Supervised Machine Learning for Extractive Query Based Summarisation of Biomedical Data

By

Mandeep Kaur

A thesis submitted to Macquarie University

for the degree of Master of Research

Department of Computing

29 May, 2018

MACQUARIE
University
SYDNEY·AUSTRALIA

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Mandeep Kaur
May 29, 2018

Acknowledgements

I would like to express my special appreciation and thanks to my supervisor Dr. Diego Mollá Aliod, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a researcher. I would also like to thank A/Prof. Mark Dras, for his constant encouragement and words of wisdom throughout this project. Although, I have spent 8 months in this project yet, I have learnt a lot.

I am indebted to my family and friends. In particular, I thank my parents for always making me feel special and for believing in me.

Finally, to my husband Tejinder, who helps me to believe in myself, and to my beautiful little girl, Samaahra, who reminds me to approach every day with joy and wonder. Thank you for your love and your patience especially during the writing of this thesis.

Abstract

Automation of text summarisation is a pressing need due to the plethora of textual information available online. Motivated by the success of machine learning in this domain, this research explores several supervised machine learning approaches for extracting summaries in response to queries. The first objective of this research is to compare the quality of classification and regression approaches for query-based multi-document extractive summarisation. To enable the comparison, we use a common extractive summarisation framework which attempts to identify salient sentences by scoring them based on a common set of features. Our experiments are performed on biomedical data provided by the BioASQ challenges. The second objective is to address the important issue of converting the sample summaries available in the training data into annotations that can be used to train statistical classifiers for extractive summarisation. We conduct different trials of data annotation and assess their impact in the results. On the basis of our investigations for the specific dataset used in this research, we show that the classification scheme performed better than the regression and results presented by different annotation techniques reveal that annotation with threshold 0.1 outperforms the other techniques.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Text summarisation is a task of abridgement of full text to a compact version while preserving the crucial information of the original version that is relevant to a user. The continuous surge in the volume of digital text over the internet has reached such tremendous magnitude that a plethora of online-text is available in regard to a topic, through which best quality answers can be obtained. However, manual skimming of text faces paramount obstacles like information overload, which in turn makes it difficult to search and extract information from the relevant literature [1]. For instance, medical practitioners require to analyse all the relevant information to diagnose and determine the best course of action for a particular patient. Nevertheless, it is a challenging task to explore the vast literature in order to obtain the most useful information. There are cases in which these practitioners fail to pursue answers to their queries [2]. Moreover, manually searching the information is an extremely time-consuming and expensive task. Therefore, there is a strong motivation for text processing systems that can automate some of the processes involved in this practice, capable of organising and presenting information in a productive manner.

In this research work, we address the task of automatic extractive query based multi-document summarisation of biomedical text using supervised machine learning techniques. In particular, we address the following related tasks: generating summary for a given query, comparing different techniques for summary generation and annotating data for supervised machine learning approach. In the next section, we briefly introduce some of the aspects related to automatic text summarisation. The literature associated with text summarisation is cited and elaborated in Chapter 2. Section 1.3 introduces some of the issues related to the application of supervised machine learning in text summarisation. Section 1.4 describes the research questions addressed by our research and contributions of the research are provided in section 1.5. Lastly section 1.6 outlines the structure of the thesis.

1.2 Automatic Text Summarisation

Automatic text summarisation has been a topic of interest to resolve the problems corresponding to manually iterating through the immense amount of data from the late 1950’s [3]. The process of summarisation involves taking an information source, extracting content from it, and presenting the most important content to the user in a condensed form and in a manner sensitive to the user’s or application’s needs [4]. Summarisation approaches can be divided into various categories depending on the input and functionality.

1.2.1 Single Document and Multi-Document Summarisation

Based on the number of input documents, summarisation can be classified as single or multi-document [4]. Single document summarisation is the straightforward case, where the goal of the summary is to present the significant information in the document in a concise format. Therefore, the task involves identifying meaningful information in the document and produce a summary.

On the other hand, multi document summarisation is more complex than simply searching and presenting the important text segments. While summarising multiple documents, information must be presented in the accurate order for generating precise summaries. Hence, there is an additional task of making decisions about the ordering and synthesis of information from more than one document. In addition, different documents might present similar information which could lead to redundancy in the summary. Furthermore, the information depicted by the distinct documents can be inconsistent and incoherent. All these factors make it more difficult to summarise multiple documents.

1.2.2 Extractive and Abstractive Summarisation

The basic idea behind extractive summarisation techniques is to identify significant content from the input text and generate a summary [5]. In general, summaries are generated by selecting a subset of sentences (known as extract) from the original input based on statistical and linguistic characteristics such as, keywords, sentence position and frequency etc. The generated extracts are independent textual units [6] and may result in incorrect grammar, misinterpretation and lack of cohesion and adequate information.

Alternative methods generate summaries that have better structure and are usually grammatically correct. Such summaries may be generated by organising extracted material into a deliberate form, or by generating novel text from conceptual or other forms of representation (e.g., a lexical graph). Such summaries are called abstractive summaries, and this is the type of summaries that humans generally produce. The automatic generation of abstractive summaries generally requires additional processing compared to the generation of extractive summaries.

1.2.3 Query Focused Summarisation

Generic summarisation presents key information from the original documents to help the reader, by providing the summary about the document; the significance of the information is determined only in relevance to the input content. In contrast, query focused summarisation also known as user focused summarisation, extracts and summarises the content relevant to the given query [7]. These approaches take into account a specific query and try to identify the information within the documents that is relevant to the query.

1.2.4 Supervised Machine Learning in Text Summarisation

Machine learning is a popular technique for creating predictive models based on very large datasets. The basic idea of these techniques is to learn from existing data and try prediction on new data [8]. These techniques have been widely applied to the task of text summarisation [9–11]. Machine learning can be applied to the text summarisation task if a collection of documents and their corresponding reference summaries is available [9]. Summarisation systems are expected to learn the patterns which lead to summaries and when a new data is given to system the same learned patterns are utilised to produce the summaries.

Being capable of learning features automatically, once trained on input data, machine learning systems make these techniques best suitable for understanding the complex nature of language involved in processing text. Moreover, complex machine learning such as deep learning has outperformed or significantly outperformed the traditional approaches in summarising text [12]. Another major advantage of machine learning over traditional methods is that it develops more robust systems. For example, if there is suitable training data available; then there is no need to change the rules whenever the domain changes.

This research work revolves around two supervised machine learning techniques namely regression and classification for summarisation of biomedical data. Classification approaches deal with the task of summarisation as a classification problem. These algorithms aim to classify the input sentences into summary and non-summary classes.

On the other hand, regression models used for summarisation try to fit the predicted score of a sentence as close as possible to the target score instead of labelling the sentences. Details of both the approaches are provided in Chapter 2.

1.3 Issues Involved in Implementing Supervised Machine-based Summarisation

In this subsection, we briefly review and analyse the problems and barriers faced in the implementation of supervised machine learning approaches for generating summaries. Machine learning methods are nowadays widely used in the text summarisation field, as they have numerous advantages over systems that rely on hand-coded rules. Indeed, it automates the process of presenting a condensed version by providing short and reliable summaries. However, there are several burning issues associated with the execution of these procedures for creating summaries. These obstacles are explained in detail below:

1.3.1 Feature Engineering

Feature engineering refers to getting more and meaningful information out of a data set. It can consume significant time and energy for looking over the data itself to try and identify additional information that may be 'hiding' in the features already included. Moreover, feature selection is also not easy because the features that look irrelevant in isolation may be relevant in combination [8]. A good combination of features can provide the best results. However, it is a relatively difficult task to identify the best set of features. Deep learning algorithms are acting as a solution to eliminate the issues regarding features, but they have their own pitfalls of complex training requirements.

1.3.2 Evaluation Criteria

The automated evaluation of summaries is a challenging task [1, 13] and is still an open-ended question. The absence of universally accepted standards for evaluating summaries is the primary cause for creating barriers in the evaluation of summarisation systems. In addition, the evaluation systems for summaries cannot be application independent. More precisely, the measures for evaluating generic summaries is not identical to the one used for evaluating query-oriented summaries. Furthermore, evaluation techniques can also vary according to the unit of summarisation (i.e., single vs. multi-document), domain, type (extractive vs. non-extractive) and other factors (see section 2.2).

1.3.3 Lack of Annotated Data

Supervised machine learning requires annotated training data to generate summaries. Such data is not always easily available. Although many researchers attempted to tackle this issue by manually selecting the summary-worthy sentences for their experiments [14], it consumes considerable amount of time and there is no standard agreement on selecting the sentences; as different people have different perception of an ideal summary [15] resulting in different summaries for a single dataset. Another answer to this problem is semi-supervised techniques [16]. These approaches utilise a small amount of labeled data along with large amount of unlabeled data.

1.4 Research Questions Addressed

As discussed in the previous section, summarising text by using machine learning techniques faces various issues like criteria for evaluation of summaries, feature engineering to extract the most valuable information and availability of annotated data for training supervised approaches. The solution to these problems affects the eventual output of the summarisation system. However, exploration of all these research areas is outside the scope of the research described in this thesis.

Instead, we focus on the annotation issue for training data along with assessing the performance of two different machine learning approaches; we describe these now in more detail.

1.4.1 Which supervised machine learning technique works better for query focused extractive text summarisation of biomedical data with a particularly simple feature set?

In this research, we explore extractive summarisation approaches for the task of content selection. We use simple similarity base features to assign a score to the sentences. The two approaches used for summarising text are regression and classification. As mentioned earlier in this Chapter that both these techniques use different criteria for learning and predicting the summary sentences. We performed various experiments to analyse any difference in results of both the approaches. We evaluated our approaches by using ROUGE and compared the average F1 SU4 scores of both these techniques.

1.4.2 Which annotation approach is capable of generating best results for supervised classification approach for query focused text summarisation of biomedical data?

As discussed in the previous section, annotated data plays a considerable role in producing summaries. It is hard to get this data manually annotated and is still an active research question to facilitate the supervised techniques with labeled data. We explore several annotation techniques to provide the labeled training data for the classification approach. We apply separate labelling mechanisms for annotating the data and assess the performance for each mechanism. Also, we present an annotation model based on a greedy approach proposed by Marcu [17] which automatically annotates data for a given input dataset.

1.5 Research Findings

We focus on the two questions above and our investigations lead to various research findings. These are: approaches for query based multi-document text summarisation; comparison between the results of regression and classification techniques and approaches for automatically annotating data for the task of text summarisation with their performance analysis. We now briefly present descriptions of the three contributions and a summary of corresponding evaluation results. These are described in detail in subsequent Chapters.

1.5.1 Approaches for Query Focused Multi-Document Text Summarisation

We implemented a system for providing answers to queries in form of summaries for biomedical data. As mentioned earlier, the two approaches we employed in this research are regression and classification.

1.5.2 Comparison Between Approaches

Classification and regression-based approaches have been heavily exploited for the task of text summarisation. Our work focuses on evaluating the results of both these approaches on a same data set with same features. We compared the results of regression and classification approaches in terms of final summary generation by using ROUGE. Our experimental results reveal that the classification approach performs better than the regression approach.

1.5.3 Data Annotation Techniques and their Contrast

To solve the inevitable issue of availability of labeled data for classification problem, we present different trials of labeling data to support supervised machine learning techniques for automatic text summarisation and draw a comparison of these techniques. Our investigation shows that an annotation approach with threshold 0.1 of the ROUGE score outperforms all the other annotation approaches.

1.6 Thesis Outline

Chapter 2 provides a detailed overview of relevant literature related to automatic text summarisation. The review is divided into two parts. In the first part, we provide a brief description of classical and unsupervised methods used for text summarisation. In the second part of our review, we explore supervised machine learning and deep learning automatic text summarisation techniques.

Chapter 3 discusses the data we use in this research. Chapter 4 describes our approaches and methods used in this research for generating summaries; along with our methods for labeling data for experiments using the classification technique.

Chapter 5 presents the evaluation results and discussions. Finally, in Chapter 6, we conclude with a summary of the thesis and outline future directions.

2 Literature Review

2.1 Overview

The purpose of this Chapter is to review some of the literature that is relevant to the research described in this thesis. Since the field of automatic text summarisation is too vast to be discussed in a single survey, we highlight salient techniques and breakthroughs in this discipline.

The aim of this literature review is to survey the work done in the field of extractive summarisation with an attention to supervised machine learning techniques. This survey has drawn motivation from other surveys about prior work in this field [1, 4, 7]. We begin with a general background on text summarisation, including the factors by which summarisation tasks may be classified, stages involved in producing summaries through extractive summarisation systems, and a brief history of early and unsupervised approaches. Next, we discuss several supervised machine learning and deep learning approaches. Afterwards, a comparison of various query focused summarisation systems using machine learning or deep learning is presented. We conclude this Chapter by a discussion and summary section.

2.2 Factors Related to Text Summarisation

The main aim of the summary is to present the main ideas of a document in less space [18]. The following aspects characterise the research on text summarisation [1]:

- Summaries may be produced from a single document or multiple documents,
- Summaries should preserve important information, and
- Summaries should be short.

Automatic text summarisers must consider a range of factors to achieve their goals. Here we discuss some of these factors, knowledge of which is essential to understand the process of text summarisation. The factors affecting automatic text summarisation can be grouped into three main categories: input, purpose and output [13]. Although there are a number of factors associated with each category, in this section we only mention those related to our research.

2.2.1 Input

The following factors are associated with the inputs of a summarisation system:

- **Unit** – Summarisers can either take as input a single document or multiple documents (see section 1.2.1). The data used in this research is multi-document, hence our summarisation approach falls into the category of multi-document.
- **Language-** A summarising system can be mono-lingual, multi-lingual or cross-lingual. In monolingual systems, the input and output are in the same language. Multi-lingual systems comprise more than one language. In cross-lingual systems, the input and output languages are different. Our research focuses on English text only, so it is essentially a mono-lingual system.
- **Domain-** Summarisation systems can either be domain-specific or domain-independent. Domain-specific systems are designed for specific domains and use resources and knowledge available for the specific domain (e.g. news, medical etc). Domain-independent approaches, on the other hand, can be applied to documents from various domains. This summarisation research focuses on the medical domain.
- **Medium-** Medium can be text, speech, images, tables, and so on. Although most research has been done using text, there has been some investigation into non-textual media [19].

2.2.2 Purpose:

Purpose defines what the summary is for or what the summary should resemble. Various factors related to purpose are:

- **Summary type**- A summary can be extractive or abstractive (see section 1.2.2).
- **Information-** In terms of information, summaries can be generic or user-oriented. Generic summaries consider all the information found in the input documents, while user-oriented summaries aim to extract and summarise only the information that is relevant to a user's query.

2.2.3 Output:

The factors related to the output of the summarisation system are:

- **Coverage-** Coverage of sources by a summary can either be comprehensive or selective. Comprehensive summaries summarise the whole text. In contrast, selective summaries only summarise specific information. For instance, query focused summaries only summarise a portion of the source text that is relevant to the query.
- **Medium-** The output medium in most cases is text. Although, as is the case with input media, the output can consist of other media such as images and audio.

2.3 Stages of Extractive Automatic Text Summarisation

An extractive summarisation process can be categorised into three distinct stages [20]. Figure 2.1 presents a 3-stage model for extractive summarisation.

- **Intermediate Representation:** The original text is transformed to an intermediate representation in this stage. This representation can vary for different approaches. For example, topic representation approaches convert the text to an intermediate form and interpret the topic(s) discussed in the text. On the other hand, indicator approaches take into account every sentence as a list of indicators of importance such as length of the sentence, position of sentence, sentence containing certain phrases, etc.
- **Scoring sentences:** After deriving an intermediate representation, a score is assigned to each sentence indicating the importance of the sentence.
- **Select summary sentences:** In the final stage, the top n most significant sentences are selected by the summarizer to generate a summary. Sentences can be selected in different ways. Some approaches utilise greedy algorithms to select the best sentences and other approaches may choose a collection of sentences, with a goal of maximising overall importance and minimising redundancy.

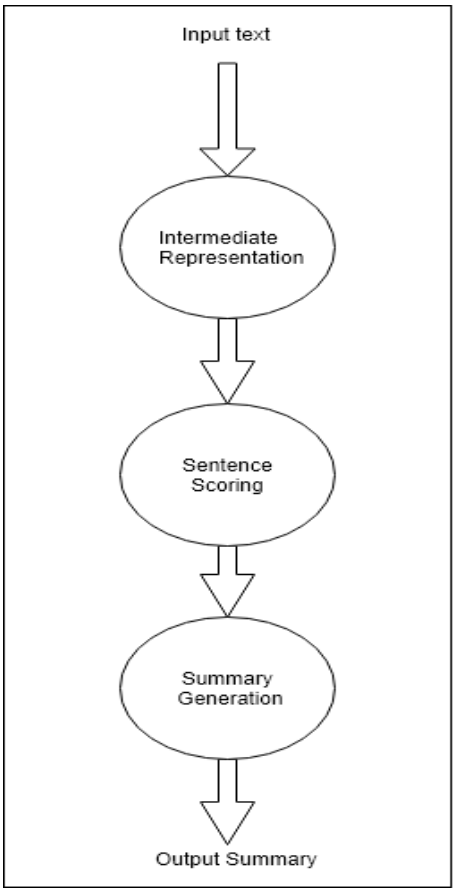

Figure 2.1: Stages in automatic extractive summarisation system.

2.4 Classic and Unsupervised Approaches

Text summarisation has a rich background of research algorithms. We first consider extractive summarisation techniques including both single and multi-document. The earliest works on text summarisation used sentence extraction as a primary component of a text summarisation system and the classic extractive approaches applied to extract summaries used statistical features for selecting significant content from the source text. The language model utilised by these approaches is only based on the frequency of words, known as bag -of-words (BOW). The statistical features have been used by both generic and query focused summarisation.

Assigning scores to the sentences is crucial in deciding the most significant sentences for the summary. Initial classic and unsupervised approaches relied on frequency features for scoring the sentences. The basic idea was that the most important information will appear more frequently in the document than less important information.

One of the earliest work on text summarisation was based on ranking words according to their frequencies and utilising those rankings to calculate the significance of a sentence. Finally, the top ranked sentences were selected as a summary [3]. Most of the prior work on extractive summarisation was based on the "Edmundson Paradigm" [21]. This work was an extension of Baxendale's method of using sentence position to identify the importance of a sentence [22]. Edmundson [21] used some other additional features to rank sentences for extraction; this method defined a framework for extraction [4].

The statistical approach for selecting sentences has been very popular in the research community and hence, a considerable amount of research associated with this approach can be seen. Some of the significant work under this roof is done by using many different features such as multiple words, noun phrases, main verbs, named entities and so on [23–26].

The features that are heavily exploited by the research in this field to discover the important content are tf-idf (see section 4.3.3) and word probability. Both of these techniques utilise frequency as a basic form of measure. Tf-idf weighting technique and its variations have been widely used for weighting a query and a document [27] .

Graph-based and cluster-based approaches have also been successfully applied to text summarisation. Graph based methods represented documents as connected graphs. [28]. These techniques outperformed various other techniques [29]. These approaches work on the assumption that sentences which are more similar to other sentences carry the most crucial ideas of the documents and are considered to be the central sentences. The most frequently used method for calculating similarity

between the sentences is cosine similarity (see section 4.3.4) with tf-idf weights for words. However, tf-idf only preserves the frequency of words and does not consider the syntactic and semantic information. Hence, the graph approaches using only tf-idf are unable to perform fine-level textual analysis [30]. More precisely, the graph-based approaches depend heavily on similarity of sentences without considering the relationship between sentences.

Cluster based approaches deal with the summarisation task according to the themes contained in the documents. The generated summaries should address various themes in the similar organisation as present in the documents. In addition to various clustering approaches [31], [32] used to identify themes, the most salient approach under this category is centroid based summarisation [33]. The centroid based models also utilised tf-idf [34] for vector representations. This approach is domain independent and easily scalable. Although cluster-based methods have been helpful in representing diversity within multiple articles, relying merely on cluster-based judgements to generate a summary does not guarantee that the summary will be meaningful.

Some of the other popular techniques applied to text summarisation are based on using lexical items with “importance-signalling” properties (such as subheadings) [35] and others utilised the discourse structure of text [36–38].

All the above-mentioned approaches aim to extract the most frequent information from the original text. However, in cases where documents contain remarkably redundant information and inconsequential content, the idea of extracting the most frequent information is not viable. It can lead to duplication of sentences in the final summary. For the purpose of summarising documents, redundancy removal is a key issue. An ideal summary should never contain repeated demonstration for the same piece of information. Hence, eliminating redundancy from the summary is a cardinal objective of both query-focused and generic summarisation.

One of the most popular approaches for selecting sentences and dealing with redundancy is maximum marginal relevance (MMR) [39]. It is a frequently used technique to ensure both minimum redundancy and maintaining relevance for ranking [39]. The approach tries to maximise the relevant sentences to the query and minimise similarity to previous selected sentences. Each time, before adding a sentence to the summary it is investigated if it is significantly similar to sentences already picked for the summary or not, in case it does, it is put aside otherwise added to the summary. Another popular approach for dealing with redundancy is based on dynamic programming [40].

2.5 Supervised Machine Learning in Text Summarisation

Machine learning is used in almost every field nowadays because it can produce models that can analyse bigger, more complex data and deliver accurate results even on a large scale. These systems can be extended easily by using more training data. Such models are robust, generalise well and behave gracefully when dealing with errors and new data [41]. The success of machine learning in natural language processing has motivated the use of machine learning procedures in text summarisation research. It has been well documented in the literature that to date, a range of machine learning methods have been developed for extractive summarization [1, 6, 42].

Based on the sentence scoring criteria these techniques can be divided into three categories: Classification, regression and learning to rank. Features used by these techniques to extract important content are generally identical to the classical approaches with some additional features such as features based on bigram n-gram model. We now describe these approaches in more detail.

2.5.1 Classification Approaches

Classification approaches deal with the task of summarisation as a classification problem. More precisely, having a set of training documents and their corresponding extractive summaries, the system categorises the sentences as either summary or non-summary. For both classes a label is assigned to each sentence (usually 1 for summary and 0 for non-summary).

The concept of summarising text by using supervised classification approaches was pioneered by Kupiec [10]. He proposed a model based on the Edmundson [21] feature set with some additional features, which are able to learn from data. Each sentence is categorised as worthy of extraction or not by a classification function, using a Naïve Bayes classifier.

In this classification approach the sentences are treated individually. However, the individual treatment of the sentences is unable to take full advantage of the relationship between the sentences. For instance, two neighbouring sentences with similar contents should not be put into a summary together, but when treated individually, this information is lost.

At first, most machine learning systems assumed feature independence and relied on Naïve Bayes methods [1] [43]. However, later models shifted the focus towards breaking the assumption that features are independent of each other [44].

Support vector machine (SVMs) are popular classification approaches used by both generic and query-focused domain [45, 46]. Methods based on SVM focus on creating a decision boundary between summary sentences and non-summary sentences. However, these methods ignore the correlation between sentences.

Hirao and Isozaki [47] used an SVM classification model to learn how to extract important sentences from manually annotated data. They reported that the SVM outperformed other machine learning models such as decision tree or boosting methods. Another noticeable work employing structural SVMs to deal with optimisation issues was done by Li in 2009 [46]. The primary goal of their research was to satisfy three constraints: diversity, coverage and balance in summary.

To address the modelling of inter-sentence dependency, document summarisation has been treated as sequence labelling task with labels indicating whether to extract a sentence for summary or not. To achieve this, hidden Markov models [48] and conditional random fields (CRF) [49] have been applied in such settings. These systems extract indicative features including sentence position, named entities, similarity or distance to query, content word frequency, etc [42].

Hidden markov models and conditional random fields can be seen as outperforming other learning models [20, 48, 49] due to explicitly modelling the dependency between sentences. However, CRFs can model more complex dependencies than HMM, and can use multiple features. On the other hand, HMM usually uses words as features. Hence, CRF based methods [49], provide an answer to the problem of sentence dependency, eventually producing better outcomes in comparison to other techniques. Nevertheless, being more expressive and powerful, CRFs are slower to train.

2.5.2 Regression Approaches

The classification-based methods described above perform binary classification of whether a sentence should be included in a summary or not. To assign a label to a sentence a threshold value is required. A variation to this approach is regression. Regression models used for summarisation try to fit the predicted score of a sentence as close as possible to the target score instead of labelling the sentences.

An earlier work similar to this research is by Ouyang et al. [11]. In their research, they applied regression models to query-focused multi-document summarization, used Support Vector Regression (SVR) to estimate the importance of a sentence in a document set and compared classification, regression and learning to rank models. In order to train models, they constructed training data semi-automatically by assigning each sentence a “nearly true” importance score using several N-grams methods and with a reference to human summaries. More details regarding this system are provided in Chapter 5.

Support vector regression (SVR) has also been used in conjunction with other techniques like integer linear programming (ILP) for generating summaries [50] and has achieved state-of-the-art results in comparison to other competitive extractive summarisers. The extractive summariser used in this technique has also been utilised by one of the recent work done in biomedical domain [51] . They generated a query -focused summary for BioASQ (see Chapter 3) data.

A system named FastSum [52] used regression SVM for training their data set by using the least expensive NLP techniques to generate the summary. The system used a set of clusters as input data and simple pre-processing was performed on the sentences. A comparison of this system with MEAD [33] showed that it is more than 4 times faster than MEAD.

2.5.3 Learning to Rank Approaches

The use of ranking models has also gained popularity in automatic text summarization. It is an active research area in the machine learning community. Several ranking algorithms have been proposed including ordinal regression [53], perceptron [54], RankNet [55], RankBoost [56] and ranking SVM [57].

Learning to rank refers to supervised machine learning techniques for training the model in a ranking task. The regression techniques try to map to a real value. Instead, learning to rank transforms the task into a simple problem of ranking spans from an original text. As such, learning to rank does not care much about the exact score that each unit gets, but cares more about the relative ordering among all the units.

Given sentences with labeled importance scores, it is straightforward to get learning to rank models to train a model capable of assigning high rank to the most important sentences. Ranking SVMs are the most commonly used approaches for ranking the sentences. Ranking SVMs are a generalization of the classical SVM formulation that learns over pairwise preferences, rather than binary labeled data [58].

The motivation behind ranking SVMs is that, for ranking problems, it is inappropriate to learn a classification model, since it does not take the structure of the problem into account [59]. In addition, it is evident from the literature that while comparing SVMs and ranking SVMs to model the relevance of sentences to queries, the results show that ranking SVMs outperform standard SVMs on a small test collection [60]. Motivated by this outcome, other researchers have also evaluated the efficiency of ranking SVMs in selecting relevant sentences against support vector regression (SVR), and gradient boosted decision trees (GBDTs) [59].

Learning to rank has also been applied to the summarisation of XML documents with a goal of learning how to best combine the sentence features such that within each document, summary sentences get higher scores than non-summary ones [61].

Another significant work done in this category is also by using ranking SVM to combine features for extractive query-focus multi-document summarization [57]. In order to do that, a graph-based method was proposed for training data generation by utilizing the sentence relationships and a cost sensitive loss was introduced to improve the robustness of learning. The method outperformed the baseline strategies.

Learning to rank techniques have not been implemented in this research due to time constraints. However, we consider employing them to the summarisation task in future.

2.6 Deep Learning in Text Summarisation

The efficiency of machine learning algorithms heavily depends on the representation of the data. Data representation or feature extraction refers to transferring the raw input data into a representation which can be efficiently used by machine learning tasks. However, it is a difficult task because not only does it need lots of human time and effort, but also the features are highly application-dependent. Moreover, in most tasks, it is hard to find the right features.

Deep learning is a branch of machine learning which solves this primary problem of data representation by replacing hand-crafted features with efficient algorithms for learning multiple levels of representation automatically [62]. The main idea behind the working of deep learning techniques is to understand the world by making a hierarchy of concepts in which each concept is defined in regard to simpler concepts. This process is called hierarchical learning and inspired by human problem-solving behaviours.

In addition, machine learning techniques used in various tasks are based on shallow models such as SVM trained on very high dimensional and sparse features. Whereas, deep neural networks provide a very robust framework for processing natural language by capturing the recursive nature of language and have been producing superior results on various NLP tasks. In this section we review some of the deep learning models used for generating extractive summaries.

Deep neural networks have presented remarkable results in the automation of text summarisation [12, 63]. It is well documented in the literature that these systems have been widely used for this task [64-66]. The major steps involved in summarising text using neural based summarisers are: transforming words to vectors, known as word embeddings; encoding sentences/documents to continuous vectors using the word embeddings and finally feeding the sentence/document representations to a model for

summary generation [64]. The representation is a crucial part of extractive summarization systems [67].

Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are commonly used in neural-based summarisers. CNNs or RNNs can be used as encoders for extracting sentence/document features. In the case of extractive summarisation neural based models can be used as regressors for ranking/selection and for abstractive systems these models act as decoders for producing abstracts. As the scope of survey is extractive systems, we only discuss extractive systems using deep architectures in this section.

CNNs application is evident in many of the summarisation systems [68-70]. Each system using CNNS have different criteria for sentence representation and training data with a common purpose of transforming a sequence of word embeddings to vectors. Cheng and Lapata [63] proposed an attentional encoder-decoder for extractive single-document summarization. Another related work addresses the problem of query-focused multi-document summarization using CNNs [70], where weighted-sum pooling was used over sentence representations to represent documents.

Also, recurrent neural networks (RNNs) have been successfully used in NLP and this type of deep neural networks has also been used for text summarization [63]. The system proposed by Cao [69] formulates the sentence ranking as a hierarchical regression task. It measures the importance of a sentence and its constituents (e.g., phrases), simultaneously. The model used hand-crafted words features as input and employs supervised learning.

RNNs using gated recurrent unit (GRU) and long short-term memory networks (LSTMs) have gained popularity in recent years and have been applied to many natural language processing tasks including text summarisation [12]. Systems using GRU have reported to be outperforming other comparable systems.

Most of the recent work in the field of summarisation has utilised abstractive techniques to generate summaries. The focus of this research work is on extractive techniques, hence providing details about the abstractive technologies would be beyond the scope of this research work. Technologies applied to produce abstractive summaries are mostly deep learning technologies such as sequence to sequence networks and encoder-decoder etc.

2.7 Comparison of Query Focused Text Summarisation Systems

This section presents a table comparing several query focused text summarisation systems. We have tried to incorporate a system from each machine learning category discussed above (i.e. sequence labelling, classification, regression and learning to rank) and two deep learning systems. The

comparison is illustrated in table 2.1. Details about various data sets mentioned in the table are available in section 3.1. The methods of evaluation such as ROUGE and Pyramid are also described in section 3.4.1 and 3.5.1 of next Chapter.

System	Input Unit	Training Data	Learning Technique	Domain	Approach	Evaluation Metric
D.Shen, 2007[49]	Single Document	DUC	Classification (Sequence Labelling)	Domain Independent	Conditional Random field	Precision, Recall and F1+ ROUGE
M.Fuentes, 2007 [45]	Multi Document	Pyramid Data	Classification	Domain Independent	Support Vector Machines	ROUGE and AutoPan[1]
C. Wang, 2007 [60]	Multi Document	TREC	Learning to Rank	Web Pages	Ranking SVM	Intrinsic evaluation (Averaged F1)
F.Schilder, 2008 [52]	Multi Document	DUC	Regression	Domain Independent	Support Vector Regression	ROUGE
C.Shen, 2011 [57]	Multi Document	Training data generated from human summary	Learning to Rank	Domain Independent	Ranking SVM	ROUGE
Y.Ouyang, 2011 [11]	Multi Document	DUC	Regression	Domain Independent	SVR, SVM and Learning to Rank	ROUGE
Y. Liu, 2012 [66]	Multi Document	DUC	Deep Learning	Domain Independent	Restricted Boltzmann Machines and Dynamic Programming	ROUGE
Z. Cao, 2016 [70]	Multi Document	DUC	Deep Learning	Domain Independent	Convolutional Neural Networks	ROUGE

Table 2.1: Comparison of various Query -focused summarisation Systems.

[1] AutoPan is a procedure for automatically matching fragments of text summaries to SCUs in pyramids [71].

2.8 Discussion

Most earlier approaches focused on single-document extractive summarization and employed the statistical based techniques discussed above. They used simple features to generate summaries. However, these methods tend to produce redundant summaries.

Although machine learning approaches provide better results than the earlier approaches, they are computationally complex and there is a lack of semantic analysis of source text. In addition to this, supervised learning approaches rely on labeled training data which is not always easily available. Getting this data manually labeled is time consuming and automatic annotation of data is still an active research question.

Undoubtedly, deep learning is a very powerful framework and it has made remarkable improvement in various fields including automatic summarisation. These techniques can capture important characteristics of natural language. This potentiality can tackle the major issue of discrete representation by using word embeddings [72] . Nevertheless, deep learning has some shortcomings such as requirement of substantial amounts of training data. Secondly, training these models is quite challenging, time consuming and computationally expensive. Furthermore, tuning deep neural networks is a complex task because there are multiple choices to make on the number of layers and hyperparameters.

Recalling from Chapter 1 the basis of this research is to draw a comparison between techniques and to explore the labeling issue. Our intention is to utilise simpler features and less complex training models; as mentioned above there are several challenges related to implementation of deep learning algorithms. Therefore, we use machine learning for our experiments in this project with a motive to implement deep architectures for future research in this field.

2.9 Summary

Our exploration of literature found that there are a variety of approaches used for generating summaries. From our study it is evident that machine learning systems are presenting better results in contrast to earlier systems because they have the ability to provide increasing levels of automation, replacing much-time consuming human activity with automatic techniques that improve accuracy and efficiency by discovering and exploiting regularities in training data. However, they come with certain limitations.

Applications of deep neural networks are also explored, which seems to be a promising approach for the future. However, owing to limitations posed by training challenges these techniques are complex to implement.

3 Summarisation Corpus and Evaluation

3.1 Overview

In previous Chapters, we discussed the task of summarisation and presented a review of literature relevant to this task. This Chapter will provide details on the data set that is the basis for all the experiments in this research along with a brief introduction of the evaluation metric utilised to evaluate the efficiency of the system.

It is quite important to have a trustable corpus for automatic text summarisation. A wide variety of data sets are available for general and domain specific applications. Among these, Document Understanding Conference[2] (DUC), Text Analysis Conference[3] (TAC), Text Retrieval Conference[4] (TREC), TREC Genomics Track, Question Answering for Machine Reading Evaluation (QA4MRE)[5], Question Answering over linked data (QALD)[6] are the most popular ones.

In our research, we utilise a corpus associated with medical domain. In the remainder of this Chapter we provide a detailed description of the corpus. In Section 3.2, we provide explanation about the data including examples from the corpus; next sections discuss evaluation methods for text summarisation systems and the evaluation metric used in our research; and lastly Section 3.7 concludes the Chapter.

3.2 Data

In our research, we utilised a biomedical corpus provided by BioASQ[7] to develop our summarisation approaches. BioASQ is a semantic indexing, question answering (QA) and information extraction challenge which constructs benchmark datasets, evaluation services, and organizes international biomedical QA competitions since 2013 [73]. In particular, data related to BioASQ task 5b is used for the experiments in our research. In the BioASQ data set each question contains the text of the question, the question type, a list of source documents, and a list of corresponding snippets from the source documents. For illustration, the data used in this research consists of 1306 biomedical questions. All questions are accompanied by gold answers and are also annotated with relevant documents, snippets, concepts and triples, containing the information required to compose their answers.

[2] http://duc.nist.gov/
[3] http://www.nist.gov/tac/
[4] http://trec.nist.gov/data.html
[5] http://celct.fbk.eu/QA4MRE/
[6] http://qald.sebastianwalter.org/index.php?x=home&q=1
[7] http://bioasq.org/

BioASQ data has several categories of questions. Following the BioASQ terminology questions can be of type yes/no, factoid, list and summary. A paragraph summarizing the most important information from relevant documents and structured data in response to an input query is called an "ideal_answer". The focus of this research is to identify the "ideal" answer for a given query. Table 3.1 below illustrates the examples of each question type with their relevant "ideal_answers".

Yes/No:	Is the protein Papilin secreted?
Ideal answer:	Yes, papilin is a secreted protein.
Factoid:	Which thyroid hormone transporter is implicated in thyroid hormone resistance syndrome?
Ideal answer:	Hemizygous MCT8 mutations causes TH resistance syndrome in males characterized by severe psychomotor retardation, known as the Allan-Herndon-Dudley syndrome (AHDS).
List:	List the human genes encoding for the dishevelled proteins?
Ideal answer:	DVL-1, DVL-2, DVL-3
Summary:	What is the role of anhedonia in coronary disease patients?
Ideal answer:	Anhedonia is associated with poor prognosis in patients with coronary disease. Namely, in patients with coronary disease, anhedonia was associated with increased mortality, greater risk for major cardiac event, impaired physical health status, more cardiac symptoms, more feelings of disability. These associations were independent from clinical and demographic factors.

Table 3.1: Examples of questions with their "ideal_answers" in BioASQ 5b.

The examples in the figure above show that "ideal_answers" do not just contain the exact answer but they also include answer justifications useful to the medical practitioners and can be understood as query-based summaries. For this reason, we tackle the task of finding the "ideal_answer" by using query-based summarisation techniques instead of question answering techniques.

The data set can be seen as a collection of question units and each unit comprises a "body" which is a question itself, "snippets", "ideal_answers" and properties. The properties include ID, a description of concept, type and corresponding information. There can be several possible related snippets in regard to a particular question. Instances of "question", "ideal_answer" and "snippets" are depicted in figures 3.1, 3.2 and 3.3 respectively. In our research we have utilised question ID, "ideal_answer" and snippets to fulfil the objective of generating a summary.

```
"id": "55031181e9bde69634000014",
      "body": "Is Hirschsprung disease a mendelian or a multifactorial
disorder?",
      "type": "summary"
```


Figure 3.1: Instance of a BioASQ “question”


```
 "ideal_answer": [
       "The 7 known EGFR ligands are: epidermal growth factor (EGF),
betacellulin (BTC), epiregulin (EPR), heparin-binding EGF (HB-EGF),
transforming growth factor-α [TGF-α], amphiregulin (AREG) and epigen
(EPG)."
```


Figure 3.2: Instance of BioASQ “ideal_answer”


```
"snippets": [
        {
          "offsetInBeginSection": 1085,
          "endSection": "abstract",
          "document": "http://www.ncbi.nlm.nih.gov/pubmed/24323361",
          "offsetInEndSection": 1199,
          "text": "the epidermal growth factor receptor (EGFR) ligands,
such as epidermal growth factor (EGF) and amphiregulin (AREG)",
          "beginSection": "abstract"
        },
        {
          "offsetInBeginSection": 1139,
          "endSection": "abstract",
          "document": "http://www.ncbi.nlm.nih.gov/pubmed/24124521",
          "offsetInEndSection": 1247,
          "text": " EGFR ligands epidermal growth factor (EGF),
amphiregulin (AREG) and transforming growth factor alpha (TGFα)",
          "beginSection": "abstract"
        },
```


Figure 3.3: Instance of BioASQ “snippets”

3.3 Evaluation Metric

Evaluating a summary is a complicated task because there does not exist any ideal summary for a given document or set of documents. In the case of query focused summarisation, the evaluation metric should be able to determine whether the extracted information correctly answers the user’s query. There are several ways to evaluate the summaries including manual and automatic approaches. The following sections illustrate both categories for evaluating summaries.

3.4 Manual Evaluation

In manual evaluations, domain experts analyse and grade summaries, based on a scale. Humans can infer, interpret and utilise real world knowledge to scrutinize the text with similar meaning but worded differently; hence there is more confidence in such evaluations. However, the process of manual evaluation is quite expensive. In addition to this, these evaluations are biased and have shown to be inconsistent and unstable as well [74].

3.4.1 Pyramid Method

One of the popular manual evaluation methods is proposed by Nenkova and Passonneau [71] known as pyramid method. Pyramid methods tries to score summaries based on semantic matchings of the content units. It works under the assumption that there's no single best summary and therefore multiple reference summaries are necessary for this system.

Instead of attempting to elicit reliable judgement from humans, this evaluation method is calibrated to human summarisation behaviour. Summary content is categorised into summarisation content units (SCU), and SCUs representing the same semantic information are annotated in each source document. Once annotation is complete, each SCU is assigned a weight equal to the number of summaries in which the SCU appears.

Next, the SCUs are partitioned into a pyramid in which each tier contains SCUs of the same weight and higher tiers contain SCUs of higher score. Therefore, an optimal summary is expected to contain SCUs from the top tier followed by (if length permits) SCUs from the next tier and so on. Finally, the score assigned to an automatically generated summary is the ratio of the sum of the weights of its SCUs to the sum of the weights of an optimal summary with the same number of SCUs.

3.5 Automatic Evaluation

Automatic system evaluation is an alternative way of evaluating summaries [75], which is still an open research topic. The most popular automatic evaluation metric is Recall-Oriented Understudy for Gisting Evaluation (ROUGE). The following sections discuss ROUGE and gold standards evaluation systems in more detail.

3.5.1 ROUGE

ROUGE is a software package that has become very much the standard for automatic evaluation [76]. The intent of ROUGE is to find the similarity between automatically generated summaries and reference summaries also known as gold standard summaries; in order to do that it counts the

overlapping units such as unigram and bigram between the reference and automatically generated summaries.

The ROUGE package yields parameters such as precision, recall and F-score. Where the precision refers to the fraction of extracted items which are relevant, and recall provides the fraction of relevant items extracted by the model. F-score can be referred as a weighted harmonic average of precision and recall.

$$\text{precision} = \frac{|\{\text{relevant senteces}\} \cap \{\text{extracted sentences}\}|}{|\{\text{extracted sentences}\}|} \tag{3.1}$$

$$\text{recall} = \frac{|\{\text{relevant senteces}\} \cap \{\text{extracted sentences}\}|}{|\{\text{relevant sentences}\}|} \tag{3.2}$$

$$\text{F-Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3.3}$$

There are several ROUGE metrics with the same purpose and different approaches. ROUGE-N, ROUGE-L and ROUGE-SU are commonly used in the summarization literatures.

- ROUGE-N determines the percentage of n-gram overlapping of system generated summary and reference summary. It requires the consecutive matches of words in n-grams (n needs to be defined and fixed) that is often not the best assumption.
- ROUGE-L seeks to find the longest common subsequence (LCS) between two summaries, with the logic that summaries with longer LCSs are more similar. It considers the sentence-level word orders and automatically identifies the longest in-sequence word overlapping without a pre-defined n.
- ROUGE-SU measures the percentage of skip-bigrams and unigrams overlapping. Skip bigram consists of two words from the sentence with arbitrary gaps in their sentence order. Applying skip-bigrams without any constraint on the distance between the words usually produce spurious bigram matchings [76]. So, ROUGE-SU is usually used with a limited maximum skip distance, such as ROUGE-SU4 with maximum skip distance of 4.

The various ROUGE metrics have been shown to have good correlation with human-produced rankings of summarisers. Since the introduction of ROUGE, its popularity has seen its widespread use in evaluating automatic summarisation systems across various domains including the medical domain.

These are one of the most frequently used approaches for the automatic evaluation of summaries. Although these approaches are very robust and efficient, the main issue related to these techniques is

that if two summaries are generated using non-overlapping vocabulary, yet conveying the same information content, the similarity score assigned by n-grams based metrics would be poor and, consequently unrepresentative of the actual information they have in common.

3.6 Gold Standards

Gold standards also known as human reference summaries are often used for evaluating automatic summarisation primarily because humans are able to capture significant source content and produce well-formed output text [6]. The expected output summaries are manually created by human experts and are supposed to contain the necessary content. For evaluating the system its generated summaries are compared with the gold reference summaries which can be done manually or automatically. The more similar a generated summary is to the gold standard, the better it is considered to be.

3.7 Summary

If summarising is a hard task, evaluating summaries is even harder. The evaluation should be able to determine if the extracted information correctly answers the user's query. In this Chapter we discussed manual evaluation and automatic evaluation and mentioned some of the issues related with each approach. We also briefly described the concept of gold standards with respect to automatic evaluation. An overview of the corpus used in our research with some of the instances from the original data set is also presented in this Chapter.

4 Research Approaches

4.1 Overview

The purpose of this Chapter is to describe our proposed methodologies and algorithms used in this research for extracting sentences, labelling data and generating query specific summaries. We first introduce the feature extraction phase and then we present an explanation of training methods used for sentence scoring. Next, we provide details about various annotation schemes used for annotating data for classification.

As described in Chapter one, the intent of this research is to particularly provide answers to two research questions. We try to answer the first question by implementing two different supervised machine learning approaches. In the end, we compare the effectiveness of both approaches to determine which approach worked better for generating query-based summary for BioASQ data.

To answer the second research question, we conduct different trials of data annotation and assess their impact on the results for classification method. We investigate the efficiency of each scheme applied by using average F1 ROUGE SU4 scores.

In next section we discuss the summarisation model used in this research in detail including feature set and learning models. An elaboration about the annotation issue and approaches is provided in section 4.5.

4.2 Summarisation Model

Our research explores two separate methodologies for generating extractive summaries. For this purpose, we propose a three-stage summarisation model as discussed in Chapter 2 (figure 2.1). In the initial stage, input text is transformed to an intermediate representation (section 4.3); in the second stage each sentence in the input is assigned an importance score or label depending on the approach applied (section 4.4) and in the final stage we select the n most significant sentences to generate a summary.

For all our experiments we focus on implementing different approaches to assign scores or labels to sentences in the second stage; leaving the first and the final stage constant. We model the task of extracting information from multiple documents, based on the information needs of a query, as a query-focused, multi-document, extractive text summarisation task. In form of input and output, we formulate the task as follows:

Input-1: A query

Input-2: A source abstract

Output: An extractive summary of input-2 based on information needs of input -1

In Chapter 3, we presented samples of these queries and target summaries called as “ideal_answers”. Our corpus contains a collection of human-authored summaries associated with each question. Figure 4.1 provides an example of a question and relevant summary (reference summary) from our corpus. This indicates that any summarisation algorithm attempting to automatically extract query-focused content from documents must be capable of providing appropriate coverage of content in regard to the query, instead of just focusing on extracting the outcome.

Question.

What is known about the effect of acupuncture in smoking cessation?

Summarised Answer.

Ear acupressure (EAP) and ear acupuncture have been used for smoking cessation, and some positive results have been reported. Auricular (ear) acupressure has been purported to be beneficial in achieving smoking cessation in some studies, while in others has been deemed insignificant. The combined acupuncture-education group showing the greatest effect from treatment.

Figure 4.1: A question and a reference summary based on the information needs of the query.

Details about the approaches we apply for the summarising task are provided in next sections. Figure 4.2 shows a high-level view of our summarisation model. The first step in the figure represents input to the system. Whereas, the following three steps corresponds to the stages of the extractive summarisation system.

The general summarisation algorithm used in our research attempts to select n sentences from the source documents by performing the following steps:

1. Divide data into training and testing set

2. For each sentence and query in the training set perform feature extraction;

4. Apply regression or classification approach;

5. Select n highest scoring sentences; and

6. Present selected n sentences as the summary.

In our summarisation approach, we use a target summary length of n sentences. In all of our experiments the value of n is 3; we kept it fixed to provide a common criterion for comparison among different approaches.

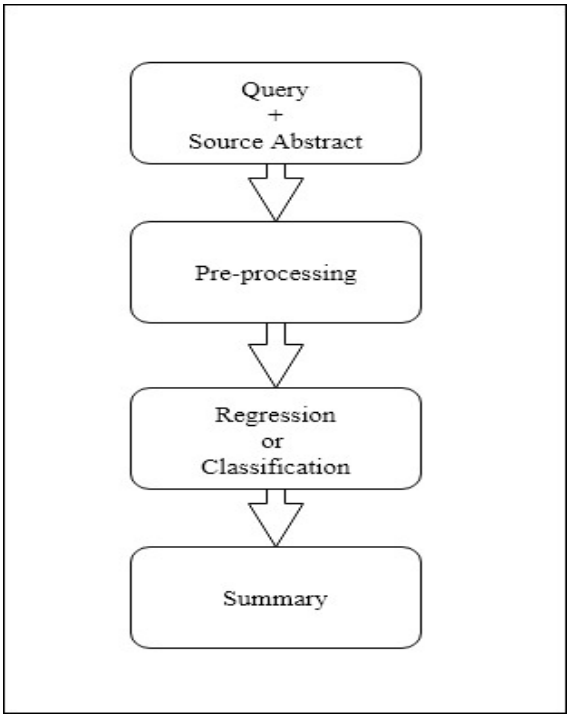

Figure 4.2: The overall summarisation model.

4.3 Pre-processing

We commence our work by dividing our corpus into two subsets: training and testing. A 10-fold cross validation is used to partition the data into training and testing set. Details about K-Fold Cross validation are provided below.

4.3.1 K- Fold Cross Validation

Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two sets namely, training set and testing set. The training set is used to learn or train a model and the testing set is used to test the model [77]. In k-fold cross-validation, the dataset is randomly partitioned into k equal size sub-samples. From these k sub-samples, a single subsample is retained as the test data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the test data. Eventually, the results from the k- folds are averaged to produce a single final estimation.

In data mining and machine learning 10-fold cross-validation (k = 10) is the most common [77]. In our experiments the number of folds is also 10. The advantage of this method is that all observations

are used for both training and testing, and each observation is used for testing exactly once. Moreover, this method of partitioning data tends to provide a less biased estimation of accuracy.

After partitioning the data, we perform vectorisation of sentences and questions also known as feature extraction. It is detailed in the following section.

4.3.2 Feature extraction

In Chapter 2 we discussed the extractive techniques used for summarisation. We reviewed the simple Edmundson approach [21] for extractive summarisation, where multiple scores are assigned to a text segment, and the final score of the segment is the weighted sum of the individual scores. Research on extractive summarisation has largely followed this paradigm. Despite its simplicity, the Edmundson paradigm is extremely flexible. As it not focuses on the features used themselves, rather the combination of features utilised to perform the experimentation. This motivates this research to utilise simple features for sentence level scoring. In addition, the features used in this research have been commonly used for text summarisation.

In addition to this, a fundamental goal of our research is to compare the efficacy of two techniques in terms of producing precise summaries. The intent of this research is not about doing feature analysis to identify the best features for generating summaries, instead it focuses on drawing a contrast between two approaches for query focused summarisation. Therefore, the features used are simple.

4.3.3 Term Frequency (tf) -Inverse Document Frequency (idf)

Primarily, we compute the term frequency (tf) and the inverse document frequency (idf) over all the sentences in the training data. tf-idf is the most commonly used representation in information retrieval and text summarization systems [27]. Although tf-idf is a relatively old weighing scheme, yet its simplicity and effectiveness has made it popular among researchers. tf-idf represents each word in the document using its term frequency tf in the document, as well as over all documents (idf):

$$\text{Tfidf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t \tag{4.1}$$

Where idf of a term t in document d is $\log \frac{N}{dft}$ (N is the total number of documents in a dataset).

In the context of this research, the tf-idf representations are constructed for each sentence as well as each query in the data.

4.3.4 Sentence Similarity to Query

Since our intent is to generate a query-focused summary, we attempt to incorporate a technique that rewards sentences similar to the associated queries. We perform this through the cosine similarity

measure. The cosine similarity is the cosine of the angle between two vectors. The cosine similarity metric represents the two sets as vectors of word occurrence features. Treating a text unit as a vector of word features is known as a vector-space approach. For most similarity metrics, including the cosine metric, a score closer to the maximum (i.e., 1) indicates that there is a high overlap between reference terms and sentence words. The cosine metric is calculated using the following equation:

$$\text{similarity} = \cos(\theta) = \frac{u.v}{|u||v|} \quad (4.2)$$

where u and v are two vectors, and the similarity is given by dividing the dot products of the vectors by the product of their magnitudes.

In our proposed approach, we compute the similarity of each candidate (s_i) sentence with the associated query (q_i), using the above-mentioned vector representations for each. The score assigned is equal to the cosine similarity of the vectors.

$$SIM_i = \text{CosSim}(s_i, q_i) \quad (4.3)$$

Where CosSim () is the cosine similarity function defined above.

To score candidate sentences for the final summary we used we use different learning approaches. Details of these techniques are provided in the following sections. We have used the same features for both training methods.

4.4 Approaches for Extracting Summaries

Moving onto our first research question. We apply regression and classification-based techniques for generating a summary for a given query. In the next section we describe our regression approach for generating summaries relevant to a provided query. The classification approach is discussed in section 4.4.2.

4.4.1 The Regression Approach

As described in Chapter 2 regression-based classifiers output a continuous importance score for a sentence. The application of regression for summarising text [50] is evident for various datasets including BioASQ [51]. The principal goal of these models is to estimate the score of a sentence based on the given feature set. In our case the feature set is as explained in section 4.3:

1. tf.idf vector of the candidate sentence.
2. Cosine similarity between the tf.idf vector of the question and the tf.idf vector of the candidate sentence.

We performed all the regression-based experiments with the use of Support Vector Regression (SVR). In our research, the target scores used to train the SVR system are the F1 ROUGE-SU4 score of each individual candidate sentence.

Supervised approaches rely on labeled training data. A typical way to construct labeled data for training is to set ROUGE, the most commonly used automatic evaluation metric, or its variants or approximations as prediction target for sentence scoring. This treatment is intuitive and has empirically justified by experiments [78].

A motivation to applying SVR for text summarization is that it can estimate the importance of sentences by providing continuous values. In particular, the regression type of SVM can rank sentences with continuous values instead of putting sentences into classes. In our case, as the target scores for each sentence is F1 ROUGE SU4 and applying SVR to predict the score of each sentence can prove useful.

For each question Q_i in the data we compute the F1 ROUGE SU4 [76] between each sentence S_i and the gold standard summary which is the "ideal_answer" of Q_i and provided as a score to each sentence S_i. More precisely, ROUGE-SU4 also considers skip bigrams with a maximum distance of 4 words between the words of each skip bigram. This measure has been found to correlate well with human judgements in extractive summarisation [76] and, hence, as mentioned earlier, training SVR to predict the rouge score of each sentence can be particularly useful. Therefore, a sentence with a high rouge score has a high overlap with the gold standard summaries; and since the sentences in the gold standard summaries are those considered significant by human authors, a sentence with a high rouge score is most likely to be important [50]. Furthermore, it is a metric used with BioASQ data and has been seen in related research experiments [51].

4.4.2 The Classification Approach

We now describe the classification approach used in our research and explain how we use it for generating summaries relevant to a provided query.

Classification is a supervised machine learning approach which deals with the summarization task as a two-class classification problem, where a sentence is labeled as "summary" if it belongs to the extractive reference summary, or as "non-summary" otherwise. We provide the "summary" class a value '1' and the "non-summary" class a value '0'. In the testing mode, the summariser labels each sentence with a value '0' and '1'. The trainable summariser is expected to learn the patterns which lead to the summaries, by identifying relevant feature values which are most correlated with the classes "summary" or "non-summary".

Classification approaches have been widely used for summarising text in both generic and query focused summarisation [45, 46] and have proved to be successful in domain specific summarisation, where classifiers can be trained to identify specific types of information such as sentences describing literature background in scientific article summarisation [7].

The success of these approaches in various domains provides a motivation to implement this technique for generating summaries for biomedical data and comparing them against our implemented regression approach. We provide the same features to the classification summariser as discussed in the previous sections, and we trained the system by using Support Vector Machines (SVMs). SVMs perform well in many classification problems.

In this project we trained our classification systems by annotating data using several techniques. All these annotation techniques are discussed in the next section.

In order to implement both regression and classification algorithms sklearn has been used. Tf-idf vectorizer form sklearn has been used for vectorisation of input text. Also, we have used cross validation form sklearn to divide data into 10 folds (with shuffling) and as mentioned earlier we have used SVR and SVM classifier from sklearn package for training and testing data. Default parameters available in sklearn for both SVR and SVM have been used for training and testing purposes.

4.5 Data Annotation for Classification Approach

In this section we provide a description about annotating data for our classification-based experiments. We start with a background of corpora annotation and moving on towards detailing several schemes used to annotate data in this research. We find that text annotation is still a developing field and it is a challenging task to provide labeled data for supporting supervised techniques for summarising text.

In computational linguistics, annotated corpora are used to train machine learning algorithms and evaluate the performance of automatic summarisation methods. The annotation or labels usually indicates the significance of each sentence in the case of summary generation.

The employment of annotated corpora to the field of summarisation dates back to late 1960s. Since then a lot of research has utilised the labeled data for their experiments including manual [10] as well as automatic procedures [11].

The decision as to whether a sentence is important enough to be annotated can be taken either by humans or by programs. Recalling from Chapter 2 the importance of sentences can be determined manually. However, asking annotators to select summary-worthy sentences is a time-consuming task, more critically, there is no standard agreement on choosing the best sentences to be included in the

summary. To provide a solution to this problem, many researchers have implemented alternative methods such as automatic alignment of human abstracts and the input [79, 80] in order to provide labeled data of summary and non-summary sentences for machine learning.

However, this approach is not suitable and has certain shortcomings. For illustration, different people may choose different content for their abstracts and consequently summary-worthy information may not be identifiable based on a single abstract. To overcome this issue, some researchers have proposed to leverage the information from manual evaluation of content selection in summarisation [81]. In this case the multiple sentences can be marked as expressing the same fact that should be in the summary [81]. Alternatively, similarity between the sentences in the human abstracts and those in the input order can be computed to identify the similar sentences without doing the full alignment [82].

Semi-supervised approaches have also been applied as a solution to this problem. In semi-supervised learning unlabeled data is utilised in the training process. There is usually a small amount of labeled data along with a large amount of unlabeled data [16].

In our research the focus is on generating automatically annotated corpora to train the system. We have used several approaches to accomplish this task. Next sections illustrate the approaches used in this research for automatically labeling data starting with the ROUGE approaches and culminating with a greedy approach proposed by Marcu [17].

4.5.1 ROUGE Annotation

We have mentioned previously in this Chapter that several systems have used ROUGE for annotating data and its application has been proved useful. So, in the classification approach we also assigned labels to the sentences based on F1 ROUGE SU4 between each sentence and the target summary.

We try several thresholds to define the labels for both summary and the non-summary classes. More precisely, we arbitrarily choose a threshold which is a specific SU4 score and then assign a label 1 to the sentences with SU4 score higher than the threshold and a label 0 to the sentences with SU4 score lesser than the same pre-defined threshold. This is done for every sentence associated with a query.

Firstly, we experiment by labelling the three highest SU4 scoring sentences as summary (i.e. label 1) class for each query in the data. Secondly, we try a threshold of 0.1. We label the sentences as “1” if its SU4 score is higher than 0.1 and label the rest as “0”.

We also experiment using various other thresholds, but the higher results are produced by using the above-mentioned thresholds, so we only discuss about only these three thresholds in this research work.

Lastly, we apply a greedy approach by Marcu for automatically labeling sentences. This approach is elaborated in the next section. The results associated with each category are provided in Chapter 5.

4.5.2 Marcu Annotation

Annotating data is a crucial phase in supervised summarising. Despite of the annotation approaches discussed above, our intent is to find an additional approach which can improve the results furthermore. In addition, the focus of this research is on investigating the annotation task for automatic summarisation. So, we experiment with one more approach to see any difference in results.

The motivation behind selecting this approach for our experiments is that it takes into account the similarity between the abstract and the input text to generate a list of significant sentences for summary [83]. This approach is discussed in detail below.

Marcu [17] proposed a greedy method to overcome the obstacle of lack of annotation data. The aim of this method is to automate the demanding task of manually identifying the extract (units in a text) used to write the abstract. The underlying idea is to build <Abstract, Extract, Text> tuples from <Abstract, Text> tuples with an assumption that an Extract corresponds to the subset of clauses in the Text whose semantic similarity with the Abstract is maximal.

This method, instead of selecting sentences which are identical to those in the abstract, eliminates sentences which do not appear to be similar to ones in the abstract. The basis of the methodology is that, a sentence from a document does not relate to any sentence or part of sentence from the abstract if the similarity between the document and its abstract does not decrease when the sentence is removed from the document [17]. This elimination process continues while similarity does not decrease. It terminates when the similarity starts decreasing or there is nothing left to be compared against.

Our annotation approach is motivated from Marcu's method, so we call it Marcu annotation. However, we have not implemented the full algorithm proposed by Marcu. The full description of Marcu approach is provided in his research paper [17]. The whole Marcu algorithm for creating an extract is divided into two parts: generating core extract and clean-up core extract.

The first part of the algorithms results in an extract through which important sentences in the text can be identified and annotated. In the second part, Marcu performed some cosmetic procedures to the generated extract which he calls clean-up procedure. In this step Marcu employed some heuristics to further reduce the set of sentences.

We only implement the first part of the algorithm. There are two reasons for not implementing the second part of the algorithm. Firstly, some of the heuristics require knowledge of the rhetorical

structure of the source to be able to apply them. This information was not available and could not be easily obtained. In addition, for some of the heuristics, the details were insufficient to know exactly how to implement them. Figure 4.6 shows the Marcu algorithm implemented in this research.

The input to the algorithm is a list of queries, abstract and text related to a given query. In step 1, pre-processing is performed on queries, abstracts and text. Pre-processing involves tokenising all these lists into sentences and then performing stemming and removing stop words. We use NLTK for this pre-processing task in contrast to Marcu, who used a shallow clause boundary and discourse marker identification (CB-DM-I) [84] algorithm for this task. This algorithm is more complex and considers the information related to various textual units to perform pre-processing.

Marcu Algorithm

Input: Abstract (A)= The reference summary

Query (Q) = A question associated with each abstract

Text (T)= Text associated with each query

Output: Extract (E) = A set of sentences from text which has maximum similarity to abstract

1. Break abstract, query and text into sentences
2. Perform stemming and delete stop words from abstract, text and query
3. E_m= Text
4. S = $\text{argmax}_{S' \in Em}$ Sim (E\S', A)
5. **for** each query Q **do**

    while (Sim (E_m, A) < Sim (E_m \ S, A)) **do**

        $E_{m=} E_m \setminus S$;

        S = $\text{argmax}_{S' \in Em}$ Sim (E\S', A)

    end while

end

Figure 4.3: Marcu extraction algorithm (generation of core-extract)

Afterwards, we do the vectorisation of all three input lists (i.e. query, abstract and extract). We use tf-idf for this purpose. Initially, we assume the extract to be whole text (step 3 in the figure 4.6). Then for each query in the data we repeat step 5.

Step 5 can be explained as follows: If we delete from E_m a sentence S that is totally distinct from the abstract A we obtain a new extract $E_m \backslash S$ whose similarity with A is higher than that of E_m. We apply a greedy approach and repeatedly delete sentences from E_m so that at each step the resulting extract has maximal similarity with the abstract, we eventually converge to a state where we can no longer delete sentences without decreasing the similarity of E_m with the abstract. The E_m at this stage is considered as the extract that we are looking for.

There are some cases where data has multiple target summaries for a question. In such cases, we concatenated all these summaries into a single target summary.

The similarity between the extract and the abstract is computed by using the cosine similarity function described in section 4.3.4 of this Chapter.

The results generated by using this approach are provided in Chapter 5. We consider implementing the second part of this algorithm (i.e. clean-up process) in our future experiments.

4.6 Discussion

In the sections above, we focused our attention on automatic text summarisation and attempted to perform query-focused, extractive text summarisation. The data used in the research comprise of multiple documents. Therefore, the summarisation task addressed in this research can be seen as a multi-document summarisation task. It must be mentioned that our objective is not to produce complete multi-document summaries, but to explore the effectiveness of the two approaches for the task.

As mentioned earlier, multi-document summarisation approaches suffer from the problems of incoherence and redundancy, and a number of approaches have been proposed in the literature to address these problems. In context of this research, we are interested in assessing the methods to score the sentences. Therefore, we do not implement any approach to deal with the issue of redundancy in the summary.

Another issue need to be mentioned in this section revolves around the data. The data set used for the experiments is not balanced. Most of the annotated SU4 scores in the training data have very low values. Therefore, the regression system tries to minimise the errors in the low values of the training data at the expense of errors in the high values. Inevitably, this highly imbalanced data results in prediction errors for the sentences with high SU4 scores that are the most significant sentences for the final summary.

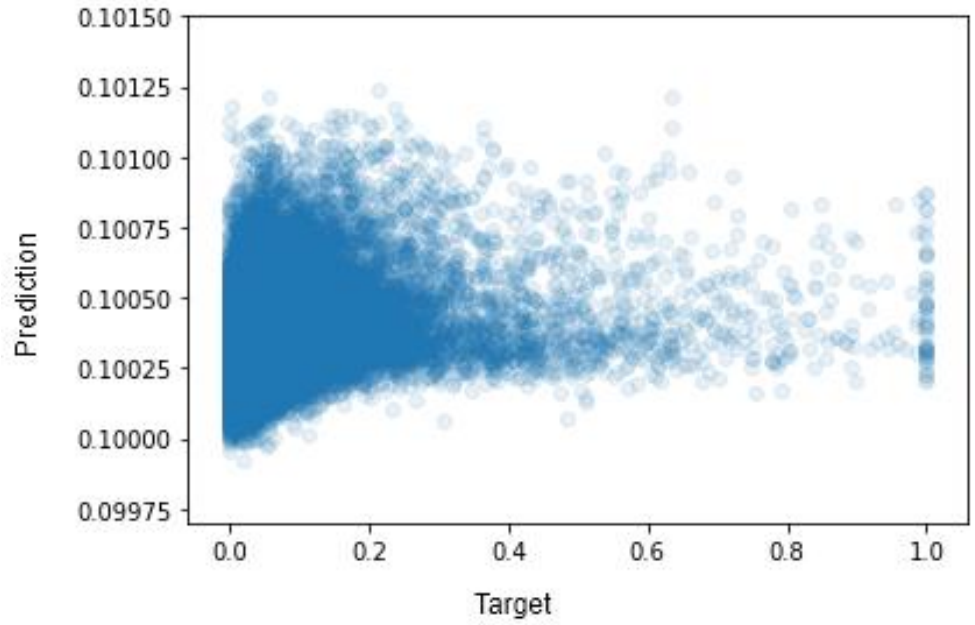

Figure 4.4: Plot of imbalanced data (Strength of opaqueness shows the concentration of data points)

Figure 4.4 depicts the imbalanced data after applying regression on test data. It plots the target against the predicted SU4 in the regression experiments. A huge cluster can be seen on around the low values in contrast to the high values. The system seems to be learning until target 0.2 but sightly before 0.4 system it seems unable to discriminate among SU4 scores. For our research we are concerned about the sentences with the highest SU4 scores as they are the sentences that can be a part of the final summary.

Note, incidentally, a vertical line of scores can be seen at target 1 these are the sentences with highest SU4 scores. A possible cause of this vertical line can be, during the annotation process annotator may have copied the same sentences form abstracts hence, producing identical copies which can be seen at target 1. Next sections elaborate the process of sampling data for both cases.

4.6.1 Sampling Data for Regression

We try to balance the data by providing more samples with high values to the SVR. In order to generate higher valued samples, we applied binning and sampling to the data set. In particular, we divided the data into 10 bins. Each bin contains 1000 samples (with replacement) from data and hence, contain more samples with high values. Figure 4.5 shows the process of sampling and binning for the data. The first bin contains the samples whose SU4 score is between 0.0 and 0.1, the second bin consist of samples between 0.1 and 0.2 and so on.

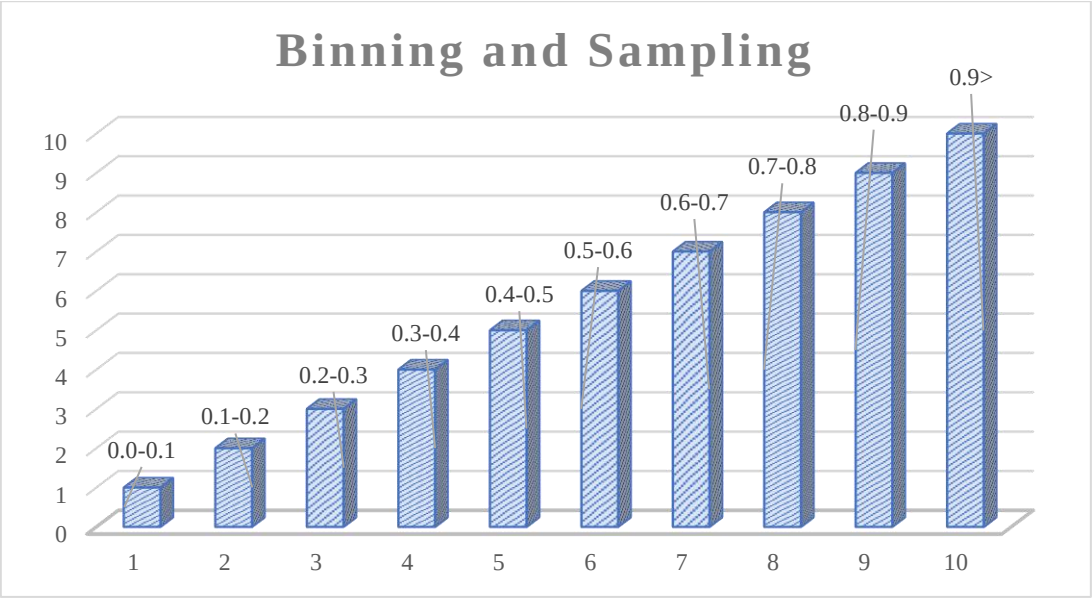

Figure 4.5: Sampling and binning of data for regression approach

We performed sampling and binning to the data before the application of SVR. However, sampling and binning data did not create any noticeable difference in balancing the data. No significant improvement was seen in the learning process. As shown in figure 4.5 the learning behaviour of the system is similar as before at the high end of target ROUGE, which is the part that matters most. Also, almost all the prediction values are between 0.492 and 0.505.

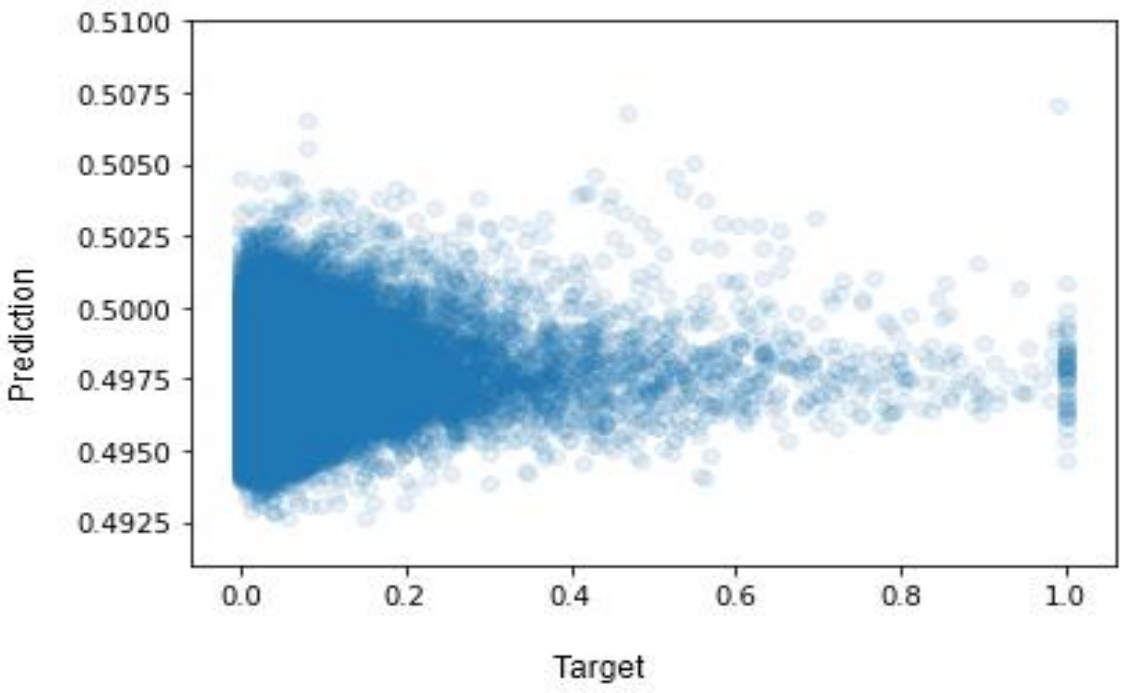

Figure 4.6: Plot of data after sampling and binning.

4.6.2 Sampling Data for Classification

In the case of classification, we also tried balancing data with an assumption of providing more positive labels to the classifier. In contrast with regression where the data is partitioned into 10 bins, here we have only two bins one containing positive samples and the other containing negative samples. Identical to the regression sampling each bin consists of 1000 samples of associated category (either 0 or 1).

While performing experiments on the sampled data using the classification approach, the system did not seem to be learning and generated almost similar probabilities for both classes for all the annotation techniques discussed in section 4.5.

4.7 Summary

In this Chapter, we provided details of the methodologies used for experiments in this research. We explained the extractive summarisation model used for generating summaries. Details of features and learning methods used for experiments in our research is also provided. We also mentioned that simple features were used to facilitate answering the research questions of the project. Additionally, as we are dealing with a burning issue of lack of annotation data to implement supervised machine learning. We have also tried to explain our approaches for annotation. An alternative sampling process performed on the data was also elaborated in this Chapter, which did not improve the final results.

5 Experimental Results and Discussions

5.1 Overview

This Chapter covers the experimental results and the methodology used to evaluate the framework described in previous Chapter. The purpose of the experiments presented in this Chapter is to provide answers to the questions set out in Chapter 1.

To satisfy the goal of the first research question, two different techniques i.e. regression and classification have been implemented to generate summaries for a given query. We apply both these techniques on a common data set associated with the medical domain. The pre-processing procedure is also same for both approaches.

To provide an answer to the second research question, several experiments are conducted using the classification approach with differently annotated data. We use ROUGE and Marcu approaches for assigning labels to the data.

In next section we provide details about the evaluation system used in this research. In section 5.3 and 5.4 results of regression and classification are demonstrated. Section 5.5 compares both machine learning techniques followed by section 5.6 which provides a contrast among various annotation approaches. Results produced after sampling the data are discussed in section 5.7. A discussion on results is presented in section 5.8.

5.2 Evaluation System

The Evaluation of the generated summary is a challenging task in text summarization. There are a range of open questions associated with it such as is there any automatic metric to evaluate the similarity between generated summary and human generated one? If not is there any manual option available? or even is there any need to have a gold standard human summary? [85]. One alternative is the pyramid metric as an indirect manual evaluation [71]. However, due to lack of resources we applied a fully automatic metric.

The ROUGE evaluation package has been commonly used as a fully automatic evaluation metric [76]. This is a set of evaluation methods that automatically determine the quality of a system summary by comparing it to human-generated summaries.

We evaluate all our approaches automatically using the ROUGE evaluation tool. Our system generated summaries are all evaluated by comparing with the associated gold standard summaries which are the “ideal_answers” in our case.

ROUGE-SU measures the percentage of skip-bigrams and unigrams overlapping. We evaluated our summarisation approaches by comparing their ROUGE-SU4 -scores. This measure has been found to correlate well with human judgements in extractive summarisation [76]. In next sections we discuss the results of regression, followed by the results of classification and then we provide a comparison between both approaches.

5.3 Regression Results

For our experiments using regression, we experiment with the use of Support Vector Regression (SVR). The regression set up and features are described in Chapter 4 (see section 4.3).

The target scores used to train the SVR system are the F1 ROUGE-SU4 score of each individual candidate sentence. We use the question and the source abstracts as input, and the "ideal_answers" from the questions as the target summaries. For each query a ROUGE SU4 score of each sentence is computed in relevance to a target summary. The same score is used as basis for all the regression related experiments.

Recalling from Chapter 4 we conduct all the experiments by using 10-fold cross-validation.

5.4 Classification Results

For evaluation of the classification technique, the features used for identifying the most salient sentences are same as that of regression. We use Support Vector Machines (SVMs) for training and testing data. Various annotations approaches used in this research are described in section 4.5 of Chapter 4.

Firstly, we perform all the experiments by choosing the three highest SU4 scoring sentences from the data to be labeled as summary class. (i.e. labeled as 1) and the rest of the sentences as non-summary (labeled as 0).

The second approach we experiment with utilises a ROUGE SU4 threshold of 0.1. As mentioned in Chapter 4 we also perform experiments using other threshold values, but the higher results are obtained using above mentioned threshold values. So, in this Chapter only results related to above mentioned thresholds are provided.

The final approach we use in our experiments is the Marcu approach as already discussed in Chapter 4.

In next section we provide a comparison between the regression and classification techniques. A comparison among all the annotation approaches is provided in section 5.6.

5.5 Regression Versus Classification

In this section we draw a comparison of both techniques on the basis of average F1 ROUGE SU4 score for 10 folds of cross-validation. To produce comparable results, we chose the same settings for both experiments as we mentioned in Chapter 4. We keep pre-processing, feature extraction and number of sentences (3 sentences) in the final summary constant for both approaches. Same data is used for training and testing in both techniques.

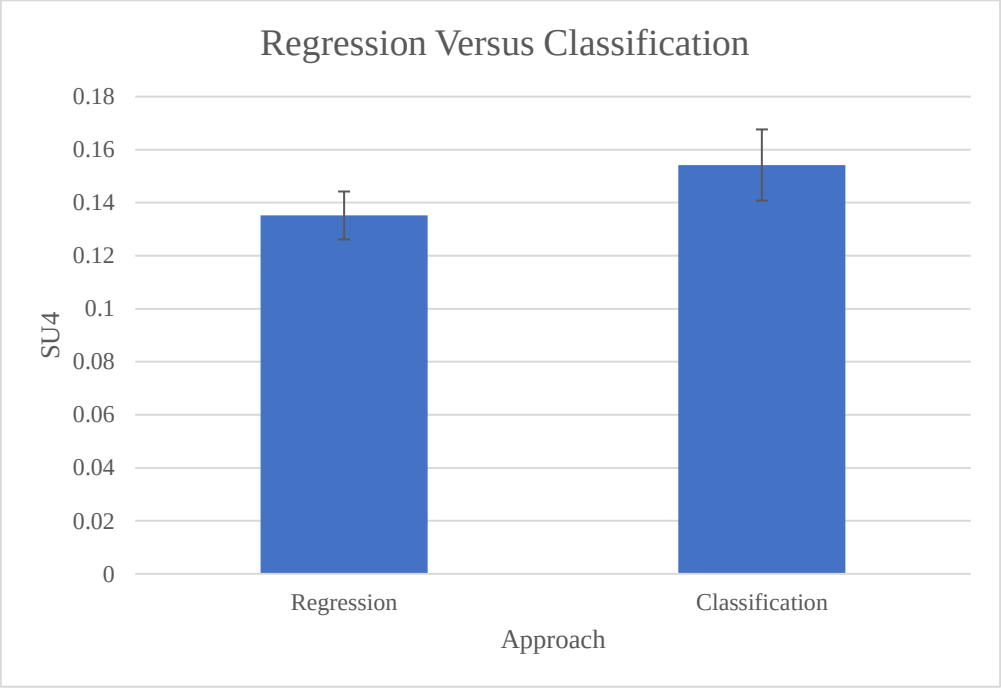

Figure 5.1: Comparison of classification (with 0.1 threshold) and regression on basis of ROUGE SU4 (error bars refer to standard deviation) in 10-fold cross-validation results.

Figure 5.1 compares F1 ROUGE SU4 scores of both approaches. The average SU4 score of classification is higher than in regression. The classification approach mentioned in figure above is one with threshold 0.1.

5.6 Comparing Annotation Approaches

In supervised learning techniques, labeled data play an important role and the outcome of the system heavily depends on the approach used for annotating data. In the last section we elaborated the results of each labeling technique used in our experiments. Now we provide a comparison among them.

Figure 5.2 shows F1 ROUGE SU4 scores of all the approaches including the approach using three sentences with highest SU4 as summary class, with threshold 0.1 and Marcu respectively.

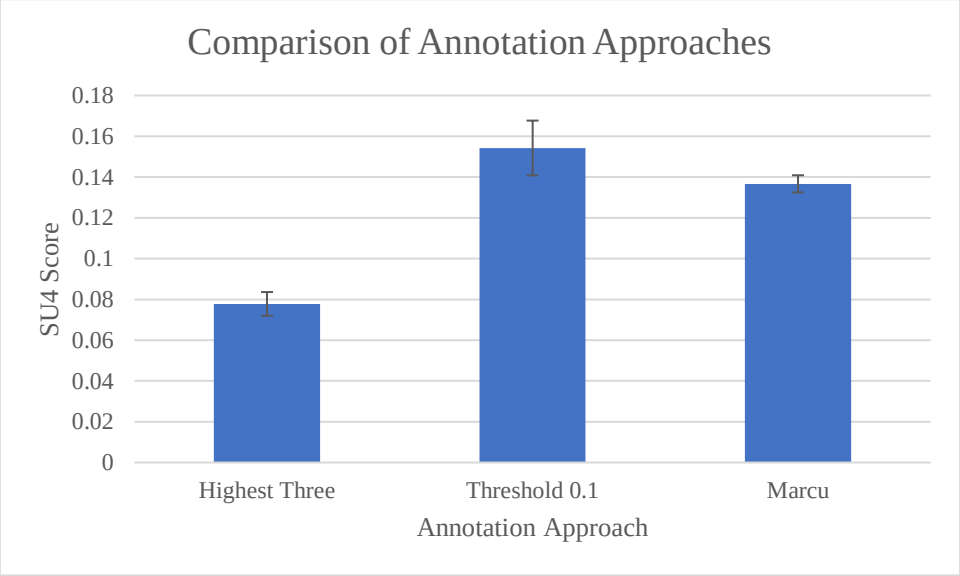

Figure 5.2: Comparison of various annotation approaches (error bars refers to standard deviation)

The second approach (i.e. with threshold 0.1) can be seen as outperforming all the other approaches. However, the first approach produces the lower SU4 score among all the three. Whereas, Marcu is better than highest three but not competing the second one. Standard deviations for all the approaches are also presented as error bars in figure 5.2.

5.7 Evaluation of Sampled Data

As we discussed in the previous Chapter we applied sampling and binning to balance the data. We trained and evaluated the sampled data in the same way as we did before sampling.

In the case of regression, after sampling a decrease in SU4 score is evident. Figure 5.3 shows the results of regression after sampling data.

Our intent is to improve the results further by generating higher valued samples through binning and sampling. Considering the plot 4.5 in fourth Chapter, the system does not seem to be learning for the sentences with higher SU4 scores, these are the sentences which have high chances of being part of the final summary for a given query. As they are the sentences which are considered to be containing the most salient content. However, no improvement has been found by sampling data.

In the case of classification, we categorised the data into two bins and performed sampling. For each of the annotation approaches discussed above system is generated almost similar probabilities for both classes. More precisely, the system has not learnt how to distinguish between both classes. Hence, the results produced are not reliable.

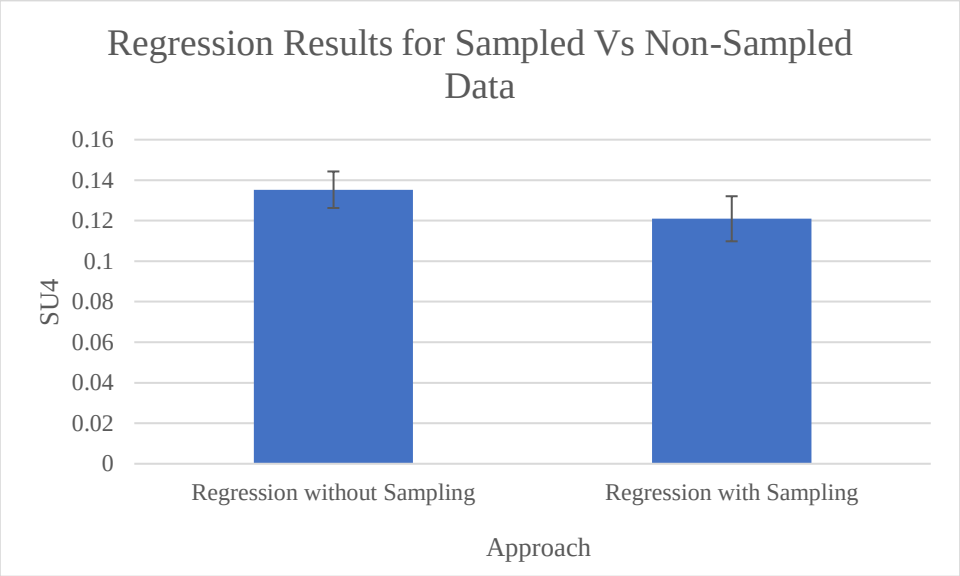

Figure 5.3: Comparison of Regression approach with Sampled and Non-Sampled Data (error bars refers to standard deviation)

5.8 Discussion

In our experiments classification works better than regression for query based extractive summarisation of biomedical data. However, a similar work performed by Ouyang et al. [11] reported regression better than classification in their experiments. They evaluated their systems on DUC data with different features and annotation approaches in contrast to our work.

The annotation utilised in their classification approach for assigning labels to sentences used two thresholds. They positively annotated the sentences with ROUGE score higher than 0.7 and negatively annotated those with score lesser than 0.3.

We replicated their approach in our experiments to see any differences in the results by assigning labels to sentences in a similar fashion (i.e. 1 for summary class and 0 for non-summary class).

We use the same settings for this experiment, which we used for all our earlier experiments including same data, features and number of sentences for the final summary. The metric used for evaluating is ROUGE as we have done for all the other experiments.

Our experiment shows results compatible to those reported by Ouyang et al. The classification approach in this scenario produce lower results than our regression approach. However, their proposed annotation approach uses two thresholds and selects sentences higher than 0.7 and lower

than 0.3; ignoring the sentences with score 0.7 and 0.3. This can be a possible reason for the difference in results compared to our classification approach.

Figure 5.4 provides a comparison of our best performing annotation with Ouyang et al. approach by showing variation of SU4 over all cross-validation folds. By referring to the figure, it can be seen that our annotation approach performs better in contrast to the Ouyang et al. approach.

Our experiment with this annotation supports the results provided by Ouyang et al. By comparing this classification technique to our regression approach, we can say that it is not performing better than regression. Hence, regression is a preferred technique for this scenario.

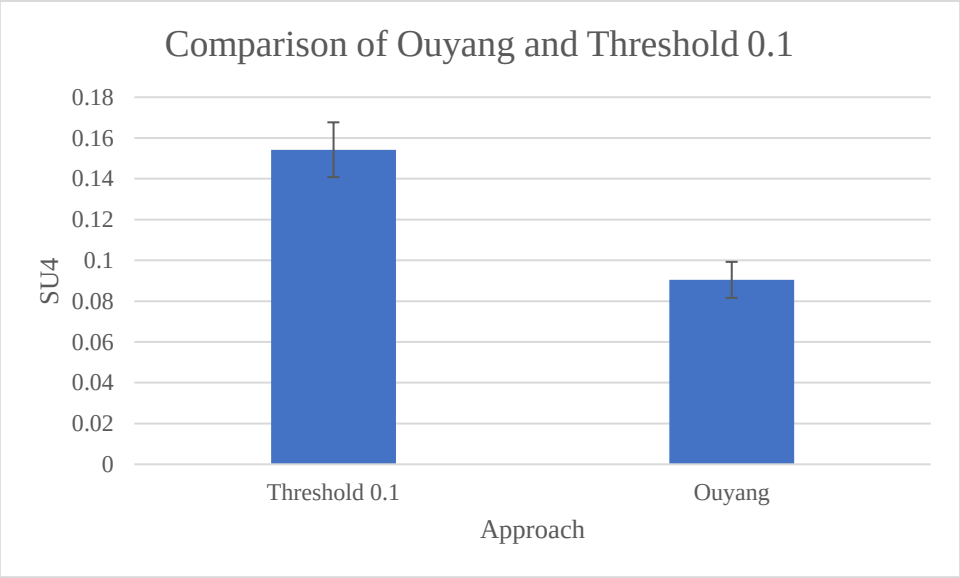

Figure 5.4: Classification with Ouyang et al. and our annotation approach (0.1 as threshold).

Moving on to our annotation approaches used in this research. Marcu approach presented better results than the worst performing classification approach but cannot compete the highest performing approach. Marcu selects the summary class sentences by comparing the cosine similarity between the abstract and the input list of sentences.

The metric used for identifying the similarity between abstract and the input has an impact on the outcome of the system. We consider evaluating our system by using the Marcu approach with ROUGE as a similarity measure instead of cosine similarity to see any differences in results for future research.

The focus of the research is to provide a comparison between the two techniques so, we do not experiment and compare our results with any baseline results. Although we have performed experiments using other threshold values in case of classification and taking three random sentences as summary sentences in case of regression to see the difference in results, but we present only those results which are highest among the others.

Another point needs to be mentioned here is that there are other researches which performed summarisation experiments using BioASQ data used in this research. The results reported in this research work are not directly comparable with these results for the reason that, the system implemented in this research uses the entire source summaries as input. In contrast, the other systems used additional information about what snippets from the source summaries are most relevant.

6 Conclusion and Future Work

6.1 Concluding Remarks

In this thesis, we presented a comparison of two supervised machine learning techniques for extractive summarization. In specific, we used regression and classification. Our system is query-based, intended to summarize multiple documents, satisfying a request for information expressed by a user's query.

We also explore the demanding phase of annotating data for supervised summarisation; drawing a comparison among several annotation approaches.

To evaluate the model for both approaches, we conducted an automatic evaluation and compared the performance of our system with human generated systems by using ROUGE. A series of experiments have been conducted by sampling data and labeling data by different mechanisms for classification-based approaches.

Our experiments revealed that classification performs better than the regression when a threshold of 0.1 SU4 is applied for annotating data.

While comparing the different annotation approaches, there is considerable difference between the results when using threshold 0.1 and highest three SU4 scoring sentences.

Additionally, we experimented by annotating three random sentences as summary sentences to see the results. The results in this case were very poor in comparison to the other annotation approaches. Hence, it can be said that the annotation approaches discussed above improves the performance of the system and the system seems to work with the change in annotation scheme.

Sampling and binning of data is also performed to balance the unbalanced data. As the data contained a huge number of sentences with low SU4 scores in contrast to sentence with high SU4. However, no noticeable change is reported after performing the sampling and binning procedure for both regression and classification. Furthermore, the system is not even seemed to be learning when trained on sampled data for classification.

Although the data used in the experiments is biomedical data, yet we have not utilised any terminology from the data in our experiments. So, the results produced by our system may be compatible with other data sets using same features and learning techniques.

6.2 Future Work

In our work, our extractive system does not deal with the issue of redundant sentence in the summary. For future work, we will try to incorporate redundancy removal in the system.

Further experiments, related to error analysis for difference between regression and classification are to be done to identify the uncertainty present in current results and implementing solutions to decrease that for more precise evaluation. It is beyond the scope of this research. So, we aim to perform this in future to analyse the difference, which will be helpful in determining the best alternatives to improve the efficacy of the current system.

Our annotation approach motivated by Marcu's approach is implemented partially in this research. We consider implementing the other half of the approach to see any improvement in results. In addition, we are interested in evaluating our system by using Marcu annotation approach utilising ROUGE as a similarity measure.

Another focus of future work will be to conduct experiments to analyse if the current results are compatible while summarising single documents. Furthermore, we are interested in comparison of results by using more and different combination of features along with application of other techniques.

We intend to extend the capabilities of our summarisation model by the application of more complex regression and classification models. This goal can be aligned by applying deep learning for summary generation as these techniques have achieved remarkable results in many natural processing (NLP) tasks.

In future we are keen to run experiments for generating summaries for other domains and evaluating results. In addition, we are interested in summary generation by using abstractive approaches. These approaches are nowadays very popular for generating summaries and have presented state-of-the-art results in text summarisation. We aim to reimplement a state-of-the-art system for generating summaries for medical domain.

In addition, there is another objective of doing analysis of experiments by using learning to rank approaches. This type of learning algorithm may help improve the performance.

References

[1] D. Das and A. F. Martins, "A survey on automatic text summarization," *Lit. Surv. Lang. Stat. II Course CMU*, vol. 4, pp. 192–195, 2007.

[2] J. W. Ely, J. A. Osheroff, M. L. Chambliss, M. H. Ebell, and M. E. Rosenbaum, "Answering physicians' clinical questions: obstacles and potential solutions," *J. Am. Med. Inform. Assoc.*, vol. 12, no. 2, pp. 217–224, 2005.

[3] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, 1958.

[4] I. Mani, *Automatic summarization*, vol. 3. John Benjamins Publishing, 2001.

[5] C.-Y. Lin, "Automated Text Summarization," in *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts*, 2005.

[6] K. S. Jones, "Automatic summarising: The state of the art," *Inf. Process. Manag.*, vol. 43, no. 6, pp. 1449–1481, 2007.

[7] A. Nenkova and K. McKeown, "Automatic summarization," *Found. Trends® Inf. Retr.*, vol. 5, no. 2–3, pp. 103–233, 2011.

[8] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[9] J. L. Neto, A. A. Freitas, and C. A. Kaestner, "Automatic text summarization using a machine learning approach," in *Brazilian Symposium on Artificial Intelligence*, 2002, pp. 205–215.

[10] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 1995, pp. 68–73.

[11] Y. Ouyang, W. Li, S. Li, and Q. Lu, "Applying regression models to query-focused multi-document summarization," *Inf. Process. Manag.*, vol. 47, no. 2, pp. 227–237, 2011.

[12] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents.," in *AAAI*, 2017, pp. 3075–3081.

[13] K. S. Jones, "Automatic summarizing: factors and directions," *Adv. Autom. Text Summ.*, pp. 1–12, 1999.

[14] J. Ulrich, G. Murray, and G. Carenini, "A publicly available annotated corpus for supervised email summarization," in *Proc. of aaai email-2008 workshop, chicago, usa*, 2008.

[15] G. J. Rath, A. Resnick, and T. R. Savage, "The formation of abstracts by the selection of sentences. Part I. Sentence selection by men and machines," *J. Assoc. Inf. Sci. Technol.*, vol. 12, no. 2, pp. 139–141, 1961.

[16] K.-F. Wong, M. Wu, and W. Li, "Extractive summarization using supervised and semi-supervised learning," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 2008, pp. 985–992.

[17] D. Marcu, “The automatic construction of large-scale corpora for summarization research,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 137–144.

[18] D. R. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization,” *Comput. Linguist.*, vol. 28, no. 4, pp. 399–408, 2002.

[19] M. T. Maybury, “Generating summaries from event data,” *Inf. Process. Manag.*, vol. 31, no. 5, pp. 735–751, 1995.

[20] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in *Mining text data*, Springer, 2012, pp. 43–76.

[21] H. P. Edmundson, “New methods in automatic extracting,” *J. ACM JACM*, vol. 16, no. 2, pp. 264–285, 1969.

[22] P. B. Baxendale, “Machine-made index for technical literature—an experiment,” *IBM J. Res. Dev.*, vol. 2, no. 4, pp. 354–361, 1958.

[23] R. Barzilay and M. Elhadad, “Using lexical chains for text summarization,” *Adv. Autom. Text Summ.*, pp. 111–121, 1999.

[24] S. Harabagiu and F. Lacatusu, “Topic themes for multi-document summarization,” in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 202–209.

[25] E. Filatova and V. Hatzivassiloglou, “Event-based extractive summarization,” *Text Summ. Branches Out*, 2004.

[26] C.-Y. Lin and E. Hovy, “The automated acquisition of topic signatures for text summarization,” in *Proceedings of the 18th conference on Computational linguistics-Volume 1*, 2000, pp. 495–501.

[27] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, “Interpreting tf-idf term weights as making relevance decisions,” *ACM Trans. Inf. Syst. TOIS*, vol. 26, no. 3, p. 13, 2008.

[28] G. Erkan and D. R. Radev, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, 2004.

[29] E. Baralis, L. Cagliero, N. Mahoto, and A. Fiori, “GRAPHSUM: Discovering correlations among multiple terms for graph-based summarization,” *Inf. Sci.*, vol. 249, pp. 96–109, 2013.

[30] Y. Chali and S. R. Joty, “Improving the performance of the random walk model for answering complex questions,” in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, 2008, pp. 9–12.

[31] K. R. McKeown, J. L. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin, “Towards multidocument summarization by reformulation: Progress and prospects,” *Proc. AAAI-99*, 1999.

[32] R. Barzilay, K. R. McKeown, and M. Elhadad, “Information fusion in the context of multi-document summarization,” in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999, pp. 550–557.

[33] D. R. Radev, H. Jing, and M. Budzikowska, “Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies,” in *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*, 2000, pp. 21–30.

[34] D. R. Radev, H. Jing, M. Styś, and D. Tam, “Centroid-based summarization of multiple documents,” *Inf. Process. Manag.*, vol. 40, no. 6, pp. 919–938, 2004.

[35] S. Chakrabarti, M. Joshi, and V. Tawde, “Enhanced topic distillation using text, markup tags, and hyperlinks,” in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 208–216.

[36] M. A. Hearst, “Multi-paragraph segmentation of expository text,” in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 1994, pp. 9–16.

[37] D. Marcu, “From discourse structures to text summaries,” *Intell. Scalable Text Summ.*, 1997.

[38] U. Hahn and M. Strube, “Centering in-the-large: Computing referential discourse segments,” in *Proceedings of the eighth conference on European Chapter of the Association for Computational Linguistics*, 1997, pp. 104–111.

[39] J. Carbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 335–336.

[40] R. McDonald, “A study of global inference algorithms in multi-document summarization,” in *European Conference on Information Retrieval*, 2007, pp. 557–564.

[41] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999.

[42] J. Yao, X. Wan, and J. Xiao, “Recent advances in document summarization,” *Knowl. Inf. Syst.*, vol. 53, no. 2, pp. 297–336, 2017.

[43] B. Larsen, “A trainable summarizer with knowledge acquired from robust NLP techniques,” *Adv. Autom. Text Summ.*, vol. 71, 1999.

[44] C.-Y. Lin, “Training a selection function for extraction,” in *Proceedings of the eighth international conference on Information and knowledge management*, 1999, pp. 55–62.

[45] M. Fuentes, E. Alfonseca, and H. Rodríguez, “Support vector machines for query-focused summarization trained and evaluated on pyramid data,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 2007, pp. 57–60.

[46] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu, “Enhancing diversity, coverage and balance for summarization through structure learning,” in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 71–80.

[47] T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto, “Extracting important sentences with support vector machines,” in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002, pp. 1–7.

[48] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 406–407.

[49] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document Summarization Using Conditional Random Fields.," in *IJCAI*, 2007, vol. 7, pp. 2862–2867.

[50] D. Galanis, G. Lampouras, and I. Androutsopoulos, "Extractive multi-document summarization with integer linear programming and support vector regression," *Proc. COLING 2012*, pp. 911–926, 2012.

[51] P. Malakasiotis, E. Archontakis, I. Androutsopoulos, D. Galanis, and H. Papageorgiou, "Biomedical Question-focused Multi-document Summarization: ILSP and AUEB at BioASQ3.," in *CLEF (Working Notes)*, 2015.

[52] F. Schilder and R. Kondadadi, "FastSum: fast and accurate query-based multi-document summarization," in *Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies: Short papers*, 2008, pp. 205–208.

[53] R. Herbrich, "Large margin rank boundaries for ordinal regression," *Adv. Large Margin Classif.*, pp. 115–132, 2000.

[54] K. Crammer and Y. Singer, "Pranking with ranking," in *Advances in neural information processing systems*, 2002, pp. 641–647.

[55] C. Burges *et al.*, "Learning to rank using gradient descent," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 89–96.

[56] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, no. Nov, pp. 933–969, 2003.

[57] C. Shen and T. Li, "Learning to rank for query-focused multi-document summarization," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, 2011, pp. 626–634.

[58] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133–142.

[59] D. Metzler and T. Kanungo, "Machine learned sentence selection strategies for query-biased summarization," in *Sigir learning to rank workshop*, 2008, pp. 40–47.

[60] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang, "Learning query-biased web page summarization," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 555–562.

[61] M. R. Amini, A. Tombros, N. Usunier, and M. Lalmas, "Learning-based summarisation of XML documents," *Inf. Retr.*, vol. 10, no. 3, pp. 233–255, 2007.

[62] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[63] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," *ArXiv Prepr. ArXiv160307252*, 2016.

[64] Y. Dong, “A Survey on Neural Network-Based Summarization Methods,” *ArXiv Prepr. ArXiv180404589*, 2018.

[65] K. Svore, L. Vanderwende, and C. Burges, “Enhancing single-document summarization by combining RankNet and third-party sources,” in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[66] Y. Liu, S. Zhong, and W. Li, “Query-Oriented Multi-Document Summarization via Unsupervised Deep Learning.,” in *AAAI*, 2012.

[67] M. K\a agebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, “Extractive summarization using continuous vector space models,” in *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, 2014, pp. 31–39.

[68] W. Yin and Y. Pei, “Optimizing Sentence Modeling and Selection for Document Summarization.,” in *IJCAI*, 2015, pp. 1383–1389.

[69] Z. Cao, F. Wei, S. Li, W. Li, M. Zhou, and W. Houfeng, “Learning summary prior representation for extractive summarization,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, vol. 2, pp. 829–833.

[70] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, “Attsum: Joint learning of focusing and summarization with neural attention,” *ArXiv Prepr. ArXiv160400125*, 2016.

[71] A. Nenkova and R. Passonneau, “Evaluating content selection in summarization: The pyramid method,” in *Proceedings of the human language technology conference of the north american Chapter of the association for computational linguistics: Hlt-naacl 2004*, 2004.

[72] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[73] G. Tsatsaronis *et al.*, “An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition,” *BMC Bioinformatics*, vol. 16, no. 1, p. 138, 2015.

[74] C.-Y. Lin and E. Hovy, “Manual and automatic evaluation of summaries,” in *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, 2002, pp. 45–51.

[75] H. Dalianis, M. Hassel, K. de Smedt, A. Liseth, T. C. Lech, and J. Wedekind, “Porting and evaluation of automatic summarization,” *Nord. Sprogteknologi*, pp. 2000–2004, 2003.

[76] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” *Text Summ. Branches Out*, 2004.

[77] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation,” in *Encyclopedia of database systems*, Springer, 2009, pp. 532–538.

[78] M. Peyrard and J. Eckle-Kohler, “Optimizing an approximation of rouge-a problem-reduction approach to extractive multi-document summarization,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, vol. 1, pp. 1825–1836.

[79] R. Barzilay and N. Elhadad, “Sentence alignment for monolingual comparable corpora,” in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 2003, pp. 25–32.

[80] H. Daumé III and D. Marcu, “A phrase-based hmm approach to document/abstract alignment,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[81] T. Copeck and S. Szpakowicz, “Leveraging pyramids,” in *Proceedings of the Document Understanding Conference*, 2005.

[82] Y. Chali, S. A. Hasan, and S. R. Joty, “Do automatic annotation techniques have any impact on supervised complex question answering?,” in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009, pp. 329–332.

[83] C. Orăsan, “Automatic annotation of corpora for text summarisation: a comparative study,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2005, pp. 670–681.

[84] D. Marcu, “The rhetorical parsing of natural language texts,” in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 1997, pp. 96–103.

[85] A. Louis and A. Nenkova, “Automatically assessing machine summary content without a gold standard,” *Comput. Linguist.*, vol. 39, no. 2, pp. 267–300, 2013.