TEXT MINING & SEARCH

FINAL PROJECT

Text Classification and Text Summarization of BBC News

Authors:

Yuliia Tsymbal, 894213, y.tsymbal@campus.unimib.it

Paola Maria Cavana, 859341, p.cavana1@campus.unimib.it

Contents

1. Introduction

The British Broadcasting Corporation (BBC) is a British public service broadcaster headquartered at Broadcasting House in London. BBC News is a renowned global news organization delivering timely and impartial coverage across various topics, including world affairs, politics, business, technology, and culture. Known for its credibility, BBC News provides reliable and up-to-the-minute information to a diverse audience worldwide.

In our project, we utilized a dataset available on the Kaggle platform, comprising BBC news articles [1] categorized into different topics. Our primary focus involved employing two key text-mining tasks: text classification and text summarization. Through these tasks, we aimed to enhance our understanding of the content, categorize articles based on topics, and generate concise summaries, contributing to efficient information retrieval and comprehension.

2. Dataset

The dataset used for this project was found on the Kaggle platform. It consists of 2225 articles from the BBC news website corresponding to stories in five topical areas from 2004-2005. Features presented in the dataset are:

- Category of article
- Article id
- Text of article
- Summary of article

Sample form dataset presented in Figure 1.

category	article_id	text	summary
business	73	German growth goes into reverse Germany's economy shrank 0.2% in the last three months of 2004, upsetting hopes of a sustained recovery. The figures confounded hopes of a 0.2% expansion in the fourth quarter in Europe's biggest economy. The Federal Statistics Office said growth for the whole of 2004 was 1.6%, after a year of contraction in 2003, down from an earlier estimate of 1.7%. It said growth in the third quarter had been zero, putting the economy at a standstill from July onward. Germany has been reliant on exports to get its economy back on track, as unemployment of more than five million and impending cuts to welfare mean German consumers have kept their money to themselves. Major companies including Volkswagen, DaimlerChrysler and Siemens have spent much of 2004 in tough talks with unions about trimming jobs and costs. According to the statistics office, Destatis, rising exports were outweighed in the fourth quarter by the continuing weakness of domestic demand. But the relentless rise in the value of the euro last year has also hit the competitiveness of German products overseas. The effect has been to depress prospects for the 12-nation eurozone as a whole, as well as Germany. Eurozone interest rates are at 2%, but senior officials at the rate-setting European Central Bank are beginning to talk about the threat of inflation, prompting fears that interest rates may rise. The ECB's mandate is to fight rising prices by boosting interest rates - and that could further threaten Germany's hopes of recovery.	The figures confounded hopes of a 0.2% expansion in the fourth quarter in Europe's biggest economy.Germany's economy shrank 0.2% in the last three months of 2004, upsetting hopes of a sustained recovery.The ECB's mandate is to fight rising prices by boosting interest rates - and that could further threaten Germany's hopes of recovery.It said growth in the third quarter had been zero, putting the economy at a standstill from July onward.Germany has been reliant on exports to get its economy back on track, as unemployment of more than five million and impending cuts to welfare mean German consumers have kept their money to themselves.

Figure 1. Data example

News is divided into 5 classes, which are:

- Business
- Entertainment
- Politics
- Sport
- Tech

The distribution of the categories shows that the dataset is well-balanced and each category consists of nearly 400 articles (figure 2).

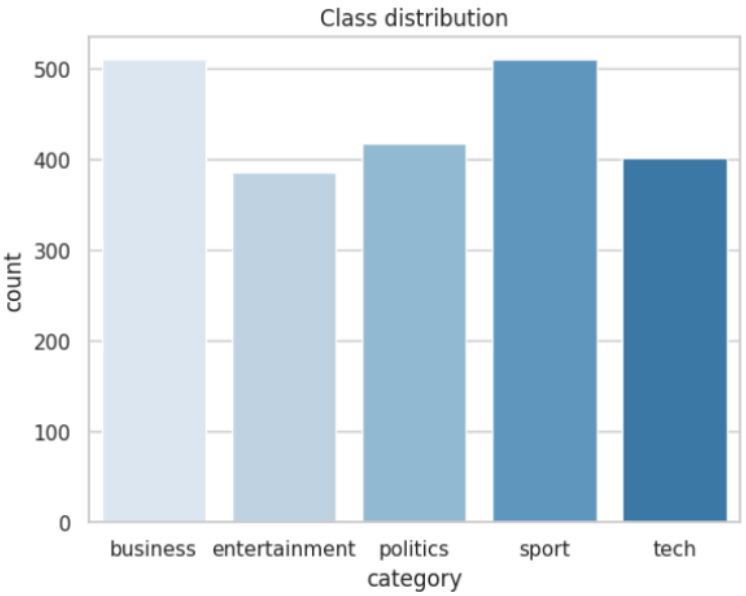
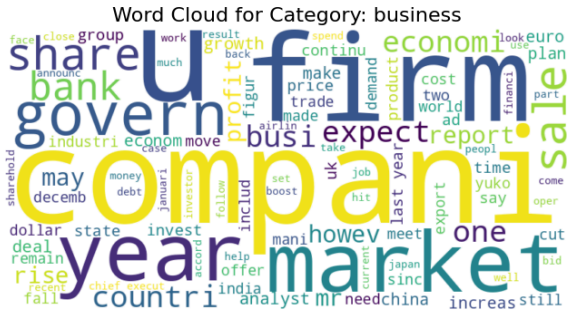
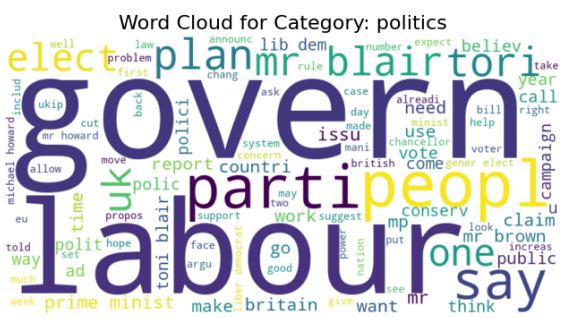


Figure 2. Distribution of categories in the dataset





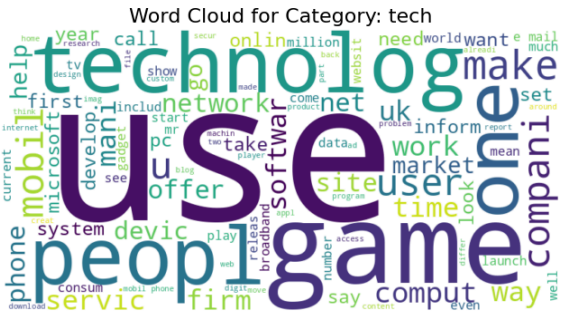


Figure 3. Word clouds for categories

Based on the word cloud (figure 3) for each category we can admire that all of them have specific characteristics in the sense of words that are mostly used in articles. In the case of business, we see top words such as company, firm, and market. On the other hand, the category of entertainment is characterized differently, the main words are music, film, and show.

In our case, for text summarization, it was essential for the dataset to include not only the main body of the news but also an existing summary. This allowed us to conduct a more accurate evaluation upon task completion. Hence, the dataset we selected incorporates this particular feature.

3. Preprocessing

To achieve the goal of classifying and summarizing the news, it was necessary to uniform the texts through the preprocessing phase. In our case, we did:

- Text Normalization
- Stop-Words removal
- Text Tokenization
- Stemming and Lemmatization

All detailed code and solutions can be found on our GitHub repository [2].

3.1 Text Normalization

Text normalization is a preprocessing step in natural language processing (NLP) that involves transforming text data into a standardized or normalized form. The goal is to ensure consistency and reduce variations in the text, making it easier to analyze and extract meaningful insights. In our case, this includes:

- Lowercasing: Converting all letters in the text to lowercase. This helps ensure uniformity and avoids treating words with different cases as distinct.
- Removing punctuation, numbers, and special characters: Eliminating non-alphabetic characters, punctuation, and numerical digits from the text. This simplifies the text and removes elements that may not contribute significantly to the analysis.
- Eliminating Accents: Removing accents from characters to simplify the text and ensure compatibility across different language representations.
- Handling Contractions and Abbreviations: Expanding contractions (e.g., converting "don't" to "do not") and resolving abbreviations contribute to a standardized representation of words.
- Handling Whitespace: Addressing extra spaces, leading or trailing whitespaces, and other spacing issues to maintain a consistent and clean text structure.

Text normalization is crucial in NLP tasks like text classification and summarization because it helps create a more uniform and standardized input, allowing algorithms to focus on the semantic content of the text rather than being hindered by irrelevant variations.

3.2 Stop-Words removal

Stop words are words that are very common and have little or no semantic value, such as articles, prepositions, and conjunctions. Thus, top word removal is a preprocessing step in natural language processing (NLP) that involves excluding common words that typically do not carry significant meaning and are often ignored in the analysis of text data. These words, known as "stop words," are commonly occurring terms such as articles (e.g., "the," "a," "an"), prepositions (e.g., "in," "on," "at"), conjunctions (e.g., "and," "but," "or"), and other high-frequency words. The purpose of removing stop words is to reduce the dimensionality of the data and improve the efficiency and accuracy of text analysis tasks. Since stop words appear frequently across different texts and do not contribute much to the understanding of the content or context, excluding them can lead to more meaningful and relevant representations of the text. However, it's essential to note that the list of stop words may vary based on the context and the specific requirements of the NLP task. In some cases, certain stop words may carry significance, and their removal could affect the overall meaning of the text. Therefore, the decision to remove stop words has been made with consideration for the specific goals of our analysis.

3.3 Text Tokenization

Text tokenization is the process of breaking down a continuous sequence of text into individual units or pieces, referred to as tokens. A token is a fundamental unit that can be a word, phrase, symbol, or any other meaningful element. Tokenization is a crucial preprocessing step in natural language processing (NLP) that enables the analysis and understanding of textual data by converting it into manageable components. In our project we choose the word tokenization, that is breaking down the text into words. Words are typically the basic units of meaning in natural language.

For example, the sentence "`news corp eyes video games market`" has been tokenized in the following way: ["`news`", "`corp`", "`eyes`", "`video`", "`games`", "`market`"].

3.4 Stemming and Lemmatization

Stemming and lemmatization are techniques that reduce words to their base or root form. Stemming usually involves chopping off the suffixes of words, while lemmatization uses a dictionary or a morphological analyzer to find the canonical form of words. Both techniques can help to group words that have similar meanings and reduce the dimensionality and sparsity of data. However, we should be aware of the trade-offs between accuracy and simplicity. For example, stemming may be faster and easier, but it may also produce inaccurate or non-existent words. At the same time, lemmatization may be more accurate and meaningful, but it may also require more resources and time. To achieve our goal, we used the Porter stemmer provided by NLTK.
The final result of the preprocessing step can be found in Table 1.

Table 1.

category	article_id	text	processed_text
business	73	Germany's economy shrank 0.2% in the last three months of 2004, upsetting hopes of a sustained recovery. The figures confounded hopes of a 0.2% expansion in the fourth quarter in Europe's biggest economy. The Federal Statistics Office said growth for the whole of 2004 was 1.6%, after a year of contraction in 2003, down from an earlier estimate of 1.7%. It said growth in the third quarter had been zero, putting the economy at a standstill from July onward.	"german,growth,goe,revers,germani,economi,shrank,last,three,month,upset,hope,sustain,recoveri,figur,confound,hope,expans,fourth,quarter,europ,biggest,economi,feder,statist,offic,growth,whole,year,contract,earlier,estim,growth,third,quarter,zero,put,economi,standstil,juli,onward”

4. Text Classification

Text classification is a fundamental task in text mining that involves categorizing or assigning predefined labels or categories to a given piece of text. The primary goal is to automatically organize and structure large volumes of unstructured text data, making it more manageable and useful for analysis. Text classification is widely used in various applications such as spam detection, sentiment analysis, topic categorization, and document categorization. In the realm of text classification, the choice of text representation and classification algorithms significantly impacts the model's performance. This chapter discusses the utilization of TF-IDF (Term Frequency-Inverse Document Frequency) and Word2Vec for text representation, followed by an exploration of various classification algorithms applied to each representation method. The selected algorithms include Logistic Regression, Decision Tree Classifier, Random Forest Classifier, XGBoost Classifier, and Support Vector Machine (SVM). All detailed code and solutions can be found on our GitHub repository [2].

4.2 Text Representation

The TF-IDF method is a classic approach for converting textual data into a numerical format suitable for machine learning models. It represents documents as vectors based on the frequency of terms within each document, weighted by the inverse document frequency.

Word2Vec is an advanced technique representing words as vectors in a continuous vector space. It captures semantic relationships between words, making it particularly useful for understanding context and meaning in textual data.

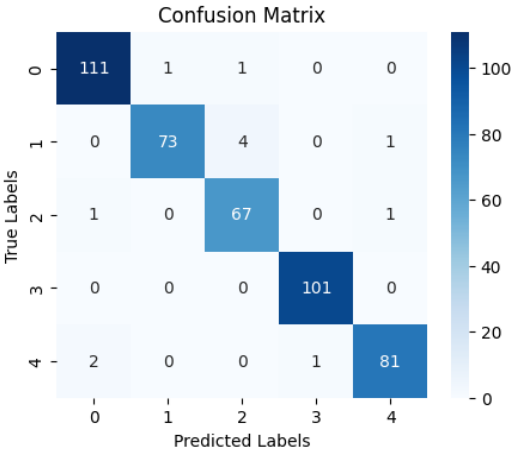
4.2.1 Logistic Regression

Logistic Regression is a linear model that is well-suited for binary and multiclass classification. It works by modeling the probability that a given instance belongs to a particular class.

The results based on the TF-IDF method can be observed using a classification report obtained on a test set of data.

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	113
1	0.99	0.94	0.96	78
2	0.93	0.97	0.95	69
3	0.99	1.00	1.00	101
4	0.98	0.96	0.97	84
accuracy			0.97	445
macro avg	0.97	0.97	0.97	445
weighted avg	0.97	0.97	0.97	445

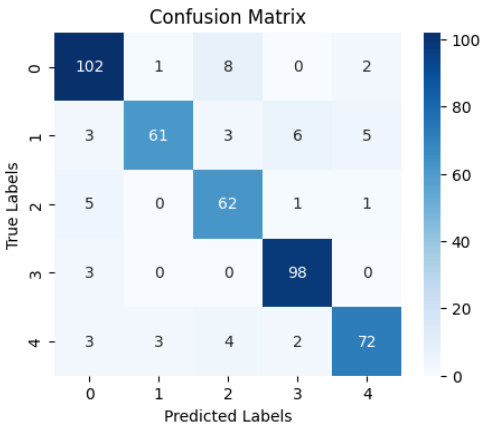


For each category, we get a high value of precision, which is in a range from 0.93 to 0.99. Recall and F1-score have a high value as well. For the feather comparison, we will use the weighted average f1-score, which is 0.97 in this case.
We can also observe a confusion matrix and almost all values are correctly classified, it generally indicates that our model is performing well.

The results of the Word2Vec method can be obtained from the classification report as well.

Classification Report:

	precision	recall	f1-score	support
0	0.88	0.90	0.89	113
1	0.94	0.78	0.85	78
2	0.81	0.90	0.85	69
3	0.92	0.97	0.94	101
4	0.90	0.86	0.88	84
accuracy			0.89	445
macro avg	0.89	0.88	0.88	445
weighted avg	0.89	0.89	0.89	445



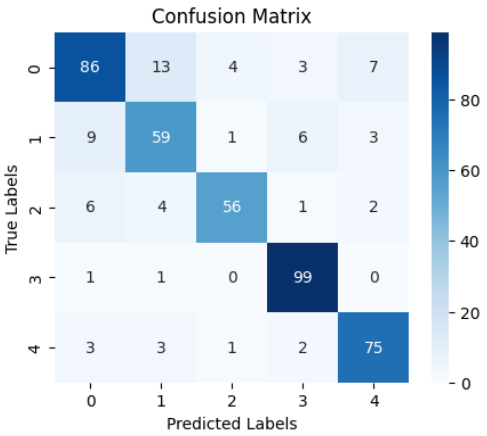
For each category, we see significant differences in the f1-score value and the weighted average f1-score is 0.89, which is much lower than the result obtained based on the tf-idf method.

4.2.2 Decision Tree Classifier

A Decision Tree Classifier is a non-linear model that recursively splits the data based on the features to create a tree-like structure. It is capable of handling both numerical and categorical data, making it versatile for text classification tasks.

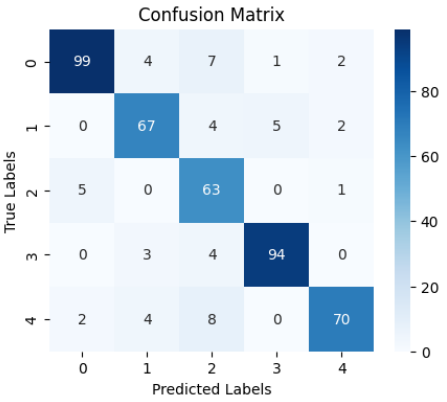
Using TF-IDF representation with a decision tree model we got much lower precision and recall in comparison to Logistic regression. Weighted average f1-score is 0.84.

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

           0       0.82      0.76      0.79       113
           1       0.74      0.76      0.75        78
           2       0.90      0.81      0.85        69
           3       0.89      0.98      0.93       101
           4       0.86      0.89      0.88        84

    accuracy                           0.84       445
   macro avg       0.84      0.84      0.84       445
weighted avg       0.84      0.84      0.84       445
```



At the same time using Word2Vec text representation with the same model we got a bit better performance in comparison with tf-idf, but still worse than logistic regression. The weighted average f1-score is 0.88.

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

           0       0.93      0.88      0.90       113
           1       0.86      0.86      0.86        78
           2       0.73      0.91      0.81        69
           3       0.94      0.93      0.94       101
           4       0.93      0.83      0.88        84

    accuracy                           0.88       445
   macro avg       0.88      0.88      0.88       445
weighted avg       0.89      0.88      0.88       445
```

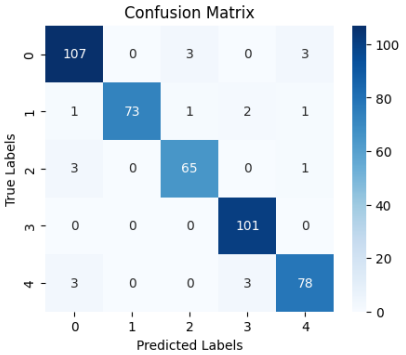


4.2.3 Random Forest Classifier

A Random Forest Classifier is an ensemble learning method that constructs multiple decision trees and merges their predictions. This helps reduce overfitting and improves overall accuracy. In our can for both textual representations random forest gives better performance in comparison with decision trees.

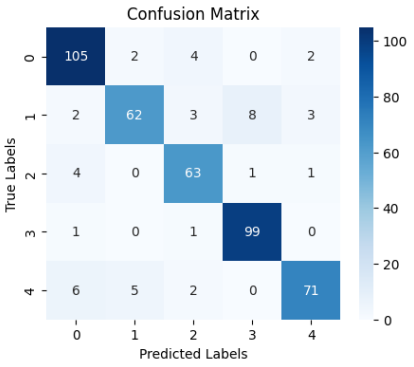
Using TF-IDF we have a weighted average f1-score equal to 0.95, which is better than the previous model on 6 points.

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

           0       0.94      0.95      0.94       113
           1       1.00      0.94      0.97        78
           2       0.94      0.94      0.94        69
           3       0.95      1.00      0.98       101
           4       0.94      0.93      0.93        84

    accuracy                           0.95       445
   macro avg       0.95      0.95      0.95       445
weighted avg       0.95      0.95      0.95       445
```



Using Word2Vec representation we have a weighted average f1-score equal to 0.9, which is not significantly better than the previous model on 2 points.

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

           0       0.89      0.93      0.91       113
           1       0.90      0.79      0.84        78
           2       0.86      0.91      0.89        69
           3       0.92      0.98      0.95       101
           4       0.92      0.85      0.88        84

    accuracy                           0.90       445
   macro avg       0.90      0.89      0.89       445
weighted avg       0.90      0.90      0.90       445
```

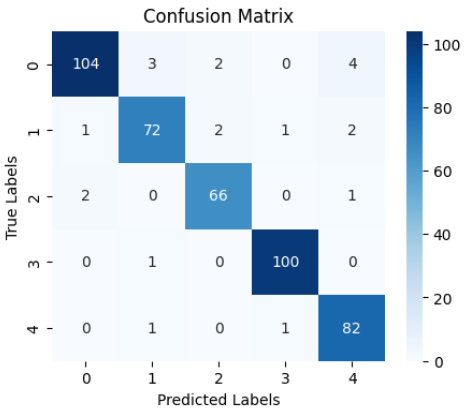


4.2.4 XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a gradient-boosting algorithm known for its efficiency and accuracy. It sequentially builds a series of weak learners to create a robust predictive model. This is only one model out of 5 we test that gives high accuracy based on both textual representations.

For the TF-IDF XGBoost gives 0.95 weighted average f1-score.

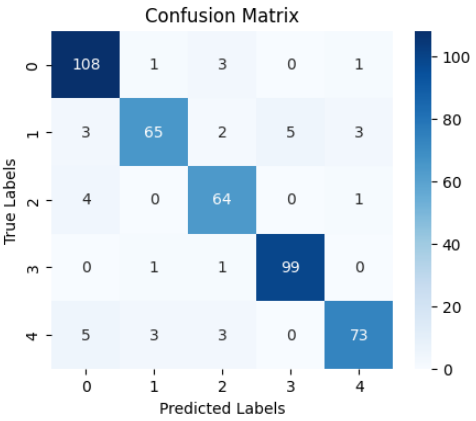
```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.92      0.95       113
           1       0.94      0.92      0.93        78
           2       0.94      0.96      0.95        69
           3       0.98      0.99      0.99       101
           4       0.92      0.98      0.95        84

    accuracy                           0.95       445
   macro avg       0.95      0.95      0.95       445
weighted avg       0.95      0.95      0.95       445
```



For the Word2Vec we have 0.92 weighted average f1-score.

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.96	0.93	113
1	0.93	0.83	0.88	78
2	0.88	0.93	0.90	69
3	0.95	0.98	0.97	101
4	0.94	0.87	0.90	84
accuracy			0.92	445
macro avg	0.92	0.91	0.91	445
weighted avg	0.92	0.92	0.92	445



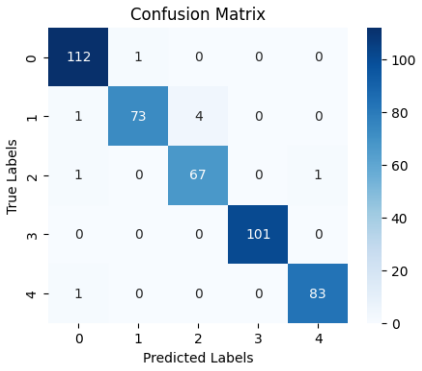
4.2.5 Support Vector Machine (SVM)

A Support Vector Machine is a powerful classification algorithm that works by finding the hyperplane that best separates different classes in the feature space. SVM is effective in high-dimensional spaces, making it suitable for text classification.

Based on the TF-IDF representation SVM model performed the best out of all the models we used for classification. Precision and recall for each category are not less than 0.96 and the weighted average f1-score is 0.98.

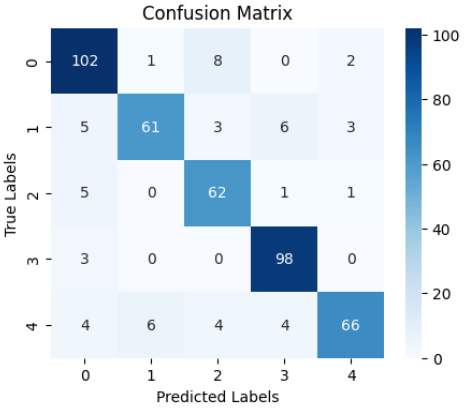
Classification Report:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	113
1	0.99	0.94	0.96	78
2	0.94	0.97	0.96	69
3	1.00	1.00	1.00	101
4	0.99	0.99	0.99	84
accuracy			0.98	445
macro avg	0.98	0.98	0.98	445
weighted avg	0.98	0.98	0.98	445



At the same time based on Word2Vec representation, the SVM model does not the best performance. For categories 1 and 4 we have low recall, not more than 0.79. Almost all categories have an f1-score of less than 0.88. Weighted average f1-score is 0.87.

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

           0       0.86      0.90      0.88       113
           1       0.90      0.78      0.84        78
           2       0.81      0.90      0.85        69
           3       0.90      0.97      0.93       101
           4       0.92      0.79      0.85        84

    accuracy                           0.87       445
   macro avg       0.88      0.87      0.87       445
weighted avg       0.88      0.87      0.87       445
```



4.2.6 Conclusion

This chapter delves into the use of TF-IDF and Word2Vec for text representation, coupled with a selection of classification algorithms. The exploration of both traditional methods like Logistic regression and advanced techniques like XGBoost demonstrates the versatility and adaptability of text classification in handling diverse datasets. The choice between TF-IDF and Word2Vec depends on the specific characteristics of the text data and the objectives of the classification task. As we navigate through the intricacies of each algorithm, we gain insights into their strengths and weaknesses in the context of text classification.

Summarizing all the results presented in this chapter we can identify the best textual representation and algorithm for the classification of our data. For better understanding, we aggregate all results into table 2.

Table 2.

	Model	f1-score_tfidf	f1-score_w2v
0	Logistic Regression	**0.973**	0.887
1	SVM	**0.980**	0.873
2	XGBoost	0.953	0.918
3	DecisionTree	0.841	0.885
4	Random Forest	0.953	0.898

Based on this table we see that the best solution for classifying our dataset is using TF-IDF textual representation and SVM or Logistic regression model. The main

tendency we mention here is that text classification based on Tf-Idf gives better results and requires less computation than the Word2Vec approach.
We can explain these results for several reasons. For classification task that involves assigning documents or pieces of text to predefined categories, and interpretability is crucial, TF-IDF can be a good choice. It provides a representation of the document based on the importance of terms within that document relative to the entire corpus. Instead, word embeddings often capture semantic similarities and relationships between words, which can be beneficial for tasks where context is essential, which is not our case. Also, TF-IDF can perform well with smaller datasets and may be more robust in situations with limited labeled data. Word2Vec models, especially when trained from scratch, may require more data to capture meaningful word representations.

5. Text Summarization

Text summarization is generating a concise and coherent summary of a given text while retaining its essential information and meaning. The goal is to condense the content of a document, article, or any textual material while preserving its key concepts and conveying the main ideas. Text summarization plays a crucial role in information retrieval, allowing users to quickly grasp the main points of a document without going through the entire text.

There are two main approaches to text summarization:

- Extractive Summarization:
    - Extractive summarization involves selecting and combining sentences or phrases directly from the original text to form a summary.
    - It relies on identifying the most important sentences based on various criteria such as word frequency, importance of sentences, or relevance to the document's main theme.
    - Extractive summarization methods do not generate new sentences but rather extract and rearrange existing ones.
- Abstractive Summarization:
    - Abstractive summarization goes beyond the extraction of sentences and aims to generate new sentences that convey the main ideas of the original text more concisely and coherently.

- This approach involves understanding the context of the text and creating a summary using paraphrasing, sentence compression, and even the generation of entirely new sentences.
- Abstractive summarization requires a deeper understanding of language and context and often involves natural language generation techniques.

In our project, we performed both. All detailed code and solutions can be found on our GitHub repository [2].

5.1 Extractive Summarization

In our project, we employed extractive summarization techniques, and specifically, we leveraged a method known as TextRank. TextRank is an algorithm based on graph theory and was originally designed for keyword extraction. However, it has been adapted successfully for extractive summarization as well. It represents sentences as nodes in a graph, with edges indicating the similarity between sentences. By ranking sentences using graph algorithms like PageRank, TextRank identifies the most important sentences, forming a concise summary by selecting top-ranked sentences. For our project, we have implemented the TextRank algorithm using the PyTextRank and spaCy library. SpaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python [3]. PyTextRank is an implementation of the TextRank graph algorithm in Python for use in spaCy pipelines which provides fast, effective phrase extraction from texts, along with extractive summarization. PyTextRank simplifies the implementation process by providing a user-friendly interface and robust functionality for applying TextRank to a given text corpus [4]. We did not implement the TextRank algorithm from scratch due to computational complexity and limited capabilities.

5.1.2 Evaluation

The sample of the extractive summarization results can be observed in Figure 4.

Text	Extracted summary	Reference summary
German growth goes into reverse Germany's economy shrank 0.2% in the last three months of 2004, upsetting hopes of a sustained recovery. The figures confounded hopes of a 0.2% expansion in the fourth quarter in Europe's biggest economy. The Federal Statistics Office said growth for the whole of 2004 was 1.6%, after a year of contraction in 2003, down from an earlier estimate of 1.7%. It said growth in the third quarter had been zero, putting the economy at a standstill from July onward. Germany has been reliant on exports to get its economy back on track, as unemployment of more than five million and impending cuts to welfare mean German consumers have kept their money to themselves. Major companies including Volkswagen, DaimlerChrysler and Siemens have spent much of 2004 in tough talks with unions about trimming jobs and costs. According to the statistics office, Destatis, rising exports were outweighed in the fourth quarter by the continuing weakness of domestic demand. But the relentless rise in the value of the euro last year has also hit the competitiveness of German products overseas. The effect has been to depress prospects for the 12-nation eurozone as a whole, as well as Germany. Eurozone interest rates are at 2%, but senior officials at the rate-setting European Central Bank are beginning to talk about the threat of inflation, prompting fears that interest rates may rise. The ECB's mandate is to fight rising prices by boosting interest rates - and that could further threaten Germany's hopes of recovery.	Eurozone interest rates are at 2%, but senior officials at the rate-setting European Central Bank are beginning to talk about the threat of inflation, prompting fears that interest rates may rise. The ECB's mandate is to fight rising prices by boosting interest rates - and that could further threaten Germany's hopes of recovery. German growth goes into reverse Germany's economy shrank 0.2% in the last three months of 2004, upsetting hopes of a sustained recovery. But the relentless rise in the value of the euro last year has also hit the competitiveness of German products overseas. Germany has been reliant on exports to get its economy back on track, as unemployment of more than five million and impending cuts to welfare mean German consumers have kept their money to themselves.	The figures confounded hopes of a 0.2% expansion in the fourth quarter in Europe's biggest economy.Germany's economy shrank 0.2% in the last three months of 2004, upsetting hopes of a sustained recovery.The ECB's mandate is to fight rising prices by boosting interest rates - and that could further threaten Germany's hopes of recovery.It said growth in the third quarter had been zero, putting the economy at a standstill from July onward.Germany has been reliant on exports to get its economy back on track, as unemployment of more than five million and impending cuts to welfare mean German consumers have kept their money to themselves.

Figure 4. Extractive summarization results

In the evaluation of extractive text summarization, we employed a set of metrics to assess the quality and effectiveness of the generated summaries. The chosen metrics include Rouge1, RougeL, and BLEU.

- ROUGE-1: This metric measures the overlap of unigram (single-word) tokens between the reference summary and the generated summary. It provides insight into the precision of the generated summary at the word level.
- ROUGE-L: RougeL focuses on the longest common subsequence between the reference and generated summaries. This metric takes into account the overall structure and content overlap, allowing for some flexibility in word order and phrasing.
- BLEU: BLEU evaluates the precision of the generated summary by comparing n-grams (typically up to 4-grams) in the generated summary against n-grams in one or more reference summaries. It is commonly used in machine translation and text generation tasks.

ROUGE-1	0.63
ROUGE-l	0.42
BLEU	0.40

The average **Rouge-1** score of **0.63** and the average Rouge-L score of **0.42** suggest that, on average, there is a moderate to good overlap between the generated summaries and the reference summaries. A **Rouge-1** score of **0.63** indicates that, on average, over half of the unigrams in the generated summaries match those in the reference summaries. A **Rouge-L** score of **0.424** suggests that, on average, there is a

good overlap in terms of the longest common subsequences between the generated and reference summaries. However, the interpretation of Rouge scores can vary depending on the specific domain and the nature of the summarization task. Rouge scores provide a quantitative measure of similarity but may not capture all aspects of summary quality. It's often useful to complement automated metrics with human evaluation to obtain a more comprehensive understanding of the generated summaries' quality. A **BLEU** score of **0.404** says that there is some overlap between the extracted summaries and the reference summaries, but there is room for improvement.

5.3 Abstractive Summarization

For the abstractive summarization part, we employed BART [5] (Bidirectional and Auto-Regressive Transformers) from Hugging Face (platform and library for state-of-the-art NLP models and resources). BART is a transformer-based model that has been pre-trained on a diverse range of text data, making it proficient in capturing complex language patterns and generating coherent and contextually relevant text.
To adapt BART for our specific abstractive summarization task, we conducted a fine-tuning process on our dataset. This involves training the model on a more specific and targeted dataset to adapt it to the nuances of our summarization requirements. During the abstractive summarization phase, BART takes in the input text and generates a concise and coherent summary. Unlike extractive summarization, where existing sentences are selected, abstractive summarization involves the model generating new sentences that convey the main ideas in a more condensed form. BART's bidirectional architecture allows it to consider the entire context of the input text, ensuring that the generated summaries are contextually relevant and semantically coherent. The output of BART is a well-constructed summary that distills the essential information from the original text, presenting it in a way that captures the main ideas and key concepts. By incorporating BART into our project, we exploited the power of transformer models to enhance our ability to generate informative and coherent summaries from the input text.

The steps we have taken to perform abstract summarization are listed below:

1. Tokenize and process data (Drop rows where word count exceeds 400. We do this to ensure our model can tokenize with extra padding and still stay under the max length limit of 512)
2. Apply train test split. 80/20

3. Prepare data loaders including attention mask, padding and truncation
4. Train BART-base model using an optimizer
5. Evaluation

During training we have implemented Gradient Accumulation. Gradient Accumulation is a technique that simulates a larger batch size by accumulating gradients from multiple small batches before performing a weight update. This technique has been useful because the available memory in Google Colab is limited, and the batch size that can fit in memory is small.

5.4.1 Evaluation

The sample of the abstractive summarization results can be observed in Figure 5.

	Article	Actual Summary	Predicted Summary
0	summarize: Tsunami 'to hit Sri Lanka banks'\n \n Sri Lanka's banks face hard times following December's tsunami disaster, officials have warned.\n \n The Sri Lanka Banks Association said the waves which killed more than 30,000 people also washed away huge amounts of property which was securing loans. According to its estimate, as much as 13.6% of the loans made by private banks to clients in the disaster zone has been written off or damaged. State-owned lenders may be even worse hit, it said.\n \n The association estimates that the private banking sector has 25bn rupees ($250m; £135m) of loans outstanding in the disaster zone. On one hand, banks are dealing with the death of their customers, along with damaged or destroyed collateral. On the other, most are extending cheap loans for rebuilding and recovery, as well as giving their clients more time to repay existing borrowing. The combination means a revenue shortfall during 2005, SLBA chairman - and Commercial Bank managing director - AL Gooneratne told a news conference. "Most banks have given moratoriums and will not be collecting interest, at least in this quarter," he said. In the public sector, more than one in ten of the state-owned People's Bank's customers in the south of Sri Lanka were affected, a bank spokesman told Reuters. He estimated the bank's loss at 3bn rupees.\n	According to its estimate, as much as 13.6% of the loans made by private banks to clients in the disaster zone has been written off or damaged.In the public sector, more than one in ten of the state-owned People's Bank's customers in the south of Sri Lanka were affected, a bank spokesman told Reuters.The association estimates that the private banking sector has 25bn rupees ($250m; £135m) of loans outstanding in the disaster zone.Sri Lanka's banks face hard times following December's tsunami disaster, officials have warned."Most banks have given moratoriums and will not be collecting interest, at least in this quarter," he said.	summarize: Tsunami 'to hit Sri Lanka banks' to be even worse hit, it said. Please calm down, it doesn't be as hard as when it's on a rainy day in the capital, it is only worse hit. It is not as much as what is on the bank bank, it says. The Sri Lanka Banks Association said the waves which killed more than 30,000 people also washed away huge amounts of property which was securing loans. According to its estimate, a much as 13.6% of the loans made by private banks to clients in the disaster zone has been written off or damaged. State-owned lenders may be even bigger hit, its said. Poker

Figure 5. Abstractive summarization results

In the evaluation of abstractive text summarization, we employed a set of metrics to assess the quality and effectiveness of the generated summaries. The chosen metrics include Rouge1, Rouge2, RougeL, and BLEU.

- ROUGE-1: This metric measures the overlap of unigram (single-word) tokens between the reference summary and the generated summary. It provides insight into the precision of the generated summary at the word level.
- ROUGE-2: Rouge2 assesses the similarity in bigram (two-word) overlap between the reference and generated summaries. It considers pairs of consecutive words, providing a measure of how well the model captures the context and relationships between words.
- ROUGE-L: RougeL focuses on the longest common subsequence between the reference and generated summaries. This metric takes into account the overall

structure and content overlap, allowing for some flexibility in word order and phrasing.

- BLEU: BLEU evaluates the precision of the generated summary by comparing n-grams (typically up to 4-grams) in the generated summary against n-grams in one or more reference summaries. It is commonly used in machine translation and text generation tasks.

These metrics collectively provide a comprehensive assessment of the abstractive text summarization system. The results from these metrics help in understanding the strengths and weaknesses of the summarization model and guide improvements for enhanced performance.

ROUGE-1	0.52
ROUGE-2	0.37
ROUGE-l	0.33
BLEU	0.26

6. Conclusion

In conclusion, our project focused on the classification and summarization of BBC news articles using a carefully curated dataset from Kaggle. To achieve accurate and meaningful results, we implemented a series of text preprocessing techniques, including text normalization, stop-word removal, text tokenization, stemming, and lemmatization. These steps were essential for creating a standardized and manageable representation of the text data.

Throughout the project, our methodology prioritized precision and relevance, ensuring that the processed data accurately reflected the underlying semantics of the news articles. The implementation of these techniques not only contributed to the successful accomplishment of our classification and summarization goals but also demonstrated the importance of robust text preprocessing in optimizing the performance of natural language processing tasks. As we move forward, the insights gained from this project can be applied to various domains, fostering a deeper understanding of text mining techniques and their implications for efficient information extraction and analysis. The project showcases the capabilities of machine learning in handling vast amounts of textual data.

7. References

1. BBC News Summary Dataset. [Online]. Available: https://www.kaggle.com/datasets/pariza/bbc-news-summary
2. GitHub repository with all material for the project. Available: https://github.com/JuliaTsymbal/Text-Mining-Project
3. SpaCy: Trained Models & Pipelines. Available: https://spacy.io/models
4. SpaCy: PyTextRank implementation. Available: https://spacy.io/universe/project/spacy-pytextrank
5. BART model source. Available: https://huggingface.co/facebook/bart-base