VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE II

Master's thesis

Automatic Text Summarisation: Does the Type of Text Matters When Selecting a Text Summarisation Algorithm?

Done by:

Aleksandr Christenko signature

Supervisor:
Lekt dr. Linas Bukauskas

Vilnius
2018

Contents

Abstract

Currently majority of research in automatic text summarisation focuses on creating new or improving existing text summarisation models. However, this trend is rather dangerous as they do not ask a question: if the algorithm they are trying to improve suits the data in the first place? This master's thesis tries to fill in this gap in the literature by assessing if different types of text summarisation algorithms perform equally well with academic papers from different scientific disciplines. More specifically, it applies Pivoted QR Decomposition, Naïve Bayes, Decision Tree, Hidden Markov Model, and Support Vector Machine (SVM) text summarisation algorithms on academic papers from the fields of medicine, biology, computer science, and economics (including finance), and compares the results. According to the research, the academic field does not have an influence on the accuracy of summarisation. By applying different algorithms on different texts, independently of the text type the same algorithms (Naïve Bayes and SVM) performed the best. However, at the same time, when applying different algorithms on aggregated data, in general the accuracy achieved was smaller than when applying it on separate texts. Hence, it can be concluded that though the type of text does not influence the summarisation that much, it is better to use separate data than an aggregate in order to achieve higher summarisation accuracy.

Santrauka

Automatinis teksto apibendrinimas:
ar svarbu žinoti teksto tipą renkantis teksto apibendrinimo algoritmą?

Šiuo metu dauguma tyrimų apie automatinį tekstų apibendrinimą bando sukurti naujus arba patobulinti jau egzistuojančius algoritmus. Bet jie ignoruoja labai svarbų klausimą: ar siūlomas algoritmas apskritai tinka tyrinėjamiems duomenims? Šis tyrimas bando užpildyti šią spragą akademinėje literatūroje išsiaiškinant ar skirtingi algoritmai veikia vienodai gerai su skirtingais teksto tipais. Atsakymas yra pasiekiamas apibendrinant medicininius, biologinius, ekonominius ir kompiuterinio mokslo akademinius darbus, naudojant pasukimo QR suskalbimą, Naivaus Bajeso, Sprendimo Medžius, Paslėpto Markovo Modelio ir Vektoriaus Palaikymo Mašinos (SVM) algoritmus. Tyrimas parode, kad teksto tipas neturi dideles įtakos apibendrinimo tikslumui. Nepriklausomai nuo to kokie algoritmai buvo taikomi kokiems tekstams, visados tie patys algoritmai (Naivusis Bajesas ir Vektoriaus Palaikymo Mašinos) apibendrino duomenis geriausiai. Tačiau, taikant algoritmus ant agreguotų duomenų dažniausiai apibrandinimo tikslumas mažėjo. Tai reiškia, kad nors teksto tipas nedaro įtakos algoritmams, vištike galima rekomenduoti, kad teksto apibendrinimas turėtu būti daromas su neagreguotais duomenimi, kad išgauti aukščiausią tikslumą.

Introduction

Due to a huge information overload that was brought by the invention of the internet, currently there exists a large demand for systems that can accurately summarise the available information [4, 14]. In recent years, the academia met this demand by creating several complex and accurate text summarisation models (e.g., MEAD [19], WebInEssence [18], GISTExter [21], etc.). In addition, large number of researchers proposed several improvements to already existing models. Suanmali, Salim & Binwahlan (2009)[11] and Babar & Patil (2015)[20] proposed to use fuzzy logic when deciding which sentences should be extracted, while Alguliev (2009)[16] recommends using an evolutionary method based on clustering to improve the quality of summaries.

However, currently majority of research focuses on creating new or improving existing text summarisation models, without answering a question to what types of texts any particular model should be applied in the first place. For example, Hiraeo et al (2002)/citeHirao applied Support Vector Machine text summarisation algorithm and Kim et al (2006) [22] applied Naïve Bayes algorithm on news reports, without distinguishing them by type. Similarly, Conroy and O'leary (2001) /citeConroy used a Hidden Markov Model to summarise an aggregated data set that included different types of documents, without separating them by type. Hence, with my master's thesis I will try to check, if such an approach of ignoring the type of data is a sound one. This will be checked by assessing if different text summarisation models perform equally well with academic papers from different scientific disciplines.

The answer is provided by comparing the quality of summaries applying Pivoted QR Decomposition, Naïve Bayes, Hidden Markov Model (HMM), Decision Tree, and Support Vector Machine (SVM) text summarisation algorithms on academic papers from the fields of medicine, computer science, biology, and economics (including finance). According to the research, the type of academic discipline does not have a substantial impact on the quality of the summary. More specifically, in all cases Naïve Bayes and Support Vector Machine performed the best, without any huge differences between the two, independently on the type of document. After performing sensitivity analysis, this conclusion was further strengthen, though it also showed that in some cases SVM loses its accuracy and Naïve Bayes becomes the dominant algorithm.

However, at the same time, the research also shows that better accuracy is achieved when academic papers from different disciplines are summarised separately rather than using their aggregates. In other words, when the data was aggregated, it almost always performed worse than when it was separated. Hence, this implies that though the best algorithms stays the best independently on the text type, it is better to summarise texts from different disciplines separately rather than aggregating them.

In this paper, the first section provides a short overview of text summarisation. This includes an explanation on what type of text summarisation exists, what text summarisation algorithms will be used in the analysis and how they work, and how currently majority of text summarisation algorithms are applied in practice. Second section explains the methodology that is used in the research, which includes explanation how the data is cleaned and prepared for the research, a short overview of the algorithms, and how the results are further checked in order to add robustness. Third section elaborates on the experiment that is used to check if the type of text has an influence on summarisation results. The master thesis ends with a conclusion to the whole work as well as a short discussion on the future work that could be done in this field of research.

1 Overview of text summarisation

According to Hovy (2005)[7] a summary is a text that: (i) is created from one, or more, document, (ii) contains a significant portion of information found in the full text, and that (iii) is at least two times shorter than the original text (most often it is much more shorter then that). In other words, a summary is a short text that retains all the important ideas of the original document. Which ideas are important often depends on the aims of the researchers that are summarising the documents. For example, if an academic wishes to build an algorithm that could help medics to diagnose diseases, for a medic important idea in this paper could be “for what diseases this model can be used”, while for a computer scientists the important idea could be “how the model works”. Nevertheless, the main point of any summarisation algorithm is to skim the fat from any text and only reveal what really matters.

In general, there are two ways to skim the cream and create a good summary of a text:

- **Extract based** - constructs the summary by extracting the most prevalent / important phrases, sentences or words from the text [6].

- **Abstract based** - automatically finds the most relevant parts of the text and paraphrases them in to a summary [8].

As can be seen from the short explanation of the two methods, there are quite a lot of difference between them. On one hand, extract based method does not alter the original text in any way. It only finds which sentences are important and which are not. On the other hand, abstract based method uses the words and ideas found in the text, but it creates its own text by paraphrasing them. Hence, for the abstract based method to create good summaries it is very important that it does not misinterpret the ideas found in the original text, which is very difficult to do. Because of this difficulty, many scholars choose to use extract based rather than abstract based methods in their work.

This research will follow in the footsteps of the masses and use extract based approach to summarise academic papers. In other words, due to the abundance of literature and its relative simplicity extract based methods will foster a more transparent research. As the main goal of this paper is not to improve an already existing model, but to check how they perform on different data, the relatively simplicity of extract based models becomes a strength rather than a weakness. In addition, as one of the goals of this research is to provide a guideline how the quality of a text summarisation model could be checked, by using a simpler approach the guidelines become accessible to more people.

The remainder of this section names and explains concrete algorithms that will be used to summarise texts as well as elaborates on the already existing applications of defined extract based text summarisation models.

1.1 Extract based text summarisation models

Majority of extract based text summarisation models come from data mining. The most well-known and widely used text summarisation models are Naïve Bayes, tree based algorithms (e.g., Decision Tree), Support Vector Machine, and Hidden Markov Models. However, there are also algorithms that do not use data mining approaches to find which text is a good summary. One of such algorithms is Pivoted QR decomposition. The remainder of this section elaborates on all of these models mentioned prior and explains how they will be applied in this work.

1.1.1 Pivoted QR decomposition

Pivoted QR decomposition is a method of text summarisation where sentences that have many ideas are considered as good summaries for the text. This approach assumes that the level of importance of each sentence heavily depends on the number of topics it covers. Sentences that have a large number of ideas are more important than those with a small number, and hence, are extracted as summaries. In addition, the pivoting aspect of this model means that each time a sentence with a particular set of ideas is extracted as a summary, the ideas extracted do not play a role when deciding which other sentences should be extracted. This means that each new sentence extracted following this approach talks about different topics found in the text.

Keeping what was discussed in mind, the most important question of this text summarisation approach is the definition of an idea. Conroy and O'leary (2001)[13] use this model to summarise text and they defined an idea as any term in the document. In other words, each word in a document is an idea, and hence should be used to estimate the importance of each sentence. This approach is very simple and because of that has several major issues.

First, not all words are equally important in a text document. For example, importance of any stop word in a document should be zero, as they do not add any additional information to the text (though they make reading documents much simpler). In addition, if a paper is about some statistical model, words that describe the model are much more important that those that only have a seldom relationship to it (e.g., words that are used to explain the model through examples).

Second, the approach used by O'leary and Conroy (2001)[13] has a positive bias toward longer sentences. Though it can be speculated that long sentences have more ideas in them this is not always the case. For example, a sentence that is used as an example could be very long, but at the same time it is mainly used to clarify some other idea. In other words, though the example sentences could be very long, it often has only a very small number of unique ideas in it.

Hence, to combat the issues discussed previously the idea is defined in this paper as follows: an idea of an academic paper is a word that was mentioned in the abstract at least once, ignoring any stop words. By using this approach: (i) only words relevant to the main topics of the documents are accounted for and (ii) the model becomes unbiased towards longer sentences that only talk about a small number of ideas. The following pseudocode contextualises the ideas describe prior:

Algorithm 1. Pivoted QR Decomposition

```
$terms \Leftarrow import\ all\ uniquel\ words\ (excluding\ stopwords)\ that\ appear\ in\ the\ abstracts$
$terms.df \Leftarrow term\ matrix\ with\ terms\ as\ col\ names\ and\ sentences\ as\ row\ names$
$n \Leftarrow number\ of\ sentences\ that\ will\ be\ used\ as\ a\ summary$

for $i\ in\ 1$ : ncol$(term.matrix)$ do
  $which.have \Leftarrow$ grep(col.names$(terms.df)[i]$, row.names$(terms.df)$)

  if length(which.have) > 0 then
    $terms.df[which.have, i] \Leftarrow term.matrix[which.have, i] + 1$
  end if
end for

$summary \Leftarrow c()$
for $i\ in\ 1 : n$ do
  $summary \Leftarrow$ which.max(rowSums$(terms.df)$)
  $terms.df \Leftarrow terms.df[, -(which)(terms.df[summary, ]! = 0)]$ {Removing terms that were used}
  $summaries \Leftarrow c(summaries,$ which.max(rowSums$(terms.df)$))
end for
```

Here the first part of the script imports the terms mentioned in the abstracts of documents for one of the four fields of science as well as create a data frame which uses the important terms as col names and sentences from the texts as row names. After that the first*for* loop counts the number of times each of the terms appears in each sentence. Second*for* loop, finds which sentences have the largest amount of information, but also at each iteration the algorithm removes which ideas were represented in the sentence that was deemed as the best at each step.

1.1.2 Naïve Bayes

Naïve Bayes is one of the most popular text summarisation techniques due to its simplicity and its relatively good summarisation capabilities [22]. It uses the Bayes rule in order to classify each sentence in a text as worthy of extraction or not [4]. However, for Naïve Bayes to work, first each sentence has to be described in terms of features (e.g., length, number of keywords in a sentence, cosine distance between the sentence and the title, etc.). That is, it cannot operate with unstructured data. Naïve Bayes algorithm uses these features to calculate the probability that the sentence is worthy of extraction (or not). The function below is used to decide the aforementioned worthiness of a sentence. In the function *F* are the features, *s* is a sentence identifier and *S* is a set of all sentences in a document:

$$P(s \in S|F_1, F_2, ..., F_k) = \frac{\prod_{i=1}^{k} P(F_i|s \in S) \cdot P(s \in S)}{\prod_{i=1}^{k} P(F_i)} \tag{1.1}$$

After assessing the worthiness of all sentences, some predefined number of sentences is extracted and used as a summary. The number of sentences that will be extracted depends on the compression rate. Compression rate is how long is the summary compared to the original text. If the compression rate is 50%, that means that the summary is two times shorter than the original text.

In addition, Naïve Bayes is a supervised learning method. This means, it has to have a dataset on which it can be trained. The training data is a dataset in which some sentences are already classified as worthy and others as not worthy. In the research, this kind of dataset will be created by finding the sentences in the text that are most similar to already existing man made summaries of said text. More on this can be found in the subsection *2.4 Creating the training data set.*

1.1.3 Tree based methods

One of the major drawbacks of the Naïve Bayes approach is that it cannot identify which features really matter for sentence reduction, and which do not. In other words, every feature is treated equally, ignoring the fact that some of them could have a more substantial effect on classification than others. This problem is solved by tree based methods. Their name stems from the fact that they create hierarchical structures in order to classify the data. Majority of them work by using entropy measures (i.e., measure of homogeneity of a dataset according to the variance of some variable) to find which features really matter. More specifically:

1. Tree based methods assume that the whole dataset is the root of the tree.
2. The data set is then split in to two, or more, parts using every variable and value of each variable (e.g., if one variable is number of words in a sentence, first split could be "all sentences that have more than 0 words", second could be "all sentences that have more than 1 word", etc.).
3. For each split, the newly created groups are evaluated based on their purity level (i.e., what is the balance of the two, or more, classes in each of the newly created groups).
4. The variable and value of said variable that produces the most pure split is considered as the best split.
5. After creating the first split the process is repeated (step 2 through 4) on every newly created group, until the algorithm does not create splits with high purity level (i.e., if the algorithm cannot find a split that would separate the data in to two groups where at least one of the newly created groups is dominated by one of the target classes).

Hence, at each level new rules are created as the data is divided, where in the end of each branch of a tree there is a subset of the original dataset. How any new observation will be classified depends on the ratio of classes in the final subset to which it belongs. That is, dominant class is the class to which any observation belonging in the subset will be assigned. Figure below provides a visual representation of this idea. In addition, the figure shows one of the main advantages of decision tree compared to other methods. That is, it can use the same feature several times at different levels of a tree. This trait of tree classifiers often allows it to outperform more simple classification methods such as Naive Bayes.

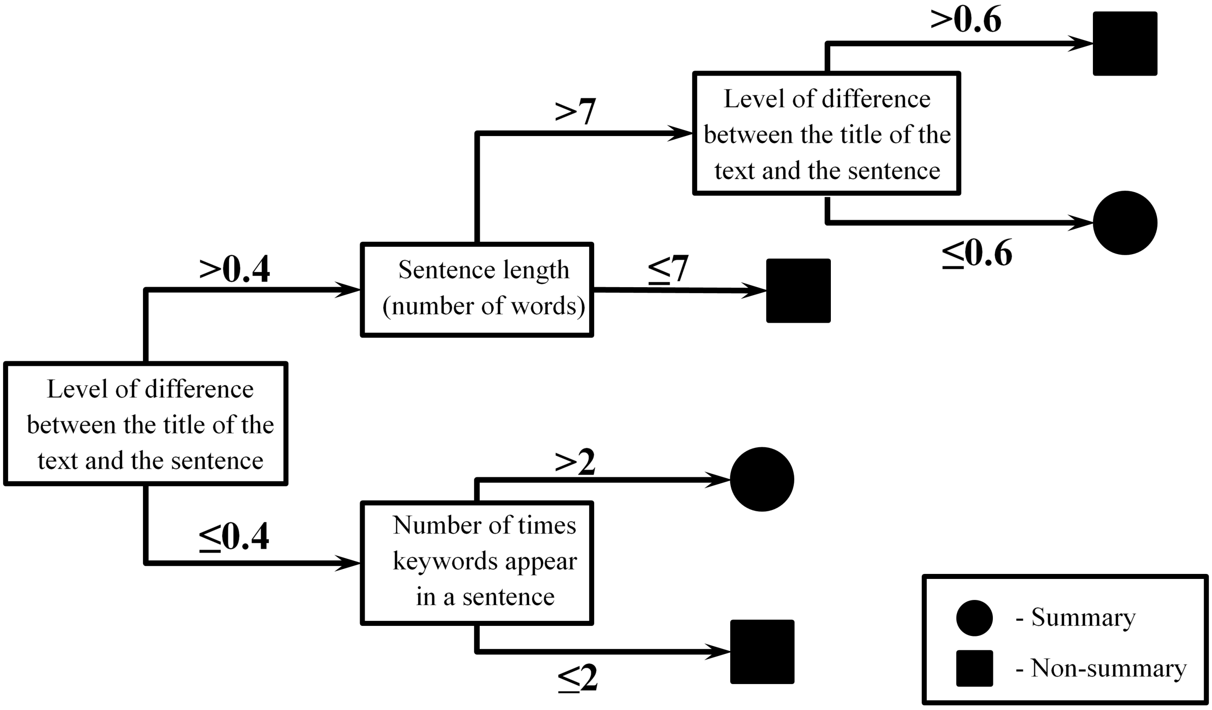

>0.6
Level of difference between the title of the text and the sentence
>7
≤0.6
>0.4
Sentence length (number of words)
≤7
Level of difference between the title of the text and the sentence
>2
≤0.4
Number of times keywords appear in a sentence
≤2
● - Summary
■ - Non-summary

Figure 1. This graph represents a hypothetical example of a tree classifier that can be built to find which sentences should be classified as summary sentences and which as non-summary. At the root of the tree (first question) the full dataset is used. It is divided in to smaller groups as it moves down the decision tree. At the terminal nodes (a.k.a. leaf nodes) there is only a small set of the original data. The terminal node is assigned to be summary or non-summary based on the ratio of summary to non-summary sentences in the data set at the terminal node (i.e., if there are more summary sentences than non-summary the terminal node is classified as summary and vice versa).

This research will use the tree based method called Decision Tree. It was selected due to its simplicity, difference from the Naïve Bayes approach as well as the algorithms good track record in text summarisation. In addition, decision tree approach is a backbone behind the Random Forest classifier that is considered as one of the best out-of-the-box classifiers (i.e., a classifier that does not require a lot of tuning) currently available. Hence, future research could easily compliment this research by growing a forest from the decision tree classifier.

1.1.4 Hidden Markov Chain

Hidden Markov Model (HMM) is an algorithm that can find hidden states of some observation by observing the features of each state. In recent years, this model gained prominence in the text summarisation community as it, unlike other models discussed previously, does not assume independence between sentences. In other words, it assumes that the probability that a particular sentence is important partially depends on the fact if the previous sentence was important or not. Main proponents of this model are Conroy and O'leary (2001)[13] that found that HMM can create very respectable results when applied to text summarisation problems. Figure below provides a graphical representation on how HMM can be used to extract summary sentences from a six sentence text.

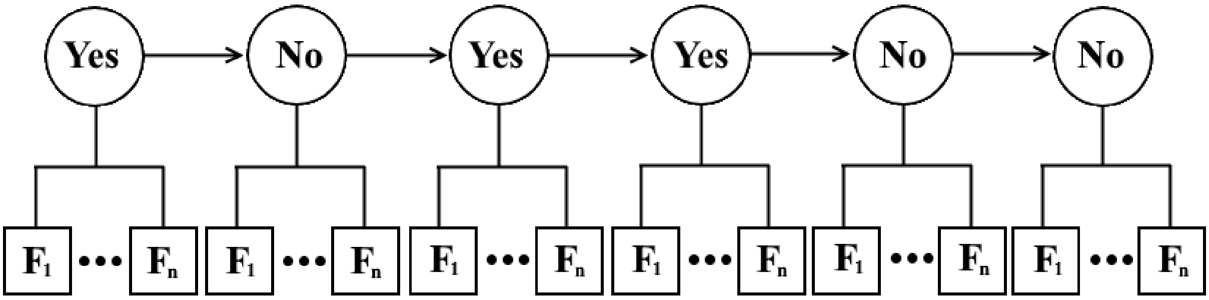


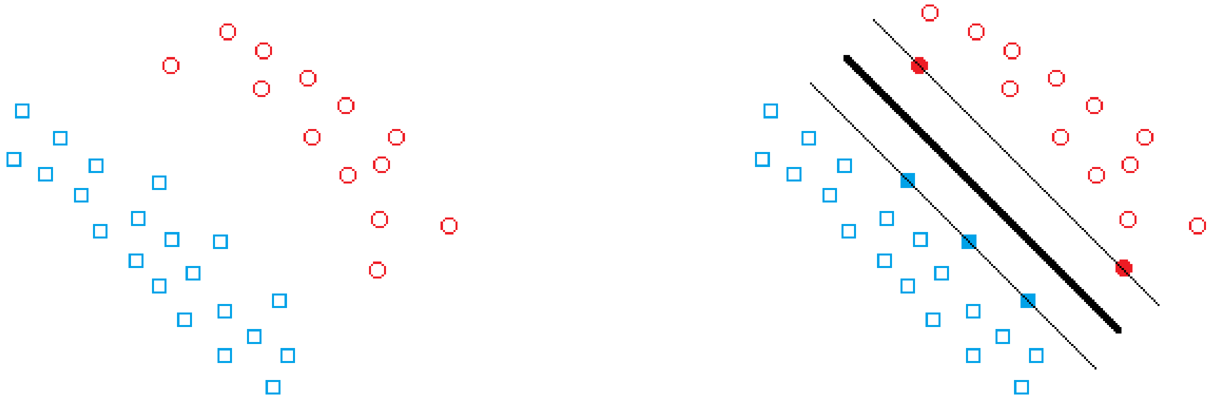
Figure 2. Summary extraction via Hidden Markov Model. Here The bubbles represent sentences, yes means sentence is extracted and no means it is not extracted; F_1,...,F_n are the features of each sentence. HMM looks at each sentence through the features and estimates if a sentence is a good summary or not. Here the hidden part is the state (summary and non-summary).

In this paper I will use the approach proposed by Conroy and O'leary (2001)[13] to summarise academic papers. Any HMM consists of three parameters: (i) the probability *p* that the first sentence is or is not a summary, (ii) the transition matrix *M* between the states, and the emission matrix *E* of observable features. In this research the probability *p* will be calculated by dividing the number of times the first sentence was classified as important by the total number of documents. The transition matrix *M* will be estimated by calculating how many times a sentence transitioned from a summary state to non-summary state, from a non-summary state to a summary state, and how many times it stayed the same (summary-summary and non-summary-non-summary). Then the estimates are divided by the total number of transitions to create the transition matrix *M*. The emission matrix *E* is estimated by calculating the covariance of all features in each group (i.e., summary and non-summary sentences). Important to note that in majority of cases the emission matrix is estimated by looking how each feature interacts with each state (e.g., by looking at the probabilities or correlation). However, following Conroys' and O'learys' (2001)[13] I will be using the covariance matrix instead. Important to note that by using the covariance matrix it is assumed that the features have a multivariate Gaussian distribution. However, this assumption is not farfetched, as Conroys' and O'learys' (2001)[13] showed in their work.

1.1.5 Support Vector Machine

Support Vector Machine (SVM) is a strong classification algorithm that often outperforms both tree based and Naïve Bayes text summarisation algorithms [10, 24, 2]. However, it is also much more computationally intensive. SVM is based on the Structural Risk Minimization principle, which tries to find a hypothesis with the lowest error. In more general sense, SVM tries to find an n-dimensional surface (hyperplane) that would separate the data the best according to some variable. The plane is constructed by selecting several points (i.e., observations) that are called support vectors [2]. Figure 2 below visualises this idea graphically.

The idea that the figure below demonstrates is that SVM finds a hyperplane that allows to divide the data in to two groups the best. After it finds this plane, any new point that would be added to the dataset would be classified to one or another group, depending where in relationship to the plane this point would appear. One of the main strengths of SVM is that this plane does not have to be straight. In other words, it can be wavy, which allows for more precise classification. However, because of this aspect of the hyperplane also it can make SVM over fit itself too strongly to the data.



(a) Two group example scatterplot

(b) Point seperation after SVM

Figure 3. The two plots provide an example how any numeric data set can be separated via SVM. Figure (a) on the left shows a scatter plot of random variables that are divided in to two groups. First group is represented as blue squares and the second group as red circles. Second figure (b) on the right shows how the provided data could be divided in to the aforementioned groups. The black line in the middle is the 2-dimensional surface discussed prior (hyperplane), which essentially divides the data points in to two groups. Shapes that have their respective collar filled in are the aforementioned support vectors.

1.2 Applications of text summarisation methods

Currently, majority of researchers in text summarisation focus on summarising news reports [23], legal documents [3] and medical documents [1, 17]. One of the main reasons why the focus is on these types of documents is that there are several large text data sets in these topics that already have summarised sentences. For example, TIPSTER data set, has a large number of news reports spanning the years of 1989 and 1992. The news reports also include short summaries and each of them is classified according to some topics. Hence, by using these types of data researchers can easily both apply their models on them as well as check the quality of summaries by comparing them to manmade summaries.

In addition, researchers rarely use different types of documents to test their created or improved text summarisation algorithms. For example, Hiraeo et al (2002) evaluated their Support Vector Machine algorithm solely using news reports [24]. Similarly, Kim et al (2006) evaluated the changes that they proposed to a standard Naïve Bayes summarisation algorithm by also using news reports [22]. Alternatively, Conroy and O'leary (2001) used a variety of different text in order to check their Hidden Markov Model text summarisation algorithm [13]. However, they did not separate texts by type and simply aggregated all the results.

Finally, researchers that try to propose new or improve existing text summarisation algorithms almost always only look at the algorithms themselves, ignoring the data. Though currently this approach is not seen as problematic, in reality it could have several negative implications. For example, if the type of document does in fact influence the results of summarisation, to an extent that one algorithm performs better with one type of documents while another with other type, the current approach could be a poor estimate of the quality of a new or improved algorithm. More specifically, by only applying text summarisation algorithms on one type of document, the algorithm that is deemed better, could be simply better for a particular data set. In other words, this type of comparison does not allow to answer if algorithm A is better than algorithm B with any certainty.

Keeping this in mind, this master's thesis tries to fill in the gap in the literature by testing if in fact the type of document does have an impact on results of summarisation. In total five text summarisation models will be used, namely: (i) Pivoted QR Decomposition, (ii) Naïve Bayes, (iii) Hidden Markov Model, (iv) Decision Tree, and (v) Support Vector machine. The five algorithms will be applied on eighty academic papers from four scientific disciplines, namely: (i) medicine, (ii) biology, (iii) computer science, and (iv) economics (including financial papers). By doing so, the research tries to provide an answer to the question, which is also the main hypothesis: Can the quality of text summaries be improved by selecting different text summarisation models for different types of texts?

In addition, the master thesis also provides a guideline on how the quality of new or old but improved text summarisation methods can be checked. The approach provided here, unlike the one mentioned prior, does not rely on simply using an aggregate dataset to test algorithms. It uses several datasets, as well as it includes sensitivity analysis. In other words, the approach described in this paper circumvents the issue of not knowing if an algorithm truly works better than others, or it simply works well with particular data.

Hence, the master thesis not only checks if the quality depends on the type of text, but also provides a guideline on how the quality of any algorithm should be checked. Though the approach described in the master thesis is only applied on academic papers, it can also be applied on other type of work. For example, it can be applied to find an algorithm that best summarises sci-fi books or romantic comedies and if the same algorithm performs the same way with both of these types of books. It could be the case that in both of these cases the same algorithm will perform the same. However, without performing a deep analysis it is impossible to say if this is the case.

The next subsection (*2 Methodology*) elaborates on the approach that was used in the research to check if the quality of summaries depend on the type of text. It also provides an in depth explanation on how text summarisation algorithms could be checked using different types of text.

2 Methodology

This section is divided in to seven parts:

1. **Data -** elaborates on the data (i.e., academic papers) that will be used in the analysis
2. **Data processing -** explains how data is cleaned and processed before using it in the analysis
3. **Estimating features -** defines the features of sentences that will be used to find collection of sentences that can serve as a summary for the whole text
4. **Creating a training data set -** this section elaborates on how by using the abstracts and cosine distance measure it is possible to find sentences in the text that are good summaries, and in turn create a training data set for the algorithms
5. **Finding summary sentences -** explains the technical aspects of the five text summarisation algorithms (see *1.1 Extract based text summarisation methods*) that will be used in the research
6. **Calculating accuracy -** provides an overview of the approach that will be used to estimate the quality of summaries
7. **Sensitity analysis -** elaborates on the approach that was used to check the sensitivity of the results

In addition, figure below provides a short overview of the methodology, while the rest of this section elaborates on it in more depth.

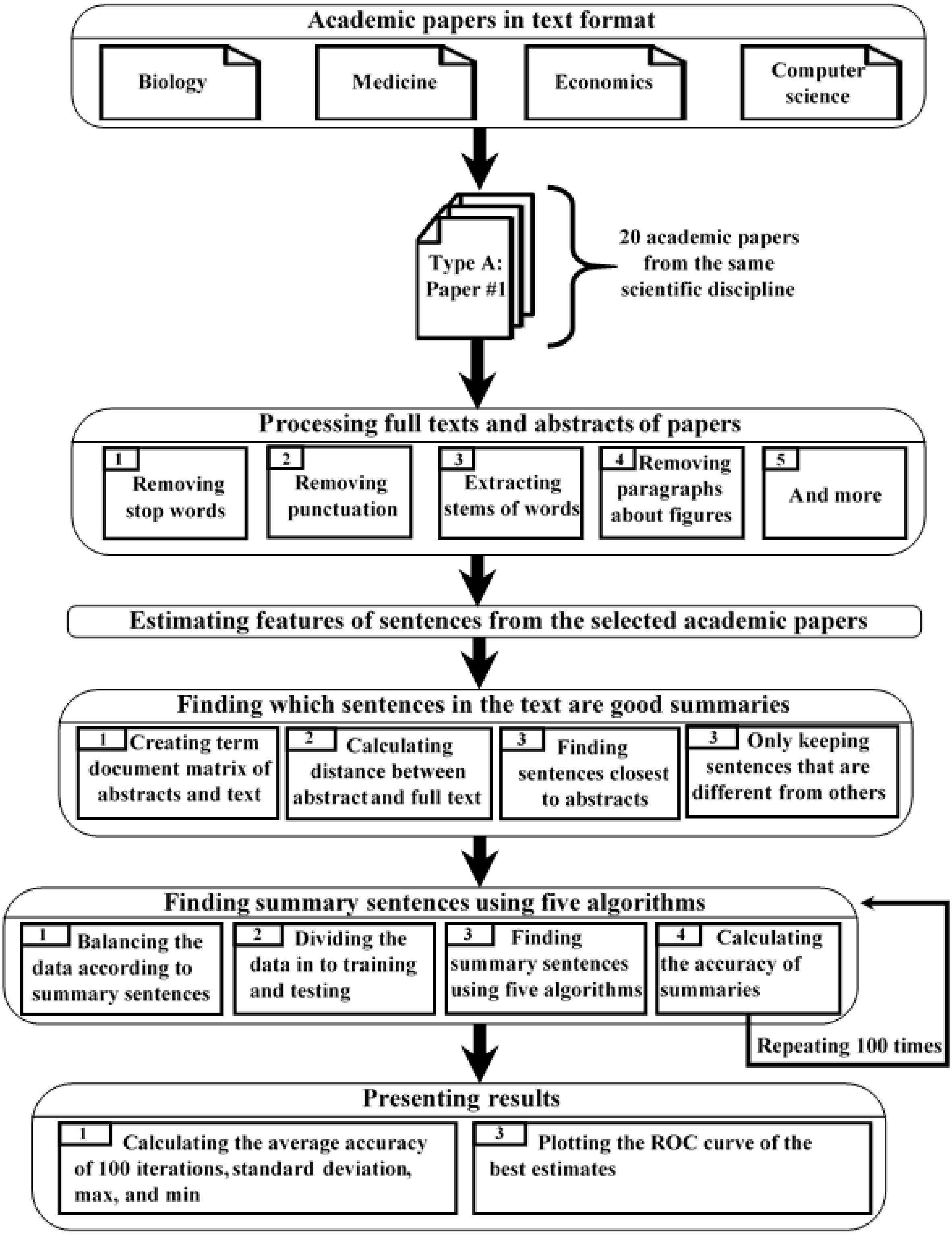

Academic papers in text format
Biology
Medicine
Economics
Computer science
Type A: Paper #1
20 academic papers from the same scientific discipline
Processing full texts and abstracts of papers
1 Removing stop words
2 Removing punctuation
3 Extracting stems of words
4 Removing paragraphs about figures
5 And more
Estimating features of sentences from the selected academic papers
Finding which sentences in the text are good summaries
1 Creating term document matrix of abstracts and text
2 Calculating distance between abstract and full text
3 Finding sentences closest to abstracts
3 Only keeping sentences that are different from others
Finding summary sentences using five algorithms
1 Balancing the data according to summary sentences
2 Dividing the data in to training and testing
3 Finding summary sentences using five algorithms
4 Calculating the accuracy of summaries
Repeating 100 times
Presenting results
1 Calculating the average accuracy of 100 iterations, standard deviation, max, and min
3 Plotting the ROC curve of the best estimates

Figure 4. This graph presents the methodology of thr research in a stylised form. It in no shape or form elaborates on every aspect that was done in the research. Hence, it only should be used as a reference tool to the methodology.

2.1 Data

In the research academic papers and articles from four different disciplines were used in order to assess if the type of text has an influence on summarisation results. Namely academic papers and articles from the fields of: (i) Biology, (ii) Medicine, (iii) Computer science, and (iv) Economics (including papers from the field of Finance). In total eighty academic papers are used in the research. For the full list of academic papers see the Appendix.

The reason why academic papers were selected as the target of the analysis is because they often are written in a technical manner. In other words, they rarely have ambiguous sentences, which is often the case with fictional literature. In addition, because they often have to follow similar structures, stylistically they tend to be similar. All these aspects foster an easier summarisation process. Finally, in the academic literature on text summarisation, often these types of documents are rarely used in the research (see section *1.2 Application*). Hence, by focusing on academic papers this research also fills in the gap in the literature.

The four types of academic papers were selected as they allow for all rounded research. The remainder of this subsection discusses this in more depth.

2.1.1 Academic work in the field of Biology

These works mainly explore different biological phenomenon such as: (i) DNA, (ii) role of glycine betaine, (iii) exploration of psychogenetic signals, etc. Hence, quite often these types of papers overlap with papers in the field of Medicine, though there are still some differences between the two. However, because the similarity exists, by including papers in this field and the field of medicine it is possible to assess how different types of algorithms perform with very similar papers. At this stage, it is expected that the accuracy of summarisation of these two types of papers will be similar. However, it is also expected that the quality of summaries will not be very large, as these kinds of papers often use complex statistics and/or computer models in order to convey, predict, and describe different biological phenomenon. This is especially the case with phenomenon that can be expressed in signal form, such as psychogenetic signals.

From the descriptive perspective, the average number of sentences that the selected papers in biology have is 196.75 with a standard deviation of 107.04. The large standard deviation shows that these papers heavily differ in terms of length. Paper with the largest number of sentences had 501 sentences, while the shortest paper had 61 sentences. These results strengthen the assumption that papers in the field of Biology heavily vary in terms of length. In addition, average length of a sentence in the papers was 15.49 words, with the standard deviation of 7.43. The longest sentence in the paper had 166 words. This sentence is this long because it is an introductory sentence with 4 bullet points separated by a semicolon.

2.1.2 Academic work in the field of Medicine

These types of papers are similar to those in the field of biology. However, unlike papers in biology, these works mainly focus on biological phenomenon related to people. For example: (i) psychical activities of people, (ii) illnesses, (iii) diagnostic tools, etc. In addition, it is also expected that the summarisation results of these papers will be better than papers in biology, as they have a narrower scope as the formal. However, the fact that these papers often include complex mathematical equations and formulas will also make summarisation more difficult.

Average number of sentences in a text is 145.7, with the standard deviation of 58.3. Maximum number of sentences was 264, while the minimum was 49. These results show that these papers are on average shorter than those in the field of biology. In addition, this also implies that these types of papers will have better summaries as the extract based text summarisation algorithms will only have to select sentences from a smaller pool of sentences. The average number of words in a sentence was 15.33, with standard deviation of 7.09. These results are very close to papers in the field of biology. The longest sentence in these types of works had 69 words.

2.1.3 Academic work in the field of Computer Science

These papers are related to different types of computer models, including but not limited to: (i) information systems, (ii) object detection, (iii) face recognition, etc. Similarly to the paper types before this one, there should be some overlap between them. This is because often both papers in the fields of biology and medicine use computer models to research and explain different biological and medical phenomenon. However, the overlap should not be too large. In addition, it is likely that these types of papers will be the most difficult to summarise as they almost always include complex mathematical formulas and pseudo code, which the extract based algorithm cannot reliably extract.

The average number of sentences was 286.75, with standard deviation of 142.23. Maximum number of sentences was 541, while the minimum was 38. These results again strengthen the assumption that these types of works will be the most difficult to summarise as they are also the longest. The average length of a sentence was 13.78 words, with the standard deviation of 6.54. The longest sentence had 104 words. This sentence is this long because it had over twenty references.

2.1.4 Academic work in the field of Economics (including Finance)

These papers explore issues related to economics and finance, such as: (i) the stock market, (ii) different policies (especially fiscal), (iii) natural resources, etc. These papers are the most different from others as they have the least amount of overlap. In addition, they are the most likely to have the largest accuracy, as these types of papers often do not define economic phenomenon using complex mathematical functions. For example, impact of any policy on the economy is often estimated using a simple ordinary least squares regression or the variant of one.

Papers in the field of economics (including finance) had an average number of sentences of 268.65, with the standard variance of 86.6. This means that, in length they are very similar to papers from the field of computer science, though the standard deviation here is smaller. Hence, this nuance of the data will allow to somewhat estimate what influences the results of the summarisation more, the length of the document or its complexity. Maximum number of sentences was 455 and the minimum was 157. The average number of words in a sentence was 14.68, with the maximum number of words of 83.

It is important to point out that even though the papers from different disciplines are of different length, this does not have a huge adverts effect on the research. This is because in the research the most important thing is to see how different algorithms perform with different types of texts. More specifically, the point of interest is that if one algorithm performs the best with one type of text, while another with another types of text. Hence, it is unnecessary to compare algorithms between types of texts. It is enough to simply look at the performance of algorithms inside each text type.

2.2 Data processing

Before analysing the academic papers, the data went through a process of cleaning, where any unnecessary information that could have an adverse effect on the results of summarisation were removed. However, by removing unnecessary information it would be quite difficult to interpret the created summaries (e.g., it is more difficult to read a text that does not have stop words, than those with them). Hence, before processing, the original sentences were saved. In other words, the estimation of features and selection of sentences that should be classified as summaries will be done using the clean sentences, while the summaries themselves will be constructed using original (i.e., not cleaned sentences).

The papers were cleaned using the following approaches:

1. **Removing stop words (e.g., a, an, and):** the stop words found in the text rarely carry any crucial contextual information. In addition, the amount and position of stop words is not only dictated by good grammar, but also by the style of the author. Hence, by removing them, the differences in style of different papers are mitigated, at least to an extent.

2. **Removing punctuations (excluding commas):** similarly to stop words, punctuations rarely carry any crucial contextual information. They help readers to grasp the concepts and ideas described in the papers, however they do not influence if a sentence is a good summary or not. In addition, similarly to stop words, they often are influenced by the style of the author. Hence, by removing them, it becomes easier to analyse the texts. However, it has to be also mentioned that punctuations themselves can help to identify different ideas. This is because often one sentence has several ideas, where each of them is divided by different punctuation marks, but analysis of the texts in such a way is already beyond the scope of this master's thesis, and hence it will be omitted.

3. **Removing sentences that describe figures, plots, and tables:** because the extract based method only extracts text and not figures, plots, or tables, the sentences that describe them will lose their meaning. In other words, even if these sentences describe profound ideas, without the reference point they are not very useful. Hence, they are removed.

4. **Only keeping the stems of words:** by only keeping the stems of words, words that use different affixes but talk about the same idea will be treated equally. For example, words such as mathematician and mathematicians in the research would be treated equally. Though these two words describe different things, the idea behind them is very similar (a person or several people from the field of mathematics). Hence, by stemming the words, words will not be separated simply because of the affixes.

5. **Additional cleaning:** a number of additional steps were performed in order to clean the data throughout. For example, all plots were removed, as well as acknowledgments, reference, and similar from the documents. The reason why all of this was removed before the analysis, is that these parts of the documents do not have any useful contextual information. Though it has to be mentioned that references can be used to pinpoint the topic of the document, this task is beyond the scope of this research.

In addition, for easier analysis the text documents are transformed in to a tabular data set. More specifically, first of all, paragraphs from each document are extracted and placed in to a tabular data set. This step is necessary as one of the features that will be used in the research is the position of a sentence in a paragraph (see section *2.3Estimating features* for more information). After that, the cleaned sentences are extracted from the paragraphs and placed in to a final tabular dataset. For each document type the algorithm created by the author creates a separate tabular dataset. The final tabular datasets are later used for feature selection, which is described in the next subsection. The cleaning process was performed using an R script written by the author and employing tm package to remove stop words and RWeka to extract word phrases (around 95% of the cleaning script is authors own work).

2.3 Estimating features

As majority of the mentioned text summarisation methods use features in order to select sentences that should be extracted (this does not apply to the Pivoted QR Decomposition as it simply uses terms) it is important to define features that describe each sentence in great detail. However, the features should not be overcomplicated as often a small number of relevant features is enough to accurately identify important sentences [4]. Hence, in the research a total of six different features is used:

- **Sentence position -** the position of a sentence in a paragraph. The position is expressed in a value from 0 to 1. First sentence of a paragraph is denoted as 0, last denoted as 1 and the sentences in between are estimated by calculating the relative position of the sentence in a paragraph. This feature was selected as it is assumed that first and the last sentence in the paragraph are the most important once.

- **Sentence length -** number of words that are in a sentence. Here, only words that are not stop words are accounted for. Sentences that are longer could have more information, and hence could be more often selected to be in a summary.

- **Level of difference between the title of the text and the sentence -** estimated by calculating the cosine distance. More specifically, both the title and the sentence that is analysed are transformed in to vectors that show the frequency of each word found in both of them (a.k.a. Term Document Matrix). The similarity between the two texts is estimated through the following equation:

$$similarity(t, s_i) = \frac{\sum_{n=1}^{N} t_n s_n}{\sqrt{\sum_{n=1}^{N} t_n^2 \sum_{n=1}^{N} s_n^2}} \tag{2.1}$$

  where t_n and s_n are the components of word frequency vectors of the title and the analysed sentence respectively. It is assumed that the title of the text also represents the main topic. Hence, sentences that talk about the main topic are most likely are the important once.

- **Number of times keywords appear in a sentence -** almost all academic papers at the start of the text have several keywords that represent the document. Hence, it is likely that sentences that have large number of these keywords also carry a lot of information about the text as a whole.

- **Number of times the five most prominent words in the text appear in the sentence -** again, here only words that are not stop words are accounted for. Words that appear most often in the text are likely the once that represent the text the best. Hence, sentences that have a large number of these words are likely to be important.

- **Number of times the three most prominent 2 or 3 word phrases in the text appear in the sentence -** the logic of including this feature is the same as is for the previous feature.

2.4 Creating a training data set

All the aforementioned text summarisation algorithms can be classified as supervised learning methods. This means that in order for them to estimate the parameters that will help to find important sentences, they have to have a data set on which they can be trained. Such data set in data mining is called a training data set. The main feature of this data set is that it has a variable that already classifies each sentence in a paper according to if it should be saved or not.

In the academic literature, most often this data set is created by experts [5]. That is, experts read each document and evaluate each sentence as worthy or not worthy of extraction. Though this approach provides very good results, it is very time consuming and often the summaries depend on the people that write them. In other words, two experts reading the same document often classify different sentences as important as they often understand the text differently. In addition, when highly technical documents are evaluated, as is the case with this master's thesis, it is even more difficult to find people to summarise them all. Hence, in the thesis the summary sentences will be identified using an algorithmic approach.

Sentences that summarise the document well were identified by finding which sentences in the text are closest to sentences in the abstract. In other words, it was assumed that the abstracts of the academic papers, which were written by the authors, are good quality summaries, and the sentences in the text that are most similar to them are the most important. By using this approach the both issues cited previously are mitigated. More specifically, first of all, it is very likely that the author understands his work the best, and hence the abstract includes only information that is crucial. In addition, by using an algorithmic approach there is no need for experts in all four fields of science from which the papers are analysed. Distance between sentences in the full text and abstracts were estimated using cosine distance measure. It was selected as it is heavily used by researchers to compare texts [12]. Table below provides a comparison of an abstract of a paper titled *The Curse of Natural resources*, which is from field of economics with the summary created using the cosine distance measure. As can be observed from the illustration, by using the cosine distance it is possible to find very similar sentences in the full text to the sentences found in the abstract.

Abstract	**Created summary**
This paper summarizes and extends previous research that has shown evidence of a 'curse of natural resourcesa countries with great natural resource wealth tend nevertheless to grow more slowly than resource-poor countries. This result is not easily explained by other variables, or by alternative ways to measure resource abundance. This paper shows that there is little direct evidence that omitted geographical or climate variables explain the curse, or that there is a bias resulting from some other unobserved growth deterrent. Resource-abundant countries tended to be high-price economies and, perhaps as a consequence, these countries tended to miss-out on export-led growth.	*Therefore, one explanation of the resource curse is that resource abundance tended to render the export sectors uncompetitive and that as a consequence resource-abundant countries never successfully pursued export-led growth. It is not easily explained by other variables, or by alternative ways to measure resource abundance. This paper shows that there is little direct evidence that omitted geographical or climate variables explain the curse, or that there is a bias resulting from some other unobserved growth deterrent.We also show evidence that resource-abundant countries tended to be high-price economies and that, partly as a consequence, these countries tended to miss-out on export-led growth.*

Table 1. The first column shows the text found in the abstract of a paper titled *The Curse of Natural resources*, while the second shows the summary created through cosine distance estimates.

In the initial stage of research twenty sentences from the text that are closest to abstracts are classified as important while the rest as unimportant. The number twenty was selected as abstracts generally have half of that many sentences. In addition, because abstracts often carry vary compressed information, while text do not, by using twenty sentences it is possible to accurately convey all the information in the abstracts using sentences from the full texts. By selecting twenty sentences, the final data set for each of the scientific fields will have a total of 400 sentences that are classified as worthy of extraction (i.e., sentences that summarise the texts well), while the rest are classified as unimportant. However, also in order to ensure robustness of the results, the summarisation will be performed several times using different number of sentences each time (more on this see *2.7 Sensitivity analysis*).

2.4.1 Keeping unique sentences

One issue by simply finding sentences that are the closest to abstracts is that some sentences could repeat themselves. In other words, if two sentences in the full text are very similar to one another, but also are very similar to one of the sentences in the abstract, it is possible that these two sentences will be deemed as worthy of extraction. Hence, it could be the case that virtually the same sentence will be extracted as a good summary. In order to ensure that this is not the case, the algorithm also checks if the sentences that are worthy of extraction are relatively different from one another. This is done through a loop that estimates the cosine distance between all extracted sentences and removes all sentences that have a cosine distance with at least one other sentences of 0.8 or larger. In addition, when two sentences are close to one another, the sentences that appears latter in the text is removed, and the one that appears first is retained. This is because almost all academic papers start with an introduction that often summarises some information found in the text. Hence, by keeping the first sentences, this important information is not removed from the analysis.

In addition, by simply removing similar sentences the number of summary sentences could become very small. Because of that, after removing any sentence that was deemed as a summary sentence, a new one is added in its place. Then the process of finding similar sentences is repeated again, until there is no more sentences that are very close to one another. At all time, the number of sentences that are evaluated is twenty, as was defined prior in the master's thesis (see section *2.4 Creating a training data set*). The pseudo code below elaborates on the algorithm in a more technical manner:

Algorithm 2. Removing similar summary sentences

```
$n \Leftarrow$ sentence to extract
order.df $\Leftarrow$ order(cosine.distance)
take.n $\Leftarrow$ $order.df[1:n]$
how.much.to.remove $\Leftarrow 1$

while $how.much.to.remove > 0$ do
    $cos.dist \Leftarrow$ dist$(take.n)$
    $remove.df \Leftarrow$ which(cos.dist > 0.8)
    $remove \Leftarrow NULL$

    if(length$(remove.df) > 0)$ do
    for $i$ $in$ $1:$ length$(remove.df)$ do
        $remove \Leftarrow$ add$(remove, remove.df[$which.max$|(remove.df[i])])$
    end for

    $remove \Leftarrow$ unique$(remove)$
    $n.old \Leftarrow n + 1$
    $how.much.to.remove \Leftarrow$ length$(remove)$
    $n \Leftarrow n + how.much.to.remove$
    $taken.n \Leftarrow (add)(taken.n[-remove], order.df[n.old : n])$
end while
```

First part of the algorithm imports all the necessary information that will be used in the research. More specifically, it takes *n* sentences from the full text that are closest to the abstracts. The second part of the algorithm is a while loop that tries to find if there are any sentences that are very close to one another and removed them as well as adds additional sentences instead of removed ones. More specifically, the *for* loop, finds which sentences are very close to one another and selects the one that appears second in the text, while the end of the algorithm removes the identified sentences, adds new sentences, and prepares for the next while loop (if it is necessary).

2.5 Finding summary sentences

2.5.1 Balancing the data and creating the training and testing data set

In order to improve the quality of the summaries the data set that is analysed was balanced according to sentence classification. In other words, the data set that is summarised has a relatively equal amount of sentences that are deemed as worthy and those that are deemed as irrelevant. The balancing was performed by randomly removing sentences that are deemed as unimportant until the number of unimportant sentences was relatively similar to the number of important once. This was done as several academics stipulate that the quality of data mining models improves drastically when a balance data set is used [9]. Some might argue that this could make the results of research lose their robustness. Hence, to mitigate this possibility, the summarisation of texts was performed not once but 1000 times. More on this see subsection *3.1Experiment*. In addition, in order to further check the robustness additional sensitivity analysis was performed where unbalanced data was used (for more on this see *2.7 Sensitivity analysis*).

In addition, before the training of the algorithm the data was split in to two data sets, testing and training. The training data set includes 60% of the balanced data, while the testing includes 40%. The following pseudocode provides a more technical explanation on how the data set was first of all balanced and then split in to a training and testing data sets:

Algorithm 3. Balancing and dividing the data in to a training and testing data set

```
$df \Leftarrow .csv\ files\ of\ a\ paper\ created\ after\ theprocessing\ step$
$df.true \Leftarrow df[df\$Should.be.saved == "Yes",]$
$df.false \Leftarrow df[df\$Should.be.saved == "No",]$

for $i\ in$ $1 : 1000$ do
    $random.sample.true \Leftarrow df[df\$Should.be.saved == "Yes",]$
    $random.sample.false \Leftarrow df[df\$Should.be.saved == "Yes",]$
    $sample.df.false \Leftarrow$ sample$(c(0, 1),$
                                        prob $= c($nrow$(df.true)/$nrow$(df.false),$
                                        $1 -$ nrow$(df.true)/$nrow$(df.false)))$ {Balancing the data}
    $df.to.split \Leftarrow$ rbind$(df.true, sample.df.false == 0)$
    $split \Leftarrow sample(c(0, 1),$ nrow$(df.to.split),$ prob $= c(.6, .4)$
    $df.train \Leftarrow df.to.split[split == 1,]$
    $df.test \Leftarrow df.to.split[split == 0,]$
    Find summary sentences {Algorithm of this step is not provided here}
    Save the accuracy {Algorithm of this step is not provided here}
end for
```

Several aspects of the code should be mentioned. First, as can be seen from the code, sentences that are worth extracting and those that are not are processed separately. This is done in order to create a relatively balanced data set. However, after processing both data set they are merged in to one data set that is later split in to training and testing data.

Second, by performing all the steps defined in the pseudocode the first sentences that appear in both *df.train* and *df.test* are those that worth extracting. Only after all sentences that are worth extracting appear, sentences that are not worth start appearing. In other words, the sequential nature of the documents is completely lost. On one hand, this is not an issue for majority of algorithms, as they use features and do not care about the sequence of sentences. On the other hand, for Hidden Markov Model this is a huge issue as it relies on the sequential nature of a text. Hence, Hidden Markov Model uses a different way to create a training and testing data set. In it, the training data set includes all-but-two of all documents, while the testing data the remaining two documents. This approach is a variation of a frequently used cross-validation method of classification algorithms called leave-one-out. Using this approach the sequential nature of the texts is not violated, as the training is done using full texts which have sentences appear in a correct sequence.

However, and argument can be made that because Hidden Markov Model uses a different approach to create training and testing datasets, this model is not comparable to others. Hence, during sensitivity analysis, the other classification algorithms will be also divided in to a training and testing data sets using the HMM approach in order to check if this approach influences the results of summarisation.

2.5.2 Using algorithms

The extraction of sentences that are good summaries was performed using the algorithms described prior in *1. Overview of text summarisation*. All the algorithms were realised in R using the authors own script and the following R packages: party, e1071, and mhsmm. It was decided to use pre-existing packages rather than build algorithms from scratch as the main aim of this study is not to improve existing algorithms, but to explore how different data effects the algorithms. In addition, by using pre-existing packages it can be assumed that they are optimised and hence bad code does not influence the final results. All the algorithms were used as they are (i.e., without any pruning). Again, this was done in order to make the study as unbiased as possible. However, this does not imply that there was no training involved. Each algorithm was trained using the training data set described prior and tested on the testing data set.

In addition, though majority of algorithms use similar data (i.e., features described prior) and work similarly, there are some nuances that have to be addressed. First, pivoted QR decomposition, unlike other models, is not considered as a standard classification method. This is because unlike other classification methods, it does not require any kind of training to select which sentences are summary and which are not. More specifically, it finds relevant sentences by finding which have the largest number of ideas. However, in this paper the pivoted QR decomposition had pseudo training in a form of finding the relevant ideas. As was mentioned prior, in order to account for ideas that are relevant to the text, only words that appear in the abstracts were used. However, even with this pseudo-training this model most likely will perform much worse than other models. Hence, in this paper this model is used more as a baseline. In order for other summarisation models to be considered viable, they have to outperform the pivoted QR decomposition algorithm.

Second, as was mentioned prior Hidden Markov Model, unlike other models, when selecting summaries takes in to account if the previous sentence was selected as a summary. Because of this the same kind of data used with Naïve Bayes, Decision Tree, SVM, and Pivoted QR Decomposition cannot be applied to HMM. More specifically, as was mentioned prior in order to improve the results of the summary the data was balanced. During the balancing process some sentences were removed from the analysis. For other models this is not a problem, but for HMM this means that the integrity of a document is ruined (i.e., it cannot always take in to account if the previous sentence was a summary as sometimes the previous sentence is missing). To mitigate this problem, for HMM model full texts were used in the analysis. More specifically, the training dataset was created using 90% of all texts, while the testing was one on the 10% that was left (i.e., in each iteration 18 texts will be used while two remaining texts will be used in testing). In addition, in order to allow for a better comparison between algorithms the accuracy was estimated by including all the texts in a training and testing datasets at least ones. Even so, the aforementioned limitations have to be kept in mind during the analysis, as because of the aforementioned limitations, it is very likely that HMM will perform worse than other algorithms. Though, the results created by it can be still used, as they will allow to see if HMM performs different with different types of texts.

2.6 Calculating accuracy

Quality of the summaries created by the algorithm were assessed employing the accuracy measure known as area under the ROC (Receiver Operating Characteristic) curve. The ROC curve that evaluates how well a binary classification algorithm works at different thresholds. More specifically, the curve shows the true positive and false positive rate of the classifier at different cut-off points. Area under the ROC curve (AUC) was chosen as a measure of accuracy as unlike other measures, such as sensitivity/specificity, out-of-bag error rate, etc., it estimates accuracy not with a point but with an area under a curve, making it more robust than the alternatives [15]. AUC was calculated employing the ROCR package in R. In addition, after estimating the average AUC the best ROC cure (i.e., the one with the largest AUC) was plotted for each algorithm in order to allow for more visual comparison between different algorithms. However, the ROC curve will not be plotted for HMM, as the results of this model are too volatile and the best result can simply represent random luck (more on this see 3.1 *Experiment*).

In addition, to provide robustness to the quality check, the AUCs created by different classification models were compared statistically through ANOVA (analysis of variance). ANOVA is a statistical model that compares if two or more groups differ from one another. The comparison is done by comparing the means of the groups. Hence, ANOVA is often considered as a generalised form of a t-test. However, one of the ANOVA's drawbacks is that as the number of groups grows the probability that it will be statistically significant exponentially grows as well. Hence, to ensure that any significance uncovered is due to groups being different and not because of the number of the groups ANOVA will be complimented by the Tukey's range test. Tukey's range test is often used together with ANOVA to find which groups differ (statistically) from one another and which do not. By combining the two statistical models with AUC, it will be possible to state if the differences observed between different classification models are statistically significant.

2.7 Sensitivity analysis

In order to ensure the robustness of the end results a sensitivity analysis will be performed. First of all, during the sensitivity analysis, different number of summary sentences will be tried out. At the initial step 20 sentences in the full text are classified as summaries. However, during sensitivity analysis this number will range from five to forty sentences. Five was selected as there are several texts with this number of sentences in their abstracts. While forty was selected in order to see what will happen with the algorithms at extreme values. Second, in the initial stage, as was describe, the data is balanced in order to improve the quality of summaries. Hence, in the second stage of sensitivity analysis the data will not be balanced in order to see if this will have an influence on the results. Third, at the initial stage the training and testing data sets are created by selecting sentences from texts at random. Hence, the sensitivity will also be checked by creating the training data set using 90% of the texts (not sentences) and testing it on the rest 10%. In other words, the algorithms will be tuned using sentences from one group of documents, while the testing will be performed on texts from completely different group of documents. In addition, to ensure consistency of results and robustness of the research, after changing one of the parameters of the model the model will be iterated 1000 times and the average will be presented in the end results.

3 Results of the research

3.1 Experiment

The hypothesis that different text summarisation models perform differently with documents from different academic disciplines was tested by comparing the AUCs and ROC curves. More specifically, the experiment was carried out using the following steps:

1. For all the sets of academic papers a distinct data set was created, where each row was a sentence from a document and the data sets had the following variables: (i) document name from which the sentence was extracted, (ii) original sentence, (iii) cleaned sentence, (iv) classification of the sentence according to its importance, (v) sentence position, (vi) sentence length, (vii) level of difference between the title of the document and the sentence, (vii) number of times keywords appear in a sentence, (ix) number of times the five most prominent words in the text appear in the sentence and (x) Number of times the three most prominent 2 or 3 word phrases in the text appear in the sentence.

2. All datasets were split in to a training and testing dataset randomly. The training dataset had 60% of all observations, while the testing data set had 40%. Here the data sets that were split were balanced data set discussed previously (see section *2.5 Finding summary sentences*).

3. Pivoted QR Decomposition, Naïve Bayes, Decision Tree, Hidden Markov Model, and Support Vector Machine text summarisation algorithms were ran on the training data sets in order to create the text summarisation models.

4. Models were tested by applying them on the testing data set and calculating the AUC of the results.

5. In order to add robustness to the end results, steps two through four were repeated 1000 times (190 for HMM), each time saving the AUC of the results in a separate file.

6. At each iteration if the new AUC was higher than any previous one, the statistics were saved in order to later create ROC curves.

7. The AUCs of 1000 iterations (190 times for HMM) were saved in to a separate file and compared in the next section.

8. The statistics of best AUCs was saved and from them ROC curves were created

3.2 Results

This subsection describes how the accuracy of text summarisation changes by using different algorithms with academic papers from different scientific disciplines.

3.2.1 Medicine

According to the table 1 below, the best algorithm for Medical papers is Naïve Bayes. Close second is SVM, while others are trailing behind. However, even though other algorithms did not perform as well, it is interesting to note that all of them performed better than Pivoted QR Decomposition (the base case). In addition, even though HMM used unbalanced dataset, it still managed to have an AUC that is around 10% higher than Pivoted QR Decomposition. Even more interesting, if we look at the max AUC for all algorithms HMM managed to outperform others by at least 4%. However, because HMM also has the largest standard deviation, this results can simply be attributed to randomness.

	Naïve Bayes	**Decision Tree**	**SVM**	**Pivoted QR**	**HMM**
Mean AUC	0.767	0.742	0.765	0.552	0.651
Max AUC	0.826	0.827	0.861	0.611	0.907
Min AUC	0.594	0.561	0.621	0.030	0.502
Standard deviation	0.032	0.026	0.030	0.030	0.101

Table 2. Area under the ROC curve for different models and academic papers from different fields. Results demonstrate that Naive Bayes outperformed SVM, but only by a very small margin.

According to ANOVA and Tukey's rank test (a.k.a. Tukey's honest significance test), the differences between all of the algorithms are statistically significant at *p*=0.01.

The ROC curve below demonstrates that generally there is no huge difference between different algorithms when applying them on the papers in the field of Medicine. This is true for SVM, Naive Bayes, and Decision Tree algorithms. In addition, as can be observed the ROC curve of the Decision Tree algorithm has small number of angles, which implies that the best decision tree is quite short. In other words, this algorithm uses only several features when creating classification rules. In addition, HMM is omitted from the plot as can be observed in the table below, it has a very huge variation (3 times larger than any other model) and also the AUC for HMM varied from 0.502 to 0.907, which is huge. Hence, as the AUC of this model varied too much, it is unwise to compare it to other models as it is likely that the best HMM result was due to randomness.

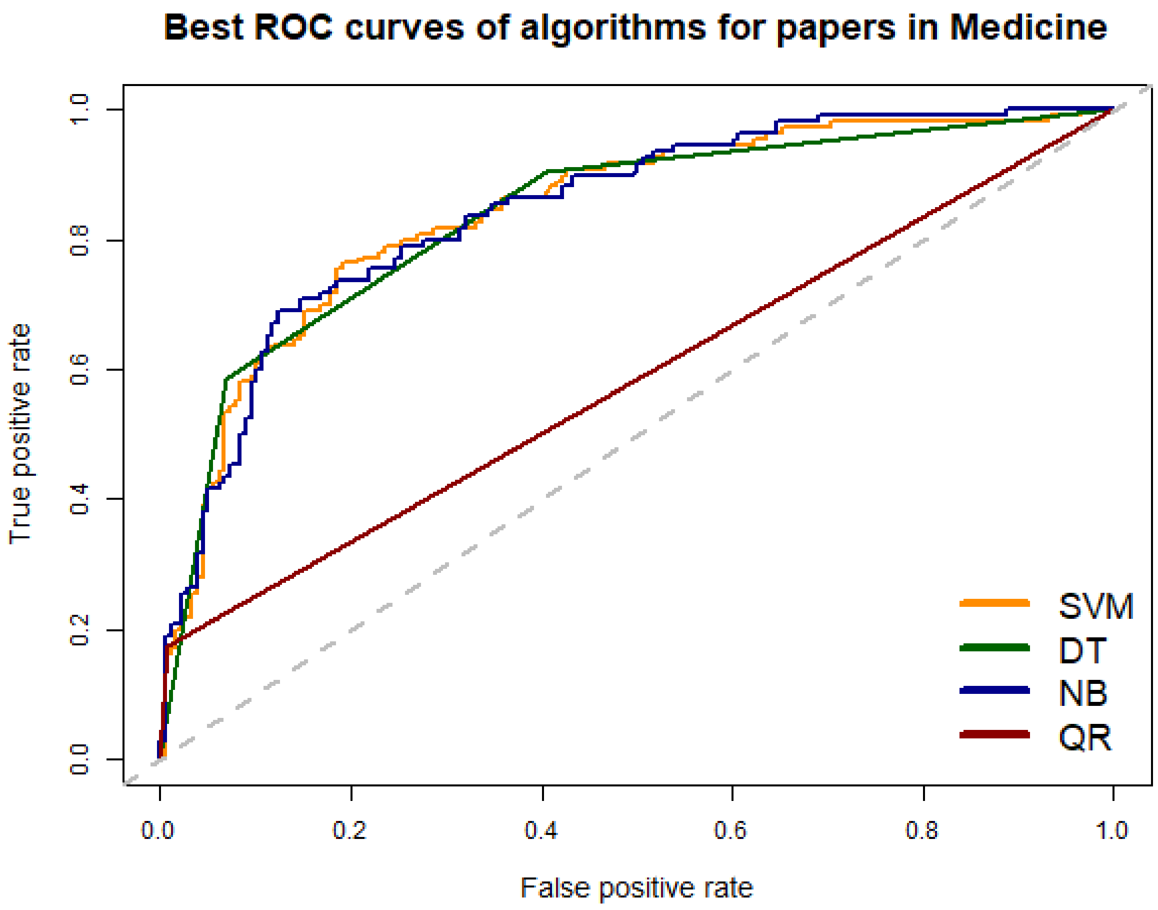


Figure 5. The graph presents the ROC curve of the iterations of each algorithm that performed the best. In other words, the plot above shows the best result that each algorithm managed to achieve through 1000 iterations. It shows that there is no major difference between Decision Tree, Support Vectors Machine, and Naive Bayes. However, the result here should be interpreted carefully, as each algorithm achieved their best result with different training and testing dataset. Even so, the ROC curve here presents some interesting insides about which algorithm is the best for papers in the field of Medicine.

3.2.2 Biology

Summarisation of papers in biology provides very similar results to the medical papers. SVM algorithm performs better than Decision Tree or Naïve Bayes. However, here SVM outperformed Naïve Bayes algorithm, by a very small margin (less than 1%). This implies that both of these algorithms summarise the text similarly. In addition, this is the second time already that Decision Tree algorithm was outperformed both by Naïve Bayes and SVM. This is interesting as Decision Tree is often considered a more robust algorithm than Naïve Bayes. Another interesting observation is that majority of the algorithms performed better with Biology papers than Medical papers. Though the difference is not huge between the two. This result is interesting as papers in Biology are less focused than those in the fields of Medicine and hence should be more difficult to summarise. However, it seems this fact does not influence the accuracy of summarisation in any major way.

The difference between results is statistically significant at p=0.01 according to both ANOVA

	Naïve Bayes	**Decision Tree**	**SVM**	**Pivoted QR**	**HMM**
Mean AUC	0.781	0.749	0.784	0.552	0.676
Max AUC	0.851	0.824	0.862	0.581	0.858
Min AUC	0.702	0.658	0.711	0.499	0.501
Standard deviation	0.023	0.027	0.025	0.013	0.107

Table 3. According to the descriptive statistics of the AUC, here SVM algorithm performs better than the alternatives. However, it outperformed Naive Bayes only by a very small margin.

and Tukey's rank test.

The ROC curve below supports the results in the table. Namely that there is no huge difference between the algorithms. Even though SVM in majority of cases outperforms others, the difference is not huge. It also has to be mentioned that with small cut-off points Naïve Bayes outperforms SVM by quite a margin, though this good performance does not last long.

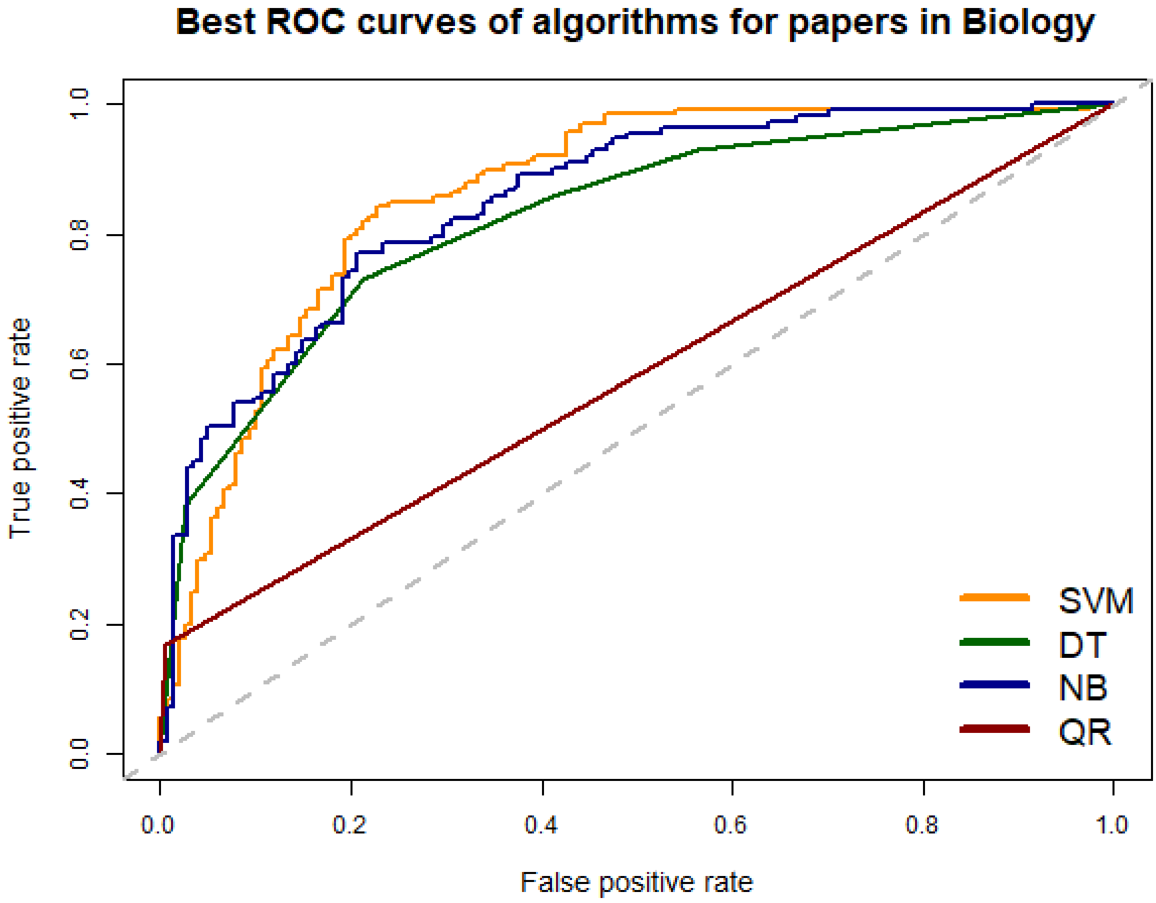


Figure 6. Similarly to the ROC curve for the papers in the field of Medicine, this plot shows that algorithms in general perform very similarly. However, here it can be observed that SVM in several instances outperforms others by a small margin.

3.2.3 Economics (including Finance)

Here the difference between using the Naïve Bayes or the SVM algorithm is less than 1%. In addition, for papers in the field of Economics (including Finance) decision tree algorithm performs almost as well as SVM and Naïve Bayes (the difference between them is less than 2%). Finally, yet again the HMM maximum AUC outperformed other algorithms. Though this result is most likely due to randomness, as was stated prior, it is still an interesting observation.

	Naïve Bayes	**Decision Tree**	**SVM**	**Pivoted QR**	**HMM**
Mean AUC	0.799	0.785	0.804	0.547	0.660
Max AUC	0.844	0.837	0.843	0.584	0.900
Min AUC	0.677	0.649	0.662	0.511	0.488
Standard deviation	0.026	0.028	0.027	0.012	0.119

Table 4. Results show that papers from the field of economics are the easiest to summarise. This can be attributed to the fact that these papers often have smaller amount of formulas and calculations and this discipline is less exact. In addition, here SVM outperforms all other methods, though only by a small margin (around 0.5%).

The differences between algorithms in the table are statistically significant at p=0.01 according to both statistical tests.

Similarly to the papers in the field of Medicine, there is no huge difference between algorithms. More specifically, in majority of cases SVM outperformed other algorithms, but the difference between them was very minor.

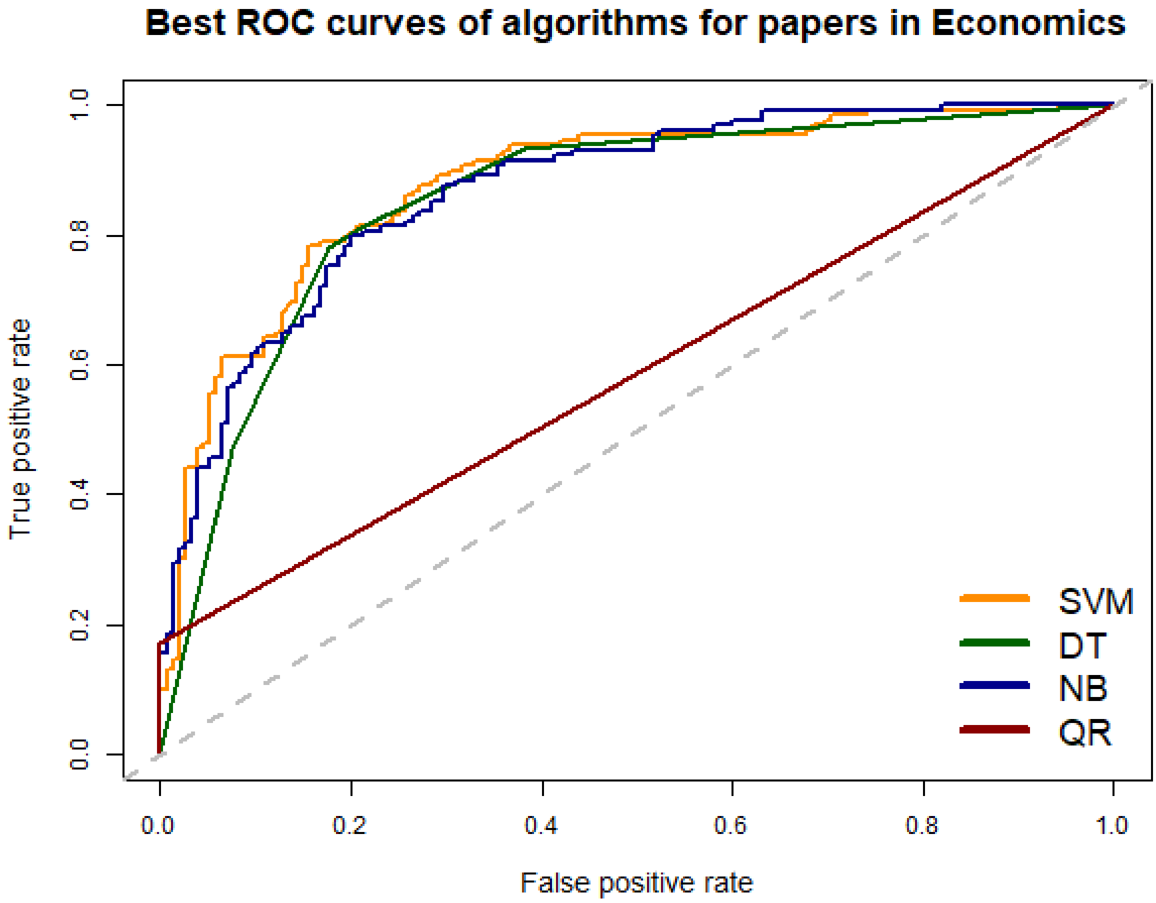


Figure 7. The ROC curve demonstrate that there is only minor differences between algorithms, except for Pivoted QR decomposition.

3.2.4 Computer science

According to the results (see table below) algorithms have the most difficulty summarising computer science academic papers. Though on average the difference between these types of papers and others is not huge (from 2% to 6%). Here, similarly to papers in Medicine, Naïve Bayes managed to outperform SVM and similarly the difference between it and SVM is relatively small.

	Naïve Bayes	Decision Tree	SVM	Pivoted QR	HMM
Mean AUC	0.749	0.709	0.746	0.531	0.653
Max AUC	0.823	0.783	0.845	0.570	0.812
Min AUC	0.653	0.621	0.645	0.483	0.452
Standard deviation	0.025	0.028	0.027	0.013	0.085

Table 5. It is clear that the algorithm has the largest difficulty with computer science papers. This can be seen by looking at the average AUC, which is smaller than for other kinds of papers.

Results are statistically significant at $p = 0.01$ with both statistical tests.

Similarly to others, algorithms performed relatively similar, though SVM in majority of cases outperformed other algorithms. In addition, similarly to the papers in the field of Medicine, in several cases (i.e., at several cut-off points) Naive Bayes outperformed SVM. However, there number of such cases was relatively small.

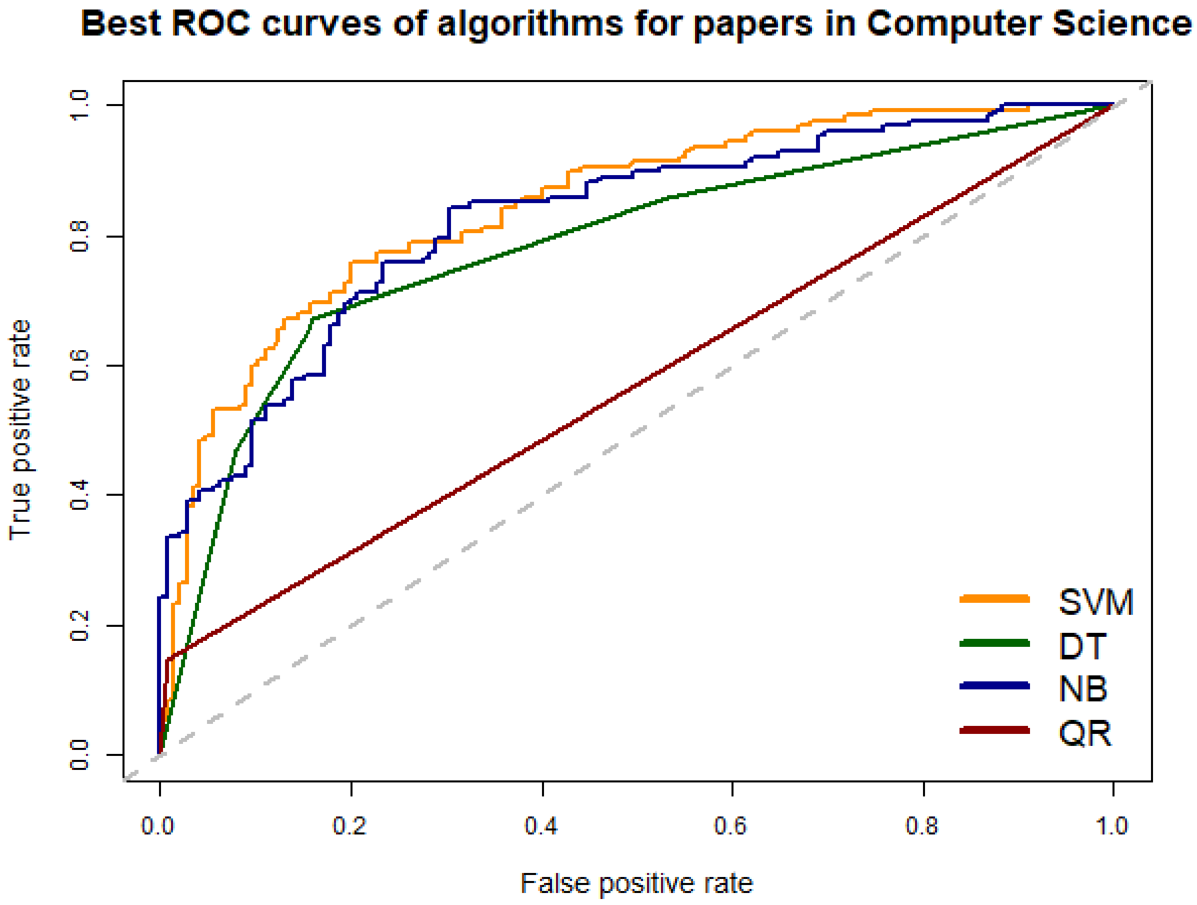


Figure 8. ROC curve shows that majority of algorithms perform relatively the same.

3.2.5 Aggregated data

In addition, for further research all the texts were aggregated in to one data set in order to see if better result could be achieved. According to the table below, the answer is not clear cut. First of all, there is no best algorithm for this type of data. SVM, Naïve Bayes, and Decision Tree in general performed very similarly. Interesting enough that here decision tree managed to achieve similar performance as other algorithms, which implies that it is more dependent on the sample size than others. Second, aggregated data managed to get higher AUC than papers in the field of Computer Science, it had a similar average AUC as with papers in the field of Medicine, though algorithms performed a bit better here, and it performed worse with aggregated data than with using papers in the field of economics and biology separately. Keeping in mind that aggregated data was four times larger than any other dataset used, the results achieved here are surprising. However, they fall in line with the hypothesis that states that the type of text matters in summarisation. In other words, it implies that to achieve better results it is much better to summarise different text separately rather than group them together.

	Naïve Bayes	Decision Tree	SVM	Pivoted QR	HMM
Mean AUC	0.773	0.771	0.773	0.511	0.669
Max AUC	0.811	0.807	0.812	0.520	.887
Min AUC	0.733	0.717	0.733	0.501	0.473
Standard deviation	0.012	0.012	0.013	0.003	0.104

Table 6. According to the table, aggregated data only performed clearly better than Computer Science data separately.

The ROC curve below supports the conclusions achieved prior. That is, there is no huge difference in performance between Decision Tree, Naive Bayes, and Support Vector Machine.

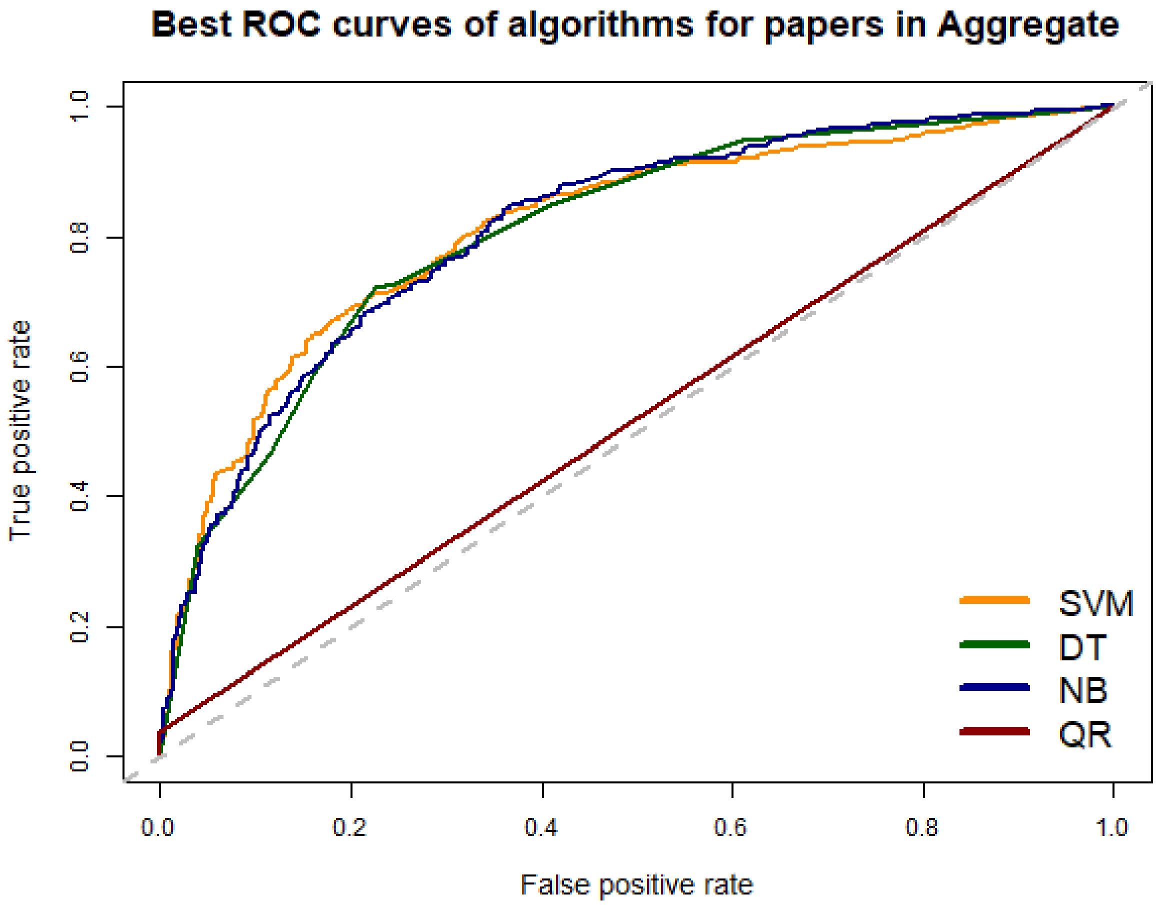


Figure 9. The figure above shows that there is no major difference between the aforementioned algorithms, though Pivoted QR decomposition here performed worse than with other types of papers.

Currently looking at the results it can be speculated that the type of the text does have an impact on the quality of summaries, though it is low. In addition, it does not seem that one algorithm perform better with one type of document, while others with other types of documents. However, in order to further support or disprove this claim, sensitivity analysis was performed.

3.3 Results of sensitivity analysis

This section provides results of the sensitivity analysis described in *2.7 Sensitivity analysis*. More specifically, this section elaborates on how would the average AUC changes if some of the parameters of the algorithm would be changed. First parameter that is changed is the number of sentences that are deemed as summary sentences. In the original algorithm, this number was set to twenty sentences, while in the sensitivity analysis it ranges from five to forty. Second, the original data set used balanced data in order to improve the accuracy of the algorithm. Hence, second sensitivity analysis looks at what happens with the AUC if the data is unbalanced. Finally, the model is checked further by selecting the training and testing dataset not at random, but by selecting some documents to serve as trainers, while others as testers (i.e., similarly how HMM was trained prior). In addition, to add robustness, the second and third sensitivity tests are combined in order to see how the result change if both unbalanced data and non-random training/testing data s1ets are used together.

3.3.1 Diffrent number of summary sentences

In order to test how the results would change if a different number of sentences would be deemed as important, the 1000 iterations of the algorithm were ran again using 5, 10, 20, 30, and 40 sentences as summary sentences. The table below provides the average AUC of the 1000 iterations of the algorithm for each document type. As can be observed from the table by changing the number of sentences the results do not change drastically. In majority of cases Naïve Bayes outperforms other algorithms, though SVM often reaches the same level of accuracy as the number of sentences increases. In addition, Decision Tree, similarly to SVM, improves in accuracy as the number of sentences grows, though it never reaches the same results as Naïve Bayes or SVM. Pivoted QR Decomposition, in general, performs as well with small number of sentences as it does with a large number. HMM independent on the number of sentences used did not manage to reach the same level of accuracy as other algorithms. However, as other its accuracy (measured via average AUC) increases as the number of sentences grows.

Papers	# of papers	Naïve Bayes	Decision Tree	SVM	Pivoted QR	HMM
Medicine	**5**	0.789	0.721	0.780	0.551	0.648
	10	0.787	0.738	0.798	0.553	0.645
	20	0.767	0.742	0.765	0.553	0.651
	30	0.777	0.755	0.773	0.551	0.655
	40	0.780	0.765	0.780	0.550	0.679
Biology	**5**	0.784	0.679	0.763	0.541	0.625
	10	0.782	0.709	0.771	0.541	0.645
	20	0.781	0.749	0.784	0.542	0.676
	30	0.788	0.760	0.783	0.538	0.672
	40	0.802	0.783	0.795	0.536	0.695
Economics	**5**	0.767	0.684	0.756	0.551	0.598
	10	0.799	0.758	0.795	0.547	0.650
	20	0.799	0.785	0.804	0.547	0.659
	30	0.801	0.789	0.800	0.545	0.674
	40	0.744	0.720	0.738	0.532	0.682
CS	**5**	0.754	0.660	0.724	0.532	0.638
	10	0.743	0.668	0.724	0.528	0.644
	20	0.749	0.709	0.748	0.531	0.653
	30	0.748	0.718	0.743	0.530	0.670
	40	0.802	0.797	0.803	0.544	0.515

Table 7. This table elaborates on what happens with the average AUC when different number of sentences is used as summary sentences. It can be seen that by changing the number of sentences the results do not change drastically, though the accuracy of the algorithms tend to improve when a larger number of sentences is used.

3.3.2 Unbalanced data

As can be seen from the table below (table 8), the average AUC of unbalanced data is almost the same as for balanced dataset. However, this result is not surprising. Because the data set is unbalanced, algorithm can simply predict that every sentence will be a bad summary sentence and have an accuracy level of 95% or even higher. Figure 10 below, which is a random confusion matrix created by using the decision tree algorithm and unbalanced data, show this idea in more detail. Namely, as can be observed, there is much more sentences that are non-summary than summary in the data frame. In addition, as can be observed SVM performed much worse than other algorithms in all cases. This is because it is much more complex than other algorithms, it tries to predict much more often that a sentence will be a summary sentence, while other algorithms simply almost always predict that every sentence is a bad summary. What is more, HMM here performed the same as prior as it originally used unbalanced data. However, even after equalizing the playing field between algorithms, HMM performed worse than other algorithms. Finally, by using unbalanced data the results of the research are almost the same. That is, Naive Bayes is the best algorithm and it performed the best with papers from the field of Economics and the worst with papers from the field of Computer Science. The only major difference that is observed by using the unbalanced data is that here the Decision Tree algorithm almost always performed just as well as Naive Bayes.

	Naïve Bayes	**Decision Tree**	**SVM**	**Pivoted QR**	**HMM**
Medicine	0.766	0.754	0.650	0.526	0.648
Biology	0.783	0.773	0.649	0.520	0.676
Economics	0.802	0.801	0.592	0.516	0.659
CS	0.750	0.726	0.611	0.505	0.653

Table 8. Table shows that the accuracy of classification of unbalanced data set is very similar to balanced. However, this is due to the fact that a very large number of observations are classified as non-summary sentences.

		Reality	
		Yes	**No**
Prediction	**Yes**	16	13
	No	133	948

Figure 10. The confusion matrix demonstrates that even though algorithms had high AUC, this is due to the fact that they almost always predicted that a sentences is a bad summary sentence. The accuracy of this random confusion matrix created during one of the iterations was 86.8%.

3.3.3 Data from diffrent documents

By using different documents rather than different text the results do not change much, as can be seen in the table below. As was previously, the best algorithm is Naïve Bayes, while the worst is Pivoted QR Decomposition. However, similarly to unbalanced data, here decision tree algorithm has comparable accuracy to the Naïve Bayes approach. In addition, similarly here SVM does not reach the same level of accuracy as it does with balanced data set. What is more, as was previously, the economic papers are the ones that are easiest to summarise, while papers in the field of computer science are the most difficult. Finally, by looking at the situation with balanced and unbalanced data, the accuracy does not change that much. However, as was elaborated prior, this does not mean that unbalanced data set performs just as well. They simply classify more sentences as non-summary sentences and do more mistakes while finding which sentences should be considered as summary sentences. In other words, the ease of randomly picking non-summary sentence balances out the difficulty that these algorithms have when trying to find a real summary sentence.

		Naïve Bayes	Decision Tree	SVM	Pivoted QR	HMM
Balanced	**Medicine**	0.771	0.761	0.671	0.549	0.648
	Biology	0.785	0.770	0.701	0.528	0.676
	Economics	0.801	0.797	0.731	0.528	0.659
	CS	0.754	0.712	0.673	0.532	0.653
Unbalanced	**Medicine**	0.772	0.762	0.632	0.537	0.648
	Biology	0.785	0.767	0.632	0.517	0.676
	Economics	0.803	0.801	0.577	0.512	0.659
	CS	0.755	0.713	0.567	0.520	0.653

Table 9. Table shows that the accuracy of classification of unbalanced data set is very similar to balanced. However, this is due to the fact that a very large number of observations are classified as non-summary sentences.

3.4 Result conclusions

The results in the research allow to stipulate two things. First of all, one type of algorithm does not perform better with one type of document, while another with another. From the research it can be observed that in most cases the same two algorithms (Naïve Bayes and Support Vector Machine) were the algorithms that performed the best with all types of text. Though in some cases on of the two performed better while in others another one performed better, the difference between them was too small to reliably state that such a difference exists. Hence, the research did not find any support for the first hypothesis. In addition, this also means that the approach of evaluating the quality of new and improved text summarisation algorithms by applying them to one type of texts is sound. However, one aspect has to be mentioned, that by using unbalanced data set some algorithms tart to perform much worse than others (e.g., SVM). Hence, even if there is no support for the first hypothesis, it is wise to check the algorithm with both balanced and unbalanced datasets in order to make sure that this aspect of the data does not have an adverse effect on the algorithms.

Second, in general, text summarisation algorithms perform better with texts of one type than with text of aggregated types. As was observed, in most cases algorithms performed better while using documents of different types rather than using the aggregated documents. Though this was not the case always, keeping in mind that the aggregated data was four times larger than any other data set, and that larger data sets tend to produce results with better accuracies, it can be concluded that it is better to use separate data sets. This finding does not really influence the way the algorithms are tested, as the best algorithms in separate data sets were also the best in the aggregated case. However, from the application perspective, this finding is important, as it is possible to build a better algorithm by separating the documents by the type.

Finally, the approach of testing text summarisation results provided in this master thesis serves as a good guideline on how algorithms could be checked. In other words, though it was not found that algorithms perform much differently with different text types, this finding might not apply to all types of documents. For example, if the algorithms would be applied on books from different genres maybe the results would be different. Hence, some researchers might find it useful to adapt the methodology defined in this thesis to check the quality of their algorithms and to add robustness to their end results.

Conclusions and Recommendations

There are several findings from the research. First, it seems that the type of text does not influence the quality of the research. In general, Naïve Bayes and SVM performed the best with all types of papers. Though in some cases, one outperformed the other, the difference between them was very minor. This observation is also supported by sensitivity analysis, where the summarisation models were changed by changing the number of summary sentences, changing the balancing of the data, and changing how training and testing data sets are created. More specifically, in sensitivity analysis, the same algorithm (in this case Naïve Bayes) was the best with all documents, though decision tree algorithm came as close second.

Second, it seems that it is better to separate documents by type rather than to aggregate them before performing summarisation. This is because, as was observed in the results, by aggregating the data, in majority of cases the summarisation quality dropped. More specifically, for papers in the field of Biology and Economics, the accuracy drastically dropped when aggregating the data, for papers in the field of Medicine the quality stayed relatively the same, and for paper in the field of Computer Science, the aggregated data performed better. This, taken together with the fact that the aggregated data was four times larger than any data set, it can be concluded that there are some nuances in the different types of text that the algorithms cannot catch when using the aggregated data set.

Third, though the research did not manage to find a link between the type of text and the quality of summarisation, it still provided useful guidelines how different summarisation algorithms could be evaluated. More specifically, currently in the academic literature algorithms are simply evaluated by applying them to one type of data and this approach is sound as was shown by the research. However, it has to be kept in mind that the research was only applied to a relatively limited number of academic papers from relatively limited number of disciplines. Hence, it could be the case that the conclusions reached in the paper do not apply to all types of documents. Because of this, any researcher that wishes to evaluate any algorithm should take in to account that the type of text might have an impact and should perform a thorough analysis of his/her algorithm by applying it to different texts as well as performing sensitivity analysis.

For future work it is recommended to further test the hypotheses described in this master's thesis by expanding the number of algorithms and expanding the dataset in both scope and type. In addition, further research could also try to build on this master's thesis by creating an algorithm that finds which algorithms performs the best with what kind of data. More specifically, though the research did not show that the type of text has a large impact on summaries, it managed to show that by changing the data some algorithms that performed very well prior tend to lag behind (e.g., SVM when unbalanced dataset was used). Finally, further research could also try to expand the number of features as well as try different approaches in identifying which sentences in the text can be considered as good summaries. For example, by employing expert opinion.

References

[1] Sarker A. Extractive summarization of medical documents using domain knowledge and corpus statistics. *Australasian Medical Journal*, 5(9):478–481, 2012.

[2] Baharudin B., Lee L. H., and K. Khan. A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology*, 1(1), 2010.

[3] Hachey B. and Grover C. Extractive summarisation of legal texts. *Artificial Intelligence and Law*, 14(4):305–345, 2007.

[4] Das D. and Martins A. F. A survey on automatic text summarization. *NA*, 2008.

[5] Hovy E. and Lin C. Y. Automated text summarization in summarist. *In Advances in Automatic Text Summarizationl*, 1, 1991.

[6] Lloret E. and Palomar M. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2011.

[7] Hovy E. H. Automated text summarization. in r. mitkov (ed), the oxford handbook of computational linguistics, chapter 32, pages 583–598. oxford university press, 2005.

[8] Luhn H.P. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[9] Stefanowski J. and Wilk S. Selective pre-processing of imbalanced data for improving classification performance. *Data Warehousing and Knowledge Discovery Lecture Notes in Computer Science*, pages 283–292, 2008.

[10] Nguyen M. L., Shimazu A., Horiguchi S., Ho B. T., and Fukushi M. Probabilistic sentence reduction using support vector machines. *Proceedings of the 20th international conference on Computational Linguistics*, 2004.

[11] Suanmali L., N. Salim, and Binwahlan M. S. Fuzzy logic based method for improving text summarization. *(IJCSIS) International Journal of Computer Science and Information Security*, 2(1), 2009.

[12] B. Li and Han L. Distance weighted cosine similarity measure for text classification. *International Conference on Intelligent Data Engineering and Automated Learning*, pages 611–61, 2013.

[13] Conroy J. M. and O'leary D. P. Text summarization via hidden markov models. *SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–467, 2001.

[14] Munot N. and Govilkar S. S. Comparative study of text summarization methods. *International Journal of Computer Applications*, 102(12):33–37, 2014.

[15] Bradley A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognitio*, 30(7):1145–1159, 1997.

[16] Alguliev R. Evolutionary algorithm for extractive text summarization. *Intelligent Information Management*, 1(2):128–138, 2009.

[17] Mishra R., Bian J., Fiszman M., Weir C. R., Jonnalagadda S., Mostafa J., and Fiol G. D. Text summarization in the biomedical domain: A systematic review of recent research. *Journal of Biomedical Informatics*, 52:457–467, 2014.

[18] Radev Dragomir R., Weiguo Fan, and Zhu Zhang. Webinessence: A personalized web-based multidocument summarization and recommendation systems. *In NAACL Workshop on Automatic Summarization. Pittsburgh, PA.t*, 2001.

[19] Dragomir R. Radev, Timothy Allison, Sasha BlairGoldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Jahna Otterbacher Danyu Liu, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, , and Zhu Zhang. Mead: A platform for multidocument multilingual text summarization. *In LREC, Lisbon, Portugal, May.*, 2004.

[20] Babar S. and Patil P. D. Improving performance of text summarization. *Procedia Computer Science*, 46:354–363, 2015.

[21] Harabagiu S. and Lacatusu F. Generating single and multi-document summaries with gistexter. in workshop on text summarization (in conjunction with the acl 2002 and including the darpa/nist sponsored duc 2002 meeting on text summarization) philadelphia, pennsylvania, usa, 2002.

[22] Kim S., Han K., H. Rim, and Myaeng S. H. Some effective techniques for naive bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1457–1466, 2006.

[23] Malhotra S. and Dixit A. An effective approach for news article summarization. *International Journal of Computer Applications*, 76(16):5–10, 2009.

[24] Hirao T., Isozaki H., Maeda E., and Matsumoto Y. Extracting important sentences with support vector machines. *Proceedings of the 19th international conference on Computational linguistics*, 2002.

Appendix

Papers in the field of medicine

[1] Adams, P. D., Afonine, P. V., Bunkóczi, G., Chen, V. B., Davis, I. W., Echols, N., . . . Zwart, P. H. (2012). PHENIX: a comprehensive Python-based system for macromolecular structure solution. International Tables for Crystallography, 539-547. doi:10.1107/97809553602060000865

[2] Chawla, A. (2012). Obesity is associated with macrophage accumulation in adipose tissue. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1016912.792403062

[3] Chen, V. B., Arendall, W. B., Headd, J. J., Keedy, D. A., Immormino, R. M., Kapral, G. J., . . . Richardson, D. C. (2012). MolProbity: all-atom structure validation for macromolecular crystallography. International Tables for Crystallography, 694-701. doi:10.1107/97809553602060000884

[4] Cruz-Jentoft, A. J., Baeyens, J. P., Bauer, J. M., Boirie, Y., Cederholm, T., Landi, F., . . . Zamboni, M. (2010). Sarcopenia: European consensus on definition and diagnosis: Report of the European Working Group on Sarcopenia in Older People. Age and Ageing, 39(4), 412-423. doi:10.1093/ageing/afq034

[5] Dhar, S. (1998). A Nonclonogenic Cytotoxicity Assay Using Primary Cultures of Patient Tumor Cells for Anticancer Drug Screening. Journal of Biomolecular Screening, 3(3), 207-216. doi:10.1177/108705719800300307

[6] Fleisher, D. R. (2014). Functional bowel disorders and functional abdominal pain. Management of Functional Gastrointestinal Disorders in Children, 111-129. doi:10.1007/978-1-4939-1089-2_4

[7] Forsburg, S. (2005). Activation of the DNA damage checkpoint and genomic instability in human precancerous lesions. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1020932.302715

[8] Heatherton, T. F., Kozlowski, L. T., Frecker, R. C., & Fagerstrom, K. (1991). The Fagerstrom Test for Nicotine Dependence: a revision of the Fagerstrom Tolerance Questionnaire. Addiction, 86(9), 1119-1127. doi:10.1111/j.1360-0443.1991.tb01879.x

[9] Kawamoto, M. (2006). Assessment of liver fibrosis by a noninvasive method of transient elastography and biochemical markers. World Journal of Gastroenterology, 12(27), 4325. doi:10.3748/wjg.v12.i27.4325

[10] Landingham, S. W., Willis, J. R., Vitale, S., & Ramulu, P. Y. (2012). Visual Field Loss and Accelerometer-Measured Physical Activity in the United States. Ophthalmology, 119(12), 2486-2492. doi:10.1016/j.ophtha.2012.06.034

[11] Mcfadden, G. (2006). Restoring function in exhausted CD8 T cells during chronic viral infection. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.10182.365703

[12] Miyakis, S., Lockshin, M. D., Atsumi, T., Branch, D. W., Brey, R. L., Cervera, R., . . . Krilis, S. A. (2006). International consensus statement on an update of the classification criteria for definite antiphospholipid syndrome (APS). Journal of Thrombosis and Haemostasis, 4(2), 295-306. doi:10.1111/j.1538-7836.2006.01753.x

[13] Re, R., Pellegrini, N., Proteggente, A., Pannala, A., Yang, M., & Rice-Evans, C. (1999). Antioxidant activity applying an improved ABTS radical cation decolorization assay. Free Radical Biology and Medicine, 26(9-10), 1231-1237. doi:10.1016/s0891-5849(98)00315-3

[14] Rosenberg, W., & Donald, A. (1995). Evidence based medicine: an approach to clinical problem-solving. Bmj, 310(6987), 1122-1126. doi:10.1136/bmj.310.6987.1122

[15] Rota, P. A. (2003). Characterization of a Novel Coronavirus Associated with Severe Acute Respiratory Syndrome. Science, 300(5624), 1394-1399. doi:10.1126/science.1085952

[16] Rubinov, M., & Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. NeuroImage, 52(3), 1059-1069. doi:10.1016/j.neuroimage.2009.10.003

[17] Soria, J. C., & Hollebecque, A. (2012). Comprehensive genomic characterization of squamous cell lung cancers. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.717960611.793463544

[18] Thomson, J. A. (1998). Embryonic Stem Cell Lines Derived from Human Blastocysts. Science, 282(5391), 1145-1147. doi:10.1126/science.282.5391.1145

[19] Wagner, E. H., Austin, B. T., Davis, C., Hindmarsh, M., Schaefer, J., & Bonomi, A. (2001). Improving Chronic Illness Care: Translating Evidence Into Action. Health Affairs, 20(6), 64-78. doi:10.1377/hlthaff.20.6.64

[20] Woolf, C. J. (2002). Chondroitinase ABC promotes functional recovery after spinal cord injury. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1005735.68205

Papers in the field of computer science

[21] Bhattacherjee, A. (2001). Understanding Information Systems Continuance: An Expectation-Confirmation Model. MIS Quarterly, 25(3), 351. doi:10.2307/3250921

[22] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145-1159. doi:10.1016/s0031-3203(96)00142-2

[23] Felzenszwalb, P., & Huttenlocher, D. (n.d.). Efficient belief propagation for early vision. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. doi:10.1109/cvpr.2004.1315041

[24] Fischler, M. A., & Bolles, R. C. (1987). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Readings in Computer Vision, 726-740. doi:10.1016/b978-0-08-051581-6.50070-2

[25] Forsyth, D. (2014). Object Detection with Discriminatively Trained Part-Based Models. Computer, 47(2), 6-7. doi:10.1109/mc.2014.42

[26] Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction, 7(2), 174-196. doi:10.1145/353485.353487

[27] Iivari, J. (2010). Twelve Theses on Design Science Research in Information Systems. Integrated Series in Information Systems Design Research in Information Systems, 43-62. doi:10.1007/978-1-4419-5653-_5

[28] Kela, N., Rattani, A., & Gupta, P. (n.d.). Illumination Invariant Elastic Bunch Graph Matching for Efficient Face Recognition. 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW06). doi:10.1109/cvprw.2006.97

[29] Li, X., & Yang, Y. (2012). An experimental comparison of localization accuracy of affine region detectors. 2012 5th International Congress on Image and Signal Processing. doi:10.1109/cisp.2012.6470015

[30] Marzetta, T. L. (2010). Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas. IEEE Transactions on Wireless Communications, 9(11), 3590-3600. doi:10.1109/twc.2010.092810.091092

[31] Nelli, F. (2015). Machine Learning with scikit-learn. Python Data Analytics, 237-264. doi:10.1007/978-1-4842-0958-5_8

[32] P, S. C. (2012). Automatic Facial Expression Analysis A Survey. International Journal of Computer Science & Engineering Survey, 3(6), 47-59. doi:10.5121/ijcses.2012.3604

[33] Ronquist, F., & Huelsenbeck, J. P. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics, 19(12), 1572-1574. doi:10.1093/bioinformatics/btg180

[34] Sadovnik, A., & Chen, T. (2011). Pictorial structures for object recognition and part labeling in drawings. 2011 18th IEEE International Conference on Image Processing. doi: 10.1109/icip.2011.6116499

[35] Snavely, N., Seitz, S. M., & Szeliski, R. (2007). Modeling the World from Internet Photo Collections. International Journal of Computer Vision, 80(2), 189-210. doi:10.1007/s11263-007-0107-3

[36] Taveter, K. (n.d.). Towards Radical Agent-Oriented Software Engineering Processes Based on AOR Modeling. Agent-Oriented Methodologies. doi:10.4018/9781591405818.ch010

[37] Tversky, B., Morrison, J. B., & Betrancourt, M. (2002). Animation: can it facilitate? International Journal of Human-Computer Studies, 57(4), 247-262. doi:10.1006/ijhc.2002.1017

[38] Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, 13(4), 600-612. doi:10.1109/tip.2003.819861

[39] Zeng, Z., Pantic, M., Roisman, G. I., & Huang, T. S. (2007). A survey of affect recognition methods. Proceedings of the ninth international conference on Multimodal interfaces - ICMI 07. doi:10.1145/1322192.1322216

[40] Zhang, Z., Miao, D., & Gao, C. (2013). Short text classification using latent Dirichlet allocation. Journal of Computer Applications, 33(6), 1587-1590. doi:10.3724/sp.j.1087.2013.01587

Papers in the field of biology

[41] Ashraf, M., & Foolad, M. (2007). Roles of glycine betaine and proline in improving plant abiotic stress resistance. Environmental and Experimental Botany, 59(2), 206-216. doi:10.1016/j.envexpbot.2005.12.006

[42] Blomberg, S. P., Garland, T., & Ives, A. R. (2003). Testing For Phylogenetic Signal In Comparative Data: Behavioral Traits Are More Labile. Evolution, 57(4), 717-745. doi:10.1111/j.0014-3820.2003.tb00285.x

[43] Desjeux, P. (2004). Leishmaniasis: current situation and new perspectives. Comparative Immunology, Microbiology and Infectious Diseases, 27(5), 305-318. doi:10.1016/j.cimid.2004.03.004

[44] Dewanto, V., Wu, X., Adom, K. K., & Liu, R. H. (2002). Thermal Processing Enhances the Nutritional Value of Tomatoes by Increasing Total Antioxidant Activity. Journal of Agricultural and Food Chemistry, 50(10), 3010-3014. doi:10.1021/jf0115589

[45] Finn, R. D. (2005). Pfam: the protein families database. Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics. doi:10.1002/047001153x.g306303

[46] Goecks, J., Nekrutenko, A., Taylor, J., & Team, T. G. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biology, 11(8). doi:10.1186/gb-2010-11-8-r86

[47] Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., . . . Yamanishi, Y. (2007). KEGG for linking genomes to life and the environment. Nucleic Acids Research, 36(Database). doi:10.1093/nar/gkm882

[48] Katoh, K. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Research, 30(14), 3059-3066. doi:10.1093/nar/gkf436

[49] Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4), 357-359. doi:10.1038/nmeth.1923

[50] Lo, C., Wang, H., Dembo, M., & Wang, Y. (2000). Cell Movement Is Guided by the Rigidity of the Substrate. Biophysical Journal, 79(1), 144-152. doi:10.1016/s0006-3495(00)76279-5

[51] Malanson, G. (2007). Maximum entropy modeling of species geographic distributions. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1085992.538943

[52] Moriasi, D. N., Arnold, J. G., Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations. Transactions of the ASABE, 50(3), 885-900. doi:10.13031/2013.23153

[53] Okazaki, Y. (2005). Combinatorial microRNA target predictions. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1025845.310832

[54] Palmer, T. (2006). Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1031256.365601

[55] Ramadan, H. A., & A., N. (2012). Biological Identifications Through DNA Barcodes. Biodiversity Conservation and Utilization in a Diverse World. doi:10.5772/49967

[56] Rehmsmeier, M. (n.d.). Prediction of MicroRNA Targets. MicroRNA Protocols, 87-100. doi:10.1385/1-59745-123-1:87

[57] Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., . . . Higgins, D. G. (2014). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Molecular Systems Biology, 7(1), 539-539. doi:10.1038/msb.2011.75

[58] Trombulak, S. C., & Frissell, C. A. (2000). Review of Ecological Effects of Roads on Terrestrial and Aquatic Communities. Conservation Biology, 14(1), 18-30. doi:10.1046/j.1523-1739.2000.99084.x

[59] Untergasser, A., Cutcutache, I., Koressaar, T., Ye, J., Faircloth, B. C., Remm, M., & Rozen, S. G. (2012). Primer3—new capabilities and interfaces. Nucleic Acids Research, 40(15). doi:10.1093/nar/gks596

[60] Whelan, S., & Goldman, N. (2001). A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach. Molecular Biology and Evolution, 18(5), 691-699. doi:10.1093/oxfordjournals.molbev.a003851

Papers in the field of economics (incl. finance)

[61] Arnould, E., & Thompson, C. (2005). Consumer Culture Theory (CCT): Twenty Years of Research. Journal of Consumer Research, 31(4), 868-882. doi:10.1086/426626

[62] Bennett, E. (2009). Defining and classifying ecosystem services for decision making. F1000 - Post-publication peer review of the biomedical literature. doi:10.3410/f.1145051.602178

[63] Boyd, J. W., & Banzhaf, H. S. (2006). What are Ecosystem Services? The Need for Standardized Environmental Accounting Units. SSRN Electronic Journal. doi:10.2139/ssrn.892425

[64] Djankov, S., LaPorta, R., Lopez-de-Silanes, F., & Shleifer, A. (2005). The law and economics of self-dealing. NBER Working Paper No. 11883.

[65] Dolan, P., Peasgood, T., & White, M. (2008). Do we really know what makes us happy? A review of the economic literature on the factors associated with subjective well-being. Journal of Economic Psychology, 29(1), 94-122. doi:10.1016/j.joep.2007.09.001

[66] Feldmana, M. P., & Audretschbc, D. B. (1999). Innovation in cities Science-based diversity, specialization and localized competition. European Economic Review, 43(2), 409-429.

[67] Frey, B. S., & Jegen, R. (1999). Motivation crowding theory - A Survey of Empirical Evidence. Working Paper Series ISSN 1424-0459 , 26 .

[68] Humphrey, J., & Schmitz, H. (n.d.). Governance in global value chains. Local Enterprises in the Global Economy. doi:10.4337/9781843769743.0001126 .

[69] Manning, W., & Mullahy, J. (1999). Estimating Log Models: To Transform or Not to Transform? doi:10.3386/t0246

[70] Mountford, A., & Uhlig, H. (2008). What are the Effects of Fiscal Policy Shocks? doi:10.3386/w14551

[71] Nikiforakis, N. (2005). Punishment and Counter-punishment in Public Good Games: Can We Still Govern Ourselves? SSRN Electronic Journal. doi:10.2139/ssrn.764185

[72] Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on Amazon mechanical turk. Judgment and Decision Making, 5(5).

[73] Peñaloza, L., & Venkatesh, A. (2006). Further evolving the new dominant logic of marketing: from services to the social construction of markets. Marketing Theory, 6(3), 299-316. doi:10.1177/1470593106066789

[74] Porta, R. L., Lopez-De-Silane, F., Shleifer, A., & Vishny, R. (1997). Legal Determinants of External Finance. doi:10.3386/w5879

[75] Porta, R. L., Lopez-de-Silanesb, F., Shleifera, A., & Vishnyc, R. (2000). Investor protection and corporate governance. Journal of Financial Economics, 58(1-2), 3-27.

[76] Roodman, D. (2008). A Note on the Theme of Too Many Instruments. SSRN Electronic Journal. doi:10.2139/ssrn.1101731

[77] Sachs, J. D., & Warner, A. M. (2001). The curse of natural resources. European Economic Review, 45, 827-838.

[78] Siegrist, M., Connor, M., & Keller, C. (2011). Trust, Confidence, Procedural Fairness, Outcome Fairness, Moral Conviction, and the Acceptance of GM Field Experiments. Risk Analysis, 32(8), 1394-1403. doi:10.1111/j.1539-6924.2011.01739.x

[79] Stern, D. I. (2004). The Rise and Fall of the Environmental Kuznets Curve. World Development, 32(8), 1419-1439.

[80] Zavadskas, E. K., & Turskis, Z. (2011). Multiple criteria decision making (MCDM) methods in economics An overview. Technological and Economic Development of Economy, 17(2), 397-427.