE is the most popular one among them. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. [5] There are four different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S.

ROUGE-N measures the n-gram overlap between the reference summary (original text) and the hypothesis summary (generated summary) where n stands for the length of n-gram. In this research, we mainly choose ROUGE-N as our automatic evaluation method for machine-generated summaries and separately compute recall, precision, and F1 scores in the context of ROUGE-1 and ROUGE-2 in our experiment.

- **ROUGE-N-R (Recall):** Recall in the context of ROUGE refers to the portion of n-gram (context) captured in the hypothesis summary of the reference summary. In ROUGE-1 where only individual words are considered, ROUGE-1-R represents the portion of overlapping words over the total number of words in the reference summary.
- **ROUGE-N-P (Precision):** Precision in the context of ROUGE refers to the portion of n-gram (context) capturing of the reference summary in the hypothesis summary. In ROUGE-1 where only individual words are considered, ROUGE-1-R represents the portion of overlapping words over the total number of words in the hypothesis summary.
- **ROUGE-N-F1 (F1 Scores):** F1 scores computes the harmonic mean of precision and recall values previously computed. ROUGE-N-F1 = 2 / (1 / ROUGE-N-R + 1 / ROUGE-N-P)

### 4.4.2 Human Evaluation

Due to the limitation of automatic evaluation methods, human evaluation is usually used to evaluate the outcome of automatic text summarizations. In this research, due to our ability limit, we are only able to provide general human feedback on our generated summaries. One medical expert is responsible for reviewing different summaries produced by different algorithms and commenting on their readability and usability. The human feedback is provided in the Results and discussed further in Conclusion.

## 5. RESULTS

The results of the ROUGE scores of 18 algorithms have been represented in Table 1. The top two scores in each column are labeled in bold. From the table, we can see that the framework's performance is the best when the TextRank method is used. In this circumstance, about 30% of 1-gram words and 10% of 20-grams phrases are covered as the original abstracts. Further discussions are listed below on different machine learning algorithms and distance methods.

### 5.1 Distance Measurement

Three distance methods are tested on three k-clustering and OPTICS algorithms. It is obvious from the table that Cosine similarity out-performed Euclidean distance and Manhattan distance. The ROUGE F1 scores of Euclidean and Manhattan are usually close and cosine similarity is clearly better in all three of them. It shows that cosine similarity can better represent the relationship between the mathematical representation of different sentences and supports a precise clustering approach.

If we dive deeper into the scores, we can see that the recall score of cosine similarity is usually two times three times of euclidean and Manhattan, but the precision score of three distance measures is usually close. This might be caused by the different lengths of summary generated by different methods. The proposed clustering approach with cosine similarity will usually select longer sentences for the summary, resulting in a longer summary than the one generated using the Euclidean and Manhattan approach and a longer summary base will result in a higher recall and a lower precision. This is why we introduce the F1 score as the main comparison proof between different approaches.

### 5.2 Clustering Algorithm

Three classes of clustering algorithms are tested in this research: centroid-based clustering, connectivity-based clustering, and density-based clustering. For clustering algorithms using different distance measurement methods, we evaluate based on the measurement method with the best score.

We can see from the table that the k-clustering and agglomerative clustering algorithms have a similar ROUGE-1 F1 score of 0.22 to 0.24 while density-based clustering lies between 0.17 to 0.20. This number refers to the number of crosswords between human summary and machine-generated summary. However, k-clustering has a ROUGE-2 F1 score of 0.075 to 0.082, better than agglomerative clustering of 0.052 to 0.070 and density-based clustering of 0.040 to 0.060. This number refers to the number of cross 2-word phrases between human summary and machine-generated summary. We can conclude from the data that k-clustering and bi-k-clustering are the most appropriate machine learning algorithms to learn the relationship of sentences and correctly identify key sentences.

It is noticeable that k-clustering and agglomerative clustering have similar ROUGE-1 F1 scores but different ROUGE-2 F2 scores. The reason might be: Though the two algorithms perform similarly in capturing keywords of the text, k-clustering can capture more key phrases compared to agglomerative clustering.

For agglomerative clustering, three methods of measuring the distance between clusters and UPGMA methods perform the best in handling the summarization task. A single link has performance in second place while a complete link performs the worst.

For density-based clustering methods, Mean-Shift performs the best. However, the eps parameter for

DBSCAN and OPTICS algorithms have a significant impact on the final result. EPS refers to the maximum distance between two members for one to be considered as a neighborhood of the other. Due to the time limit of our project, not the best eps might be set for the experiment. It is possible that DBSCAN and OPTICS algorithms can have better performance on the same task.

Our experiment has many limits. Firstly, a fixed number of clusters and compression rate is used. For different documents, a different number of clusters might have a significant impact on the final outcome. The next step of this project is to evaluate the performance of the clustering algorithm when a different number of clusters is being set. Secondly, the method of summary generation might have some limits. The sentences are ranked according to their distance from the cluster center they are assigned. However, if the cluster represents a secondary topic or a collection of noise points, the sentences might be far away from the article topic. One improvement method is that only select sentences from the largest clusters or exempt the clusters with a small number of members (a collection of noise points).

### 5.3 Human Evaluation and Comprehensive Analysis

Dr. David Cox provides a human evaluation of the machine-generated summaries. According to his feedback, the quality of summaries is overall unstable. Some summaries are exciting and some summaries are less readable compared to the original text. This outcome might be caused by different reasons. First, if there are conclusion sentences in the original document, selecting those sentences can result in a good summary quality. However, for documents with no clear conclusion sentences, extractive text summarization will perform badly. Secondly, the compression rate is set to a fixed value of 0.05 in the experiment. It might not be able to cover all key information in the document. A mathematical limit of the key information contained in a sentence should be identified to automatically determine summary length based on its content.

We also conducted a 'blind test' on different algorithms. Five samples of machine-generated summaries by different algorithms are provided to our examiner to sort in the rank of readability and summary quality. As our automatic evaluation methods indicate, the summary produced by the TextRank algorithm is placed in the first place while k-clustering, agglomerative clustering, OPTICS, and Mean-Shift and sorted afterward. This shows that the result of our automatic ROUGE evaluation is a reliable method of representing the summary quality.

The poor stability of summary quality stops applying this automatic text summarization technique in the real production environment and replaces human work. However, some potential techniques might be able to improve the quality of this system. Firstly, we can use a higher compression rate to cover more sentences like 0.3 used in [17]. After this longer summary is produced, a human summarizer can help to select sentences from this smaller scale of text and re-write sentences based on it. In this procedure, the automatic text summarization method only does half the job and a human editor will do the rest. This still saves time because it is much more efficient to summarize in a shorter text compared to the original document. This method might be the best solution for using this automatic text summarization method.

## 6. CONCLUSION

Automatic Text Summarization is a method with great potential to improve work efficiency in the biomedical domain. In this paper, we introduced the extractive text summarization framework we established and evaluated different unsupervised machine learning algorithms' performance in the framework. The proposed method will extract parameters for every sentence using pre-trained BioBERT models, and sentences will be automatically ranked using a machine learning approach.

Eighteen algorithms are selected and tested on 100 sample biomedical documents. Based on the outcome of the automatic evaluation method between human-written abstracts and machine-generated summaries, the TextRank algorithm clearly outperforms others. When the TextRank method is used, our framework is able to cover 30% of words and 10% of 2-word phrases of the human-written abstracts. Another interesting result we learned from the experiment result is that in clustering algorithms, there is a significant performance difference between different distance methods and it is shown that cosine similarity can better represent the relationship between different sentences.

It is clear that there is still room for our automatic text summarization system to improve. Our proposed method cannot match human performance on the same task and liberate the labor force. First, the limit of extractive text summarization is clear: it only used original sentences in the original text makes it less readable. Next, the original paragraph usually does not have the required information density as a summary required. Finally, sentences with different lengths may be unequally represented and shorter sentences may be ignored due to less information they contained.

Further work will be focused on these issues. We planned to involve sentence-splitting methods to perform extractive text summarization on the sub-sentence level to improve the precision of information retrieval. Second, we plan to improve our evaluation method by introducing other automatic evaluation methods and improving the human evaluation scale. Furthermore, we would like to attempt supervised machine learning using labeled data.

## REFERENCES

[1] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. Computational linguistics 28, 4 (2002), 399–408.

[2] M. Allahyari *et al.*, "Text Summarization Techniques: A Brief Survey," *arXiv:1707.02268 [cs]*, Jul. 2017, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1707.02268

[3] "Cluster analysis," *Wikipedia*. Mar. 20, 2022. Accessed: Mar. 21, 2022. [Online]. Available: https://

en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1078239941
[4] Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present, and future. In Multi-source, Multilingual Information Extraction and Summarization. Springer, 3–21.
[5] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text Summarization Branches Out*, Barcelona, Spain, Jul. 2004, pp. 74–81. Accessed: Mar. 15, 2022. [Online]. Available: https://aclanthology.org/W04-1013
[6] J. Lee *et al.*, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *arXiv:1901.08746 [cs]*, Oct. 2019, DOI: 10.1093/bioinformatics/btz682.
[7] PubMed https://pubmed.ncbi.nlm.nih.gov/about/
[8] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural Text Summarization: A Critical Evaluation," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, Nov. 2019, pp. 540–551. DOI: 10.18653/v1/D19-1051.
[9] S. Cho, K. Song, C. Li, D. Yu, H. Foroosh, and F. Liu, "Better Highlighting: Creating Sub-Sentence Summary Highlights," *arXiv:2010.10566 [cs]*, Oct. 2020, Accessed: Mar. 24, 2022. [Online]. Available: http://arxiv.org/abs/2010.10566
[10] "Mean shift," *Wikipedia*. Nov. 25, 2021. Accessed: Mar. 29, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Mean_shift&oldid=1057053424
[11] "Facts about Google and Competition." https://web.archive.org/web/20111104131332/https://www.google.com/competition/howgooglesearchworks.html (accessed Mar. 29, 2022).
[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019, Accessed: Mar. 29, 2022. [Online]. Available: http://arxiv.org/abs/1810.04805
[13] *BERT*. Google Research, 2022. Accessed: Mar. 29, 2022. [Online]. Available: https://github.com/google-research/bert
[14] M. Moradi and N. Ghadiri, "Different approaches for identifying important concepts in probabilistic biomedical text summarization," *Artificial Intelligence in Medicine*, vol. 84, pp. 101–116, Jan. 2018, doi: 10.1016/j.artmed.2017.11.004.
[15] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, p. 113679, Mar. 2021, doi: 10.1016/j.eswa.2020.113679.
[16] M. Allahyari *et al.*, "Text Summarization Techniques: A Brief Survey," *arXiv:1707.02268 [cs]*, Jul. 2017, Accessed: Jan. 03, 2022. [Online]. Available: http://arxiv.org/abs/1707.02268
[17] "Summarization of biomedical articles using domain-specific word embeddings and graph ranking | Elsevier Enhanced Reader." https://reader.elsevier.com/reader/sd/pii/S1532046420300800?token=16779B410C106368F6C609052BD7706D70DC16F795BAA517A2DBEDA6577726167452D788048069E22E377F43B9743355&originRegion=eu-west-1&originCreation=20220330004440 (accessed Mar. 30, 2022).
[18] M. Moradi and N. Ghadiri, "Different approaches for identifying important concepts in probabilistic biomedical text summarization," *Artificial Intelligence in Medicine*, vol. 84, pp. 101–116, Jan. 2018, doi: 10.1016/j.artmed.2017.11.004.
[19] M. Moradi, "CIBS: A biomedical text summarizer using topic-based sentence clustering," *Journal of Biomedical Informatics*, vol. 88, pp. 53–61, Dec. 2018, doi: 10.1016/j.jbi.2018.11.006.
[20] M. Nasr Azadani, N. Ghadiri, and E. Davoodijam, "Graph-based biomedical text summarization: An itemset mining and sentence clustering approach," *Journal of Biomedical Informatics*, vol. 84, pp. 42–58, Aug. 2018, doi: 10.1016/j.jbi.2018.06.005.
[21] O. Rouane, H. Belhadef, and M. Bouakkaz, "Combine clustering and frequent itemsets mining to enhance biomedical text summarization," *Expert Systems with Applications*, vol. 135, pp. 362–373, Nov. 2019, doi: 10.1016/j.eswa.2019.06.002.
[22] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, Feb. 2017, doi: 10.1016/j.eswa.2016.10.017.
[23] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders," *Expert Systems with Applications*, vol. 129, pp. 200–215, Sep. 2019, doi: 10.1016/j.eswa.2019.03.045.
[24] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," *Knowledge-Based Systems*, vol. 159, pp. 1–8, Nov. 2018, doi: 10.1016/j.knosys.2017.11.029.
[25] M. A. Mosa, A. S. Anwar, and A. Hamouda, "A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms," *Knowledge-Based Systems*, vol. 163, pp. 518–532, Jan. 2019, doi: 10.1016/j.knosys.2018.09.008.
[26] P. Mehta and P. Majumder, "Effective aggregation of various summarization techniques," *Information Processing & Management*, vol. 54, no. 2, pp. 145–158, Mar. 2018, doi: 10.1016/j.ipm.2017.11.002.
[27] R. Mishra *et al.*, "Text summarization in the biomedical domain: A systematic review of recent research," *Journal of Biomedical Informatics*, vol. 52, pp. 457–467, Dec. 2014, doi: 10.1016/j.jbi.2014.06.009.
[28] "Summarization from medical documents: a survey - ScienceDirect." https://www.sciencedirect.com/science/article/pii/S0933365704001320 (accessed Mar. 30, 2022).
[29] Takase, S. Okazaki, N., Positional encoding to control output sequence length, arXiv preprint arXiv:1904.07418, 2019.
[30] Liu, Y. Lapata, M. Text summarization with pretrained encoders, arXiv preprint arXiv:1908.08345, 2019.