Text Mining Project

Text Classification and Topic Modeling

Davide Croatto, Hubert Nowak, Eleonora Zullo

Contents

1 Introduction

Text mining is a fundamental area of **natural language processing (NLP)** that focuses on extracting meaningful patterns and insights from textual data. In this project, we employed **text mining techniques** to both classify news articles into categories and uncover the underlying themes within the news content. By applying a combination of preprocessing, classification, and topic modeling methods, our aim was to achieve accurate classification and revealing the themes, improving the interpretability of the news data.

The process began with **text preprocessing**, a critical step to clean and standardize the text. This phase involved tasks such as tokenization, removing non-alphabetic characters, eliminating stopwords, and lemmatization. These transformations prepared the text for subsequent analytical tasks by reducing noise and ensuring consistency.

Following preprocessing, we applied **text classification** to categorize news articles into distinct categories. In this process, we compared different traditional classifiers that utilize representations such as Bag of Words (BoW) and TF-IDF. Additionally, we incorporated an embedding-based approach, BERT, to evaluate how it performs relative to the traditional methods. By leveraging these diverse techniques, we assessed their effectiveness in organizing and categorizing the news content.

Lastly, we employed **topic modeling**, an unsupervised technique, to uncover underlying themes within the news data. By applying the **Latent Dirichlet Allocation (LDA) model** with different numbers of topics for each news category, we aimed to identify the optimal topic distribution that most effectively represents the thematic structure within each category. This approach yielded deeper insights into the topic composition of the news articles.

Together, these steps – text preprocessing, classification, and topic modeling – formed a comprehensive workflow to analyze and interpret news data. This report outlines the **methodologies used**, the results obtained, and the broader **implications for analyzing large collections of news articles** through text mining techniques.

2 Dataset

The dataset used in this project is the **AG's News Topic Classification Dataset**, a widely recognized resource available on Kaggle. This dataset was originally compiled by **ComeToMyHead**, an academic

news search engine operational since July 2004. It represents a curated subset of over 1 million news articles aggregated from more than 2,000 sources within a year, providing a robust foundation for various text mining tasks.

Each entry in the dataset comprises a **headline** and a short **description** of a news article, organized into four predefined categories: **World**, **Sports**, **Business**, and **Science/Technology**, which serve as the target labels. The dataset includes a total of **120,000 training samples** and **7,600 test samples**, with an even distribution across all categories, ensuring balanced representation and unbiased evaluation in analyses.

Each observation is structured into three key fields:

- **Class Index (1 to 4)**: identifies the topic category of the news article.
- **Title**: the headline summarizing the article's content.
- **Description**: a brief elaboration providing context and additional details beyond the headline.

Moreover, the dataset's four category labels are defined as follows:

1. **World (**`Class Index = 1`**)**: encompasses international news, politics, and global events.
2. **Sports (**`Class Index = 2`**)**: includes news on sports activities, events, and competitions.
3. **Business (**`Class Index = 3`**)**: covers economics, financial markets, and corporate affairs.
4. **Sci/Tech (**`Class Index = 4`**)**: highlights advancements, scientific research, and technological innovations.

In conclusion, the **AG's News Topic Classification Dataset** is a versatile resource for text mining tasks, including pattern exploration, classification modeling, and topic modeling. Its balanced class distribution ensures reliable analyses, making it ideal for evaluating a range of natural language processing (NLP) techniques, from traditional statistical models to advanced deep learning methods.

3 Text Preprocessing

The first step of the project involved **text preprocessing**, a crucial phase in text mining tasks that prepares raw text for analysis by ensuring consistency, reducing noise, and standardizing its structure. This process improves the performance of text mining models by allowing them to concentrate on the most relevant patterns in the data.

Specifically, we applied a preprocessing pipeline to the **AG's News Topic Classification Dataset**. As part of this pipeline, we created a new column called `processed`, which contains the result of applying the `data_processing` function to the combined `title` and `description` of each news article, from both the training and test datasets. This column serves as the cleaned and transformed text, ensuring that the input data is consistent and ready for further analysis.

The steps we followed in the `data_processing` function are described below:

- **Removing URLs**: all URLs were removed to eliminate references to external links, which do not contribute to the semantic content of the text.
- **Removing non-alphabetic characters**: this step eliminated all characters that are not part of the alphabet, such as numbers, punctuation, and special symbols. By focusing on alphabetic content, we reduced irrelevant noise that could hinder model performance.
- **Reducing multiple spaces**: consecutive spaces between words were replaced with a single space. This step ensured consistent formatting and avoided unnecessary computational overhead during tokenization and further processing.
- **Converting to lowercase**: all text was transformed to lowercase. This normalization step removed case sensitivity, ensuring that words like `News` and `news` were treated identically.
- **Removing stopwords**: commonly used words that do not contribute significantly to the semantic meaning of the text, such as `the`, `and`, or `is`, were removed. Additionally, we augmented the standard English stopword list with terms specific to the dataset, such as `ap`, `gt`, and `reuters`, which referred to news sources. This step helped focus the analysis on meaningful terms relevant to the classification task.
- **Lemmatizing**: words were reduced to their base or dictionary form, ensuring that words like `running`, `runs`, and `ran` were treated as `run`. Unlike stemming, lemmatization preserved the grammatical meaning of words, further enhancing data quality for analysis.

Beyond these preprocessing steps, an important aspect of our pipeline was **handling bigrams and trigrams** in the text.

Bigrams and trigrams are sequences of two or three consecutive words, respectively, that capture meaningful phrases in the data. For example, `New York` is a bigram, while `New York Times` is a trigram. Recognizing and preserving these multi-word expressions is crucial because they often carry semantic meaning that single words cannot capture.

More precisely, to leverage bigrams and trigrams in our analysis, we followed these steps:

- **Identifying frequent bigrams and trigrams**: we analyzed the dataset to compute the frequencies of all two-word and three-word combinations within the text, identifying the most commonly occurring ones.

- **Creating variables for frequent bigrams and trigrams**: we stored the most frequent bigrams and trigrams in separate variables, allowing us to focus on key phrases that appear consistently across the dataset.

- **Joining bigrams and trigrams**: we searched the text for the identified bigrams and trigrams and replaced them with their joined forms, connecting the words with underscores. For instance, `New York` became `new_york`, and `New York Times` became `new_york_times`.

This step was critical, particularly for subsequent tasks such as topic modeling. By treating multi-word expressions as single tokens, we preserved their contextual meaning and reduced ambiguity. For example, the word `new` alone may have multiple interpretations, but the token `new_york` clearly refers to a specific geographical location. Consequently, this approach ensures more accurate topic modeling, as different topics can emerge based on the preserved semantics of these multi-word phrases.

The combination of all these steps, implemented through the `data_processing` function, ensured that the text was clean, standardized, and ready for further analysis. The resulting `processed` column serves as the foundation for subsequent text mining tasks: text classifiation and topic modeling.

In fact, we applied **text classification** to organize the news articles into predefined categories, ensuring systematic and accurate categorization. At the same time, **topic modeling** was used to uncover the main themes within each category, offering deeper insights into the dominant topics and underlying patterns in the dataset. Together, these techniques effectively structured the data and revealed valuable thematic insights, supporting a robust and comprehensive analysis.

4 Text Classification

Text classification is a fundamental task in natural language processing that involves assigning predefined categories to textual data. In this project, we applied text classification to categorize news articles into one of four labels: World, Sports, Business, and Science/Technology.

The classification models used the cleaned and processed text from the processed column, created during the preprocessing phase described earlier.

To represent the text numerically for the models, we employed two widely-used text representation techniques:

- **Bag of Words (BoW):** this technique represents text data as a collection of words, where each word is treated as a feature. The order of words is not considered, but the presence or absence of specific words is. This representation focuses on word frequency but does not account for word context.

- **TF-IDF (Term Frequency-Inverse Document Frequency):** this method calculates the importance of each word in the document based on how frequently it appears in the document and across all documents. It assigns higher weights to words that appear frequently in a specific document but rarely across the entire dataset.

Succesively, we trained and evaluated three machine learning classifiers using both text representations. Specifically, as the AG's News Topic Classification Dataset contained training and test samples, the train dataset was used to train the models, enabling them to learn patterns and relationships between the text and their respective categories. The test dataset was then used to evaluate the models' performance and measure their ability to generalize to unseen data. The three classifiers used were:

1. **Decision Tree Classifier:** this model splits the data into branches based on feature values, ultimately leading to a decision. It is simple and interpretable, making it suitable for classification tasks with structured data.

2. **Support Vector Classifier (SVC):** this classifier is effective in high-dimensional spaces, such as text data. It finds the optimal hyperplane that best separates the different classes in the dataset, making it a powerful choice for text classification.

3. **Random Forest Classifier:** an ensemble method that combines multiple decision trees to improve the accuracy and robustness of predictions. It works well with complex datasets, providing a strong classification performance.

By applying these classifiers to both Bag of Words and TF-IDF representations, we explored how different text representations and models affect classification performance. This comparative approach helps identify the most effective combination for accurately classifying news articles into their respective categories.

The classification performance of each model was assessed using several metrics: accuracy, precision, recall, and F1-score. These metrics provide insights into how well each model classifies the data and handles various types of errors.

- **Accuracy**: Represents the percentage of correctly predicted instances in the test set.

- **Precision**: Measures how many of the predicted positive instances are actually positive.

- **Recall**: Indicates how many of the actual positive instances were correctly identified.
- **F1-score**: The harmonic mean of precision and recall, providing a balance between the two metrics.

4.1 Evaluation

The results, see tables 1 and 2, of the classification experiments using Decision Tree, Support Vector Classifier (SVC), and Random Forest across the two text representations - Bag of Words (BoW) and TF-IDF- demonstrate notable differences in performance while also highlighting consistent trends.

Among the three classifiers, **SVC consistently outperformed both Decision Tree and Random Forest in terms of accuracy and F1-score**. Specifically, SVC achieved an accuracy of approximately 0.90 for both representations, showing strong precision, recall, and F1-scores across all the four categories. This performance reflects the robustness of SVC for text classification tasks, especially when working with high-dimensional feature spaces. Its strength lies in fact in its ability to separate classes linearly, which appears to align well with the structure of the dataset. For instance, categories such as "Sport" and "World" benefited the most, achieving high F1-scores, suggesting that these topics are more distinguishable based on the word distributions present in the dataset.

The **Decision Tree classifier** exhibited the lowest performance, with an accuracy of approximately 0.78 for both text representations. While it achieved reasonable scores, it struggled compared to the more sophisticated ensemble and kernel-based methods, likely due to its propensity to overfit and limited generalization capabilities. For example, in the "Business" and "Sci/Tech" categories, the frequent use of shared terminology seems to have led to some confusion, whereas the "Sport" category, with its more distinct and specialized vocabulary, yielded much better results.

Random Forest delivered the second-best performance, achieving an accuracy of around 0.88 for both TF-IDF and BoW. This indicates its ability to capture complex patterns in the data, though it falls slightly behind SVC, particularly in precision and recall consistency. By combining multiple decision trees, it mitigated some of the overfitting issues seen in the Decision Tree model while capturing more complex patterns in the data. Yet, it still couldn't fully match the performance of the Sparse Vector Classifier, likely because it doesn't leverage the sparsity of the data as effectively. Interestingly, the Random Forest performed exceptionally well in the "Sport" category, suggesting that this category may contain subtle, non-linear patterns that this model could exploit. Actually, what stands out across all models is the consistently high performance in the "Sport" category. This suggests that the words associated with sports are more unique and less likely to appear in other categories, making it easier for the algorithms to classify. In contrast, the lower scores in "Business" and "Sci/Tech" point to a greater degree of lexical overlap, which complicates the classification process.

The results demonstrate very similar classification performance between the two text representa-

BoW									
	Decision Tree Classifier			**Sparse Vector Classifier**			**Random Forest Classifier**		
	Accuracy: 0.79			Accuracy: 0.90			Accuracy: 0.88		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Business	0.75	0.74	0.74	0.88	0.86	0.87	0.85	0.84	0.84
Sci/Tech	0.76	0.74	0.75	0.87	0.88	0.88	0.86	0.83	0.84
Sport	0.84	0.89	0.86	0.94	0.98	0.96	0.90	0.97	0.93
World	0.81	0.80	0.80	0.92	0.89	0.91	0.92	0.89	0.91

Table 1: Performance of classifiers using Bag of Words representations.

TF-IDF									
	Decision Tree Classifier			**Sparse Vector Classifier**			**Random Forest Classifier**		
	Accuracy: 0.78			Accuracy: 0.90			Accuracy: 0.88		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Business	0.73	0.73	0.73	0.89	0.87	0.88	0.85	0.84	0.85
Sci/Tech	0.75	0.73	0.74	0.88	0.89	0.89	0.85	0.83	0.84
Sport	0.83	0.88	0.85	0.95	0.98	0.96	0.90	0.97	0.93
World	0.80	0.78	0.79	0.92	0.89	0.91	0.90	0.87	0.88

Table 2: Performance of classifiers using TF-IDF representations.

tions, which can be attributed to the relatively short nature of the documents and the lack of significant repetition of terms within them. As a result, the weighting introduced by TF-IDF did not significantly differ from the raw frequency counts in Bag-of-Words, minimizing TF-IDF's typical advantage of emphasizing rare but informative terms. This highlights the importance of considering the specific characteristics of the data when selecting a text representation. Notably, when applied to the Decision Tree model, Bag-of-Words slightly outperformed TF-IDF, suggesting that the simpler representation was better suited to the dataset's structure. The short length of the documents reduced the utility of TF-IDF's weighting mechanism, allowing Bag-of-Words to effectively capture the key features necessary for classification.

4.2 Using BERT

After exploring text classification using Bag-of-Words and TF-IDF representations, we decided to take a further step by analyzing the impact of word embeddings on classification performance to determine if they could yield better results. Word embeddings are a powerful technique in natural language processing (NLP) that represent words as dense vectors in a high-dimensional space, capturing semantic and syntactic relationships between words based on their contextual usage. Unlike traditional text representations, embeddings enable models to understand nuanced word meanings and relationships, significantly enhancing performance on various NLP tasks.

For this purpose, we utilized BERT (Bidirectional Encoder Representations from Transformers), a pretrained language model developed by Google. BERT uses a bidirectional transformer architecture to learn deep contextual representations of text by considering both the left and right context of a word in a sentence simultaneously. This enables BERT to generate rich embeddings that encode not only word meanings but also the relationships between words in a given context.

In our text classification task, we employed BERT to generate word embeddings and train a classifier. The process began with renaming the dataframe variable to avoid conflicts with the previous steps. Next, we prepared the data to work seamlessly with PyTorch's machine learning functionalities, ensuring compatibility with BERT's requirements. We then defined a custom classification class to leverage BERT's embeddings for our specific task. To achieve this, the classification pipeline was divided into several key steps:

1. **Tokenization:** The input text was tokenized using BERT's tokenizer, which splits the text into subwords and adds special tokens like [CLS] (indicating the start of a sequence) and [SEP] (indicating the end or separation between sequences).
2. **Embedding Generation:** The tokenized input was passed through BERT's encoder layers, where it generated a contextualized embedding for each token. These embeddings incorporate information from both the token itself and its surrounding context in the sentence.
3. **Pooling:** From the sequence of token embeddings, the [CLS] token's embedding was extracted as a fixed-length representation of the entire input text. This embedding is often used as the input to the classification layer because it summarizes the sequence's contextual information.
4. **Classification Layer:** The [CLS] embedding was passed through a feedforward neural network (classification head) consisting of one or more fully connected layers with a softmax activation. This produced probability scores for each class label.
5. **Loss Calculation and Optimization:** The model's predictions were compared to the true labels using a cross-entropy loss function. An optimizer, such as AdamW, adjusted the model's weights during backpropagation to minimize the loss.

Training parameters and evaluation metrics were carefully outlined to optimize the model's performance. We also chose hyperparameters tailored to the dataset and task, ensuring an effective balance between speed and accuracy. With the setup complete, we prepared the data for the training step by optimizing certain processes to streamline execution and reduce runtime. This included selecting an appropriate optimizer to adjust the model's weights effectively during training. Finally, we conducted the training and evaluation phases, allowing the model to learn from the processed text data and assess its classification performance. By incorporating BERT, we aimed to leverage its capabilities to improve the quality and accuracy of our text classification results, taking full advantage of its contextual understanding of language.

BERT classifier			
Accuracy: 0.93			
	Precision	**Recall**	**F1-score**
Business	0.89	0.90	0.89
Sci/Tech	0.91	0.90	0.90
Sport	0.98	0.99	0.98
World	0.95	0.94	0.94

The BERT-based text classification model demonstrated strong overall performance, see the table above, with validation accuracy steadily improving over the epochs and stabilizing at around 93.27% by

the fourth epoch. The weighted average F1-score remained consistently high at 0.93, indicating balanced performance across the four news categories: World, Business, Sport, and Sci/Tech. Class-specific metrics revealed that the model performed particularly well in identifying the ”World” and ”Sports” categories, achieving high precision and recall, which reflected its confidence and accuracy. However, the ”Sci/Tech” and ”Business” categories showed slightly lower recall and F1-scores, ranging from 0.89 to 0.90, suggesting challenges in distinguishing these classes, possibly due to overlapping vocabulary or less distinct features.

When compared to traditional machine learning classifiers, such as Decision Tree, Sparse Vector Classifier (SVC), and Random Forest, which use TF-IDF and Bag-of-Words (BoW) text representations, BERT's superiority is clear. While the best traditional approach achieves an accuracy of 90%, BERT surpasses this with an accuracy of 93.27%. Random Forest and Decision Tree perform less effectively, with overall accuracies of 88% and around 78 - 79%, respectively. Traditional methods also show variability in performance across categories, often excelling in straightforward classes like ”Sport” but struggling in more nuanced ones like ”Business” and ”Sci/Tech.” This highlights the limitations of frequency-based text representations like TF-IDF and BoW, which fail to capture the deeper semantic and contextual nuances that BERT effectively leverages for superior classification performance.

In conclusion, the BERT model outperforms traditional methods by a significant margin due to its ability to leverage contextualized embeddings that capture deeper semantic relationships within the text.

5 Topic Modeling

Topic modeling is an unsupervised machine learning technique used to uncover the hidden thematic structure within large collections of text. By analyzing word patterns and co-occurrences, it identifies groups of words that frequently appear together, which are interpreted as latent ”topics” in the dataset.

In our project, the topic modeling task was aimed at identifying the main topics within each news category - Business, Sport, World and Sci/Tech. This process was carried out in two distinct phases: the first focused on building an LDA model with varying numbers of topics for each category, while the second phase involved analyzing the evualtion metrics for each LDA model and visualizing the results through word clouds.

In the first phase, we applied a Latent Dirichlet Allocation (LDA) model iteratively, increasing the number of topics from 3 to 20 in single increments. For each iteration, we evaluated the model using two key metrics: perplexity and coherence, which help assess the quality of the generated topics. Perplexity measures how well the model predicts the data, with lower values indicating better generalization. However, perplexity tends to decrease steadily as the number of topics increases, making it less effective for identifying the optimal number of topics. To address this, we prioritized the coherence score, particularly the c_v coherence measure, which evaluates the semantic similarity of high-probability words within each topic.

To analyze better the results, we created two distinct plots for each news category. The first plot illustrates how perplexity changes as the number of topics increases, highlighting the steady decline in perplexity values. The second plot shows how the c_v coherence score varies with the number of topics, helping to identify the point at which topics become most interpretable and coherent.

After evaluating the perplexity and coherence values, we moved to the second phase, which focused on visualizing the most significant topics for each category. Based on the insights from the first phase, we selected the optimal number of topics and generated word clouds for each category. These word clouds helped us visualize the most frequent words associated with each topic, providing a clearer understanding of the key themes and terms present in the dataset.

By combining the LDA model with iterative topic analysis and the visualization of word clouds, we were able to both quantitatively and qualitatively refine our understanding of the thematic structure in the dataset. This process allowed us to identify the most relevant topics for each category and visualize their key words, providing a more interpretable and meaningful analysis of the news articles.

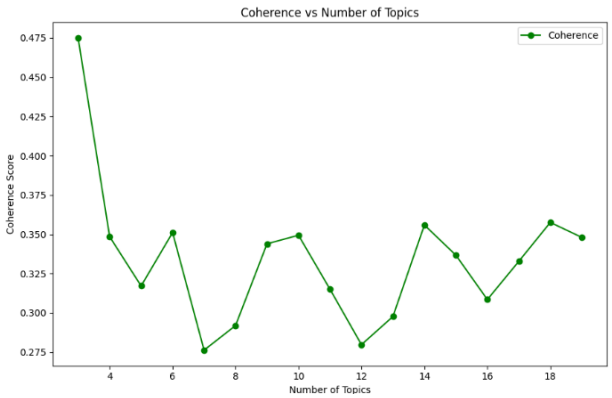
5.1 Results

In analyzing the results of our topic modeling process, we focused primarily on the c_v coherence plots for each category, as coherence provides a more interpretable measure of topic quality compared to perplexity. While perplexity is a useful metric, its behavior across all categories showed limited variability, with values remaining relatively stable up to around 10 topics before declining. This consistent trend made it less informative for determining the optimal number of topics. In contrast, the coherence plots offered clearer insights by highlighting the point at which the generated topics were most semantically meaningful and interpretable. Consequently, our analysis prioritized c_v coherence scores to identify the optimal topic structure for each category.

For the **Business** category, we evaluated models with 3, 6 and 10 topics, as these points stood out due to notable trends in the coherence scores. The 3-topic model was chosen as it represented the highest coherence score and provided a concise thematic structure. The 6-topic model was also explored due to its slight increase in coherence, indicating the po-

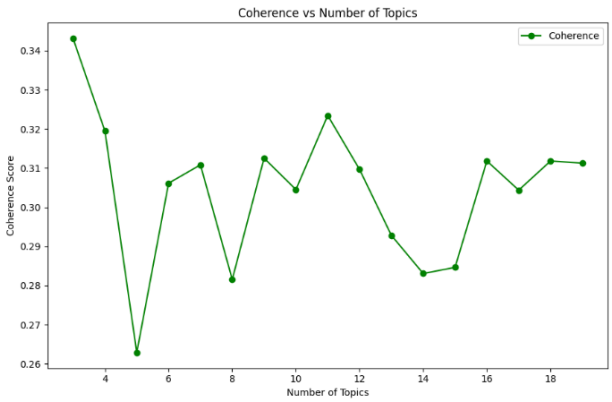
tential for more nuanced topic differentiation. Additionally, we analyzed the 10-topic model, as it exhibited another upward trend, suggesting improved interpretability with a greater number of topics. However, we chose not to evaluate models with 14 or 18 topics, despite the increase in coherence, as the larger number of topics risked introducing excessive granularity and repetition across topics.

The coherence plot for the Business category is shown in the following figure.



After examining the content and semantic clarity of the topics generated at each of these key points, we selected for the Business category the model with 6 topics. This choice allowed us to identify more distinct themes compared to the 3-topic model while avoiding the overlap observed in the 10-topic model, where the additional topics, though distinct, shared a similar context as they all focused on investment, though in different fields such as technology or companies. The 6-topic model struck the right balance, providing meaningful and interpretable insights into the thematic structure of the Business category.

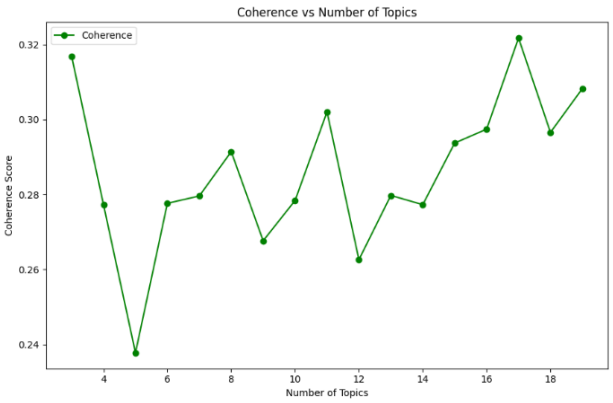
For the **Sport** category, we evaluated models with 3, 7, and 11 topics, since these points demonstrated notable trends in the coherence scores, as shown in the following image.



The 3-topic model was selected because it showed the highest c_v coherence value. The 7-topic model was also analyzed due to its significant increase in coherence after a dip at 5 topics, suggesting an improvement in the semantic quality and granularity of the topics. Additionally, the 11-topic model was chosen due to the sharp peak in coherence, indicating strong interpretability and meaningful topic differentiation at this point. Further models were not analyzed, as no other model showed a higher coherence score than the 11-topic model.

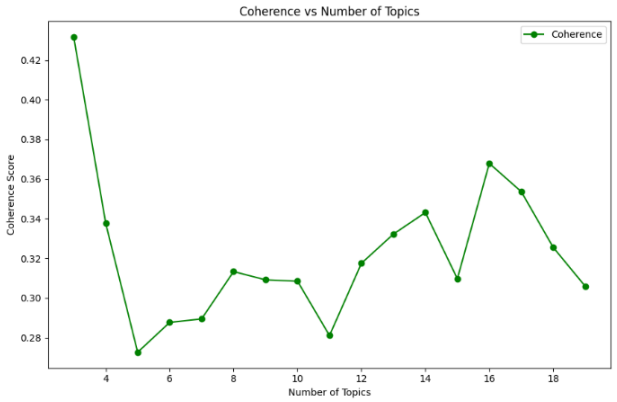
By carefully examining the content and interpretability of topics generated at 3, 7, and 11 topics, we identified the 11-topic model as the one that most effectively balanced coherence, granularity, and diversity. In the world of sports, the diversity of topics is vast, encompassing different sports, specific matches, and prominent figures. Models with 3 and 7 topics were not sufficient to capture this complexity, as they failed to represent the full range of themes present in the data. The 11-topic model, on the other hand, effectively highlights this diversity, with distinct references to football (e.g., 'championship'), tennis (e.g., 'Williams' and 'Roddick' appearing in separate topics), basketball (e.g., 'basketball' and 'New York'), and even specific matchdays (e.g., mentions of 'Friday' or 'Saturday'). This demonstrates the model's ability to uncover varied and meaningful themes, making it the most suitable for analyzing the Sport category.

For the **Science/Technology** category, we evaluated models with 3, 8, 11 and 17 topics. The 3-topic model is always the one with one of the highest c_v coherence score, the 8-topic model and the 11-topic model were selected due to the noticeable upward trend in coherence after a dip, and the 17-topic model was considered as it displayed the highest coherence score overall, indicating a strong separation and semantic quality of topics at this point. These considerations can be detected in the following plot.



After qualitatively evaluating the interpretability of the generated topics, we selected the 11-topic model, as the 17-topic model appeared somewhat redundant. In contrast, the 11 topics identified by the model cover a wide range of distinct themes, from various technology companies like Sony, Intel, Microsoft, and IBM, to broader subjects such as the internet, scientific research, NASA, and data. This model effectively avoids redundancy and repetition, confirming that the choice of 11 topics —rather than 3, 8, or 17 — strikes the right balance between thematic diversity and clarity.

Finally, for the **World** category, we evaluated models with 3, 8, 14 and 16 topics, the motivations can be seen in the following coherence plot.



The optimal number of topics identified in this case is 14, as the World category encompasses a broad and diverse range of subjects. World news often covers a variety of global issues, making it essential to balance thematic diversity with coherence. After a qualitative evaluation of the topics generated by different models, we determined that the 14-topic model strikes the best balance, capturing distinct themes without redundancy. This model addresses a wide array of issues, including political and economic challenges in countries like Russia, China, Japan, and India, which reflect the central role of foreign policy in world news. It also captures contemporary conflicts, with terms such as 'Gaza', 'Israeli', and 'Iraq' appearing across different topics, as well as political leadership themes such as 'prime minister', 'hostage', and 'soldier.' Overall, the 14-topic model effectively captures the varied, non-redundant themes that characterize global news coverage.

In conclusion, our analysis highlights the importance of c_v coherence scores in selecting the optimal number of topics, along with the crucial role of human intervention in qualitatively evaluating the topics. By considering both the coherence plots and the interpretability of the topics, we identified models that balanced thematic diversity with clarity. The chosen models capture distinct, meaningful themes that reflect the core subjects of each category, ensuring a coherent and effective topic modeling structure.

6 Conclusion

Our project has demonstrated the effectiveness of applying text mining techniques to analyze and interpret news articles. By employing structured techniques, including text preprocessing, classification, and topic modeling, we transformed unstructured textual data into valuable insights. This workflow allowed us to categorize news articles accurately and uncover underlying themes, emphasizing the power of these methods in handling real-world datasets.

The results of our analysis were promising, with classification models performing well in organizing news articles into distinct categories. Advanced approaches, such as the use of BERT embeddings, showcased significant improvements over traditional methods, highlighting the advantages of leveraging modern NLP techniques. Topic modeling further provided a deeper understanding of thematic structures, offering nuanced insights within the dataset

In summary, each stage of the text mining process was essential in transforming raw data into actionable insights. This project reaffirms that well-structured text mining workflows are not only powerful tools for understanding textual data but also crucial for tackling complex analytical tasks, such as the systematic analysis of news content.