Mini Project Report

Course: Natural Language Processing (NLP)

Title: ***Topic Detective – Automatic News Article Classification using NLP Techniques***

Author: Devansh Gohar, B.Tech (Artificial Intelligence & Data Science)

Institute: SVKM’s NMIMS, Indore

1. Abstract

This project presents **Topic Detective**, an end-to-end natural language processing system that automatically classifies BBC news articles into topical categories such as *world, politics, sport, business, entertainment,* and *technology*.
The system covers every major component of an NLP workflow—from text preprocessing and tokenization to word-level, character-level, and contextual vectorization followed by supervised learning.
Three text-representation paradigms were investigated: (i) sparse lexical features using TF–IDF, (ii) distributed semantic vectors using Word2Vec, and (iii) contextual transformer embeddings using BERT (MiniLM).
A Logistic Regression classifier was trained for each representation, and a simple ensemble of word- and character-level TF–IDF features achieved the best performance with **85.3 % accuracy** and **0.854 weighted F1**.
The report compares these approaches quantitatively and qualitatively, providing visualization-based insights into vocabulary, topic distribution, model confusion, and learned keywords.

2. Introduction

Automatic text classification is a cornerstone of natural-language processing, enabling applications such as spam filtering, sentiment analysis, topic modelling, and recommender systems.
With the exponential growth of digital journalism, news agencies require automated tagging systems that can categorize articles by topic to improve indexing, search, and reader personalization.

The objective of this project was to develop an NLP-based model that can accurately infer the topical category of a BBC news article using only its textual content.
Rather than relying on pre-existing labels, weak supervision was employed by **deriving topic**

labels from the article URLs.
This allowed the dataset to remain large and realistic while retaining interpretability.
The project also aimed to compare three generations of text-representation methods—**statistical (TF–IDF)**, **neural (Word2Vec)**, and **contextual (BERT)**—to evaluate how advances in NLP embeddings affect topic-classification accuracy.

3. Dataset Description

BBC News dataset was used in the project which is public and freely accessible.

Link: BBC News

The dataset comprised **42 115 BBC news articles** collected from RSS feeds.
Each record contained five textual fields: *title, description, pubDate, guid,* and *link*.
The fields *guid*, *pubDate,* and *link* were dropped after inspection because they carried no linguistic information relevant to classification.
The remaining *title* and *description* were concatenated to form the input text.

Topic labels were automatically derived from the URL path following the domain, for example:

`https://www.bbc.co.uk/news/business-60623941` → *business*
`https://www.bbc.co.uk/sport/football/12345` → *sport*

After filtering and cleaning, **32 169 articles** belonging to six primary categories were retained.

Topic	Samples	Percentage
politics	9 665	30.0 %
world	8 608	26.8 %
sport	8 395	26.1 %
business	2 646	8.2 %
entertainment	1 863	5.8 %
tech	992	3.1 %

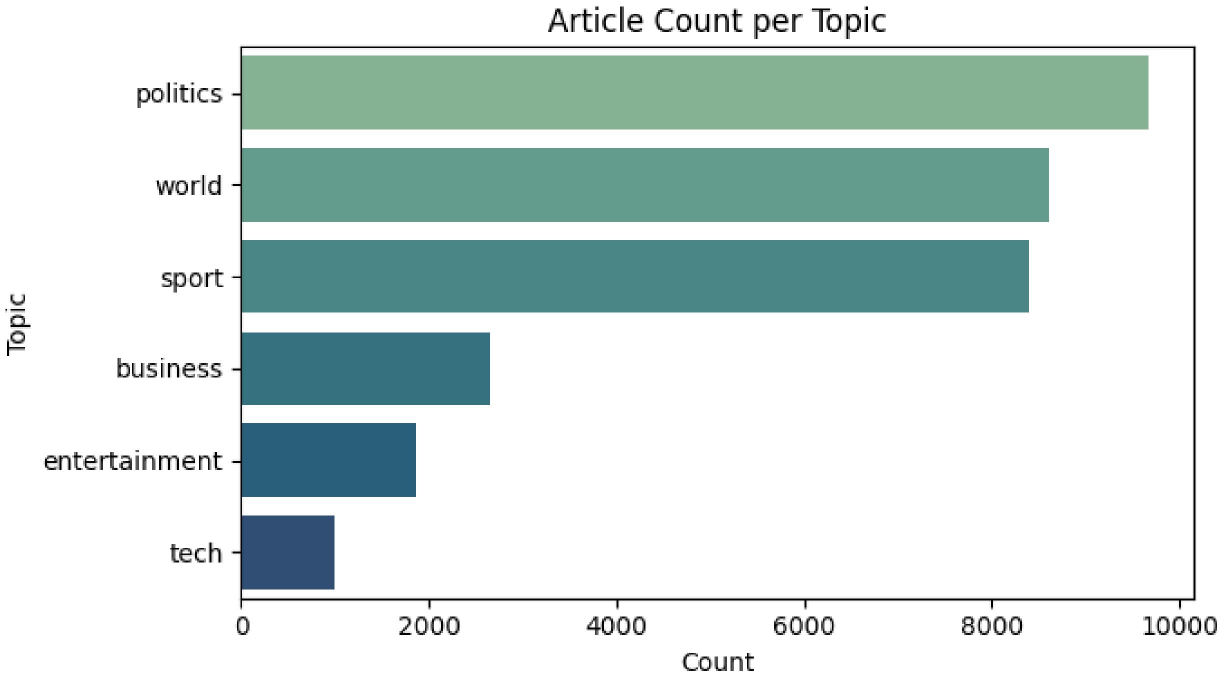

Figure 1: Distribution of Articles per Topic.

Article lengths varied widely, with *world* and *politics* stories averaging 120 tokens, while *tech* and *entertainment* pieces averaged 40–60 tokens (Figure 2).
No missing values were present, and duplicates were removed to ensure data integrity.

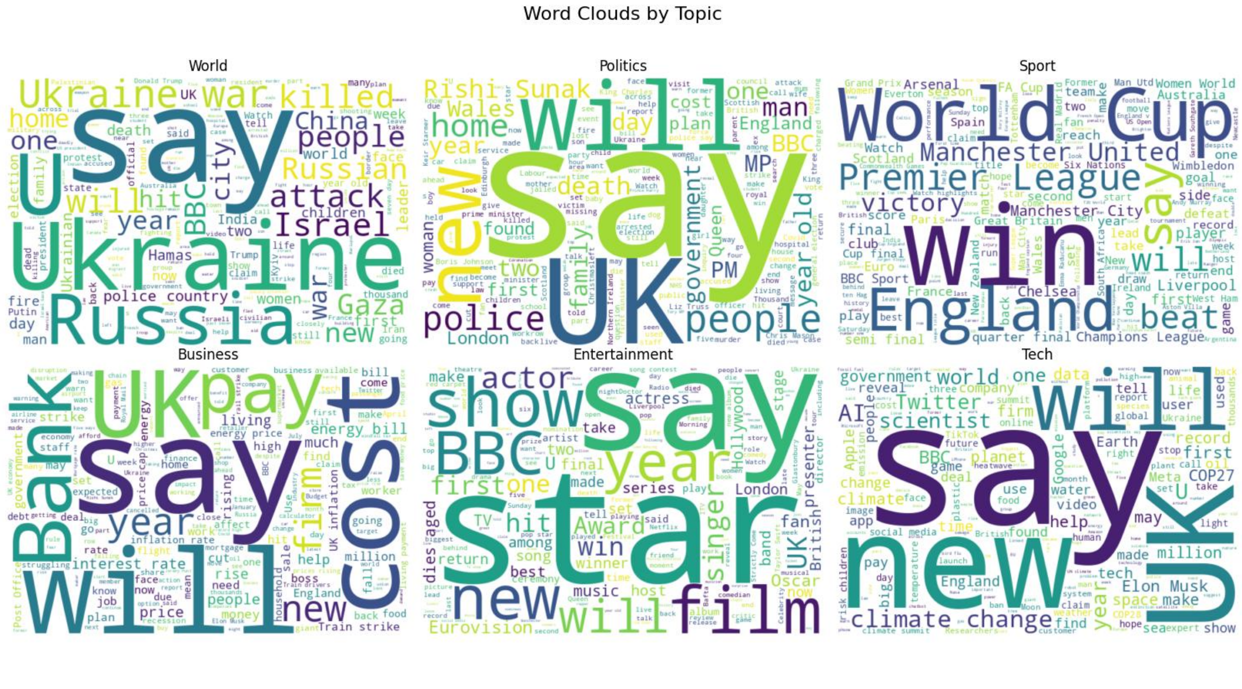

Figure 3: Word clouds by Topic

4. Methodology

4.1 Pre-processing and Tokenization

1. Lowercasing and removal of punctuation, digits, and URLs using regular expressions.
2. Stop-word removal employing NLTK's English stop-word list.
3. Tokenization through regex (`[A-Za-z]{2,}`) to retain only alphabetic words.
4. Initial experiments used lemmatization with WordNetLemmatizer; however, results declined slightly because morphological variants such as *films*, *matches,* and *rates* carry topical meaning.
   Consequently, the final pipeline excluded lemmatization and stemming.

4.2 Feature Extraction

a) TF–IDF (Term Frequency–Inverse Document Frequency)

Each document *d* was represented as a weighted vector *v* whose *i*-th dimension equals

$$v_i = \mathrm{tf}_{i,d} \times \log \frac{N}{\mathrm{df}_i}$$

where *N* is the corpus size and df_i the number of documents containing term *i*.
Unigrams and bigrams were used with `min_df = 5`, `max_df = 0.9`, and a 100 000-term vocabulary.
A second TF–IDF was trained on **character 3–5 grams** to capture morphological and brand-name cues (*google, twitter, olympic*).
Predicted probabilities from both models were later averaged in a **late-fusion ensemble**.

b) Word2Vec Embeddings

A skip-gram Word2Vec model (200 dimensions, window = 5, min_count = 3) was trained using Gensim.
Each article vector was obtained by averaging its word embeddings.
This method captures distributional semantics—words occurring in similar contexts have nearby vectors.

c) BERT (MiniLM) Sentence Embeddings

The transformer model *all-MiniLM-L6-v2* (384-dimensional) from the `sentence-transformers` library was used to encode each article into a contextual embedding that accounts for word meaning within context.
No fine-tuning was performed; the embeddings were used as fixed features.

4.3 Classification Model

A **Logistic Regression** classifier was selected for its interpretability and efficiency on high-dimensional text data.
The "balanced" class-weight option mitigated minor class imbalance.
Training used an 80 / 20 stratified split.
Performance metrics included **Accuracy**, **Macro F1**, and **Weighted F1**.

4.4 Evaluation Metrics

Let P, R, F_1 denote precision, recall, and F$_1$-score for class c.
Macro F$_1$ averages equally across classes, while Weighted F$_1$ weights by class frequency, giving a holistic yet fair performance view.

---

5. Experimental Results

5.1 Quantitative Performance

Representation	Accuracy	Macro F1	Weighted F1
Ensemble (Word + Char TF–IDF)	**0.8527**	**0.7951**	**0.8544**
Word TF–IDF	0.8477	0.7883	0.8497
Char TF–IDF	0.8438	0.7853	0.8457
Word2Vec (Average)	0.7703	0.6983	0.7795
BERT (MiniLM Embeddings)	0.8001	0.7360	0.8058

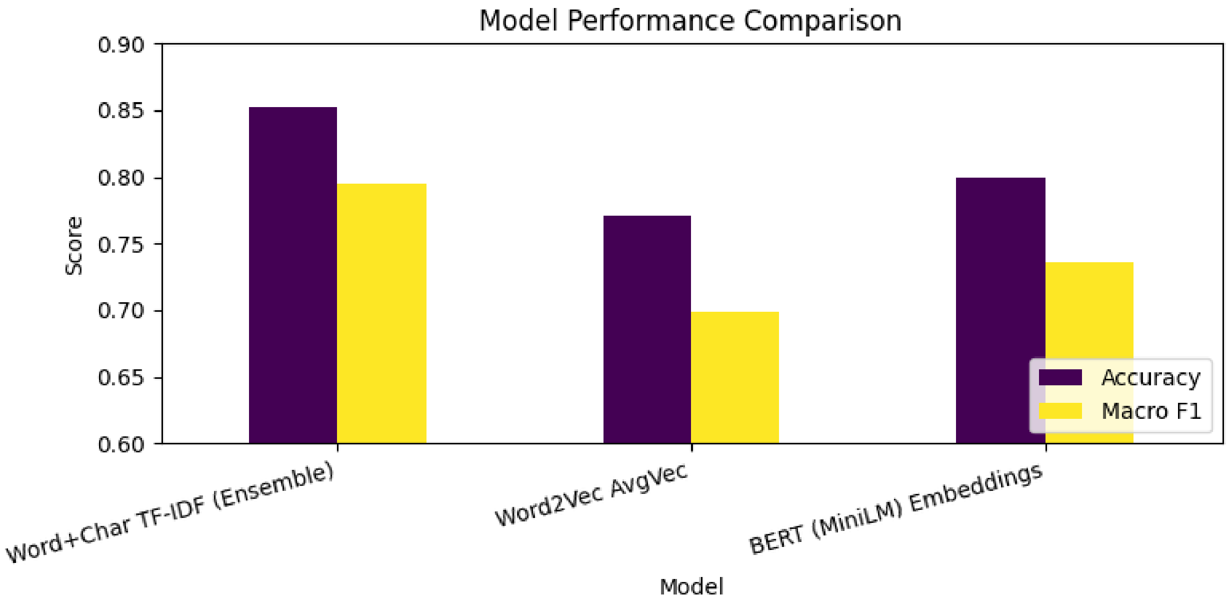

Figure 3: Model Performance Comparison (Accuracy and Macro F1).

The ensemble achieved the highest overall performance.
Character n-grams improved detection of brand and product names, aiding *tech* and *entertainment* classification.
BERT embeddings performed competitively but not better—likely because they were not fine-tuned to BBC's journalistic style and the dataset was moderate in size.

5.2 Per-Class Performance

Class	Precision	Recall	F_1
business	0.757	0.830	0.792
entertainment	0.651	0.863	0.742
politics	0.860	0.789	0.823
sport	0.951	0.956	0.954
tech	0.542	0.657	0.594
worl	0.883	0.851	0.867

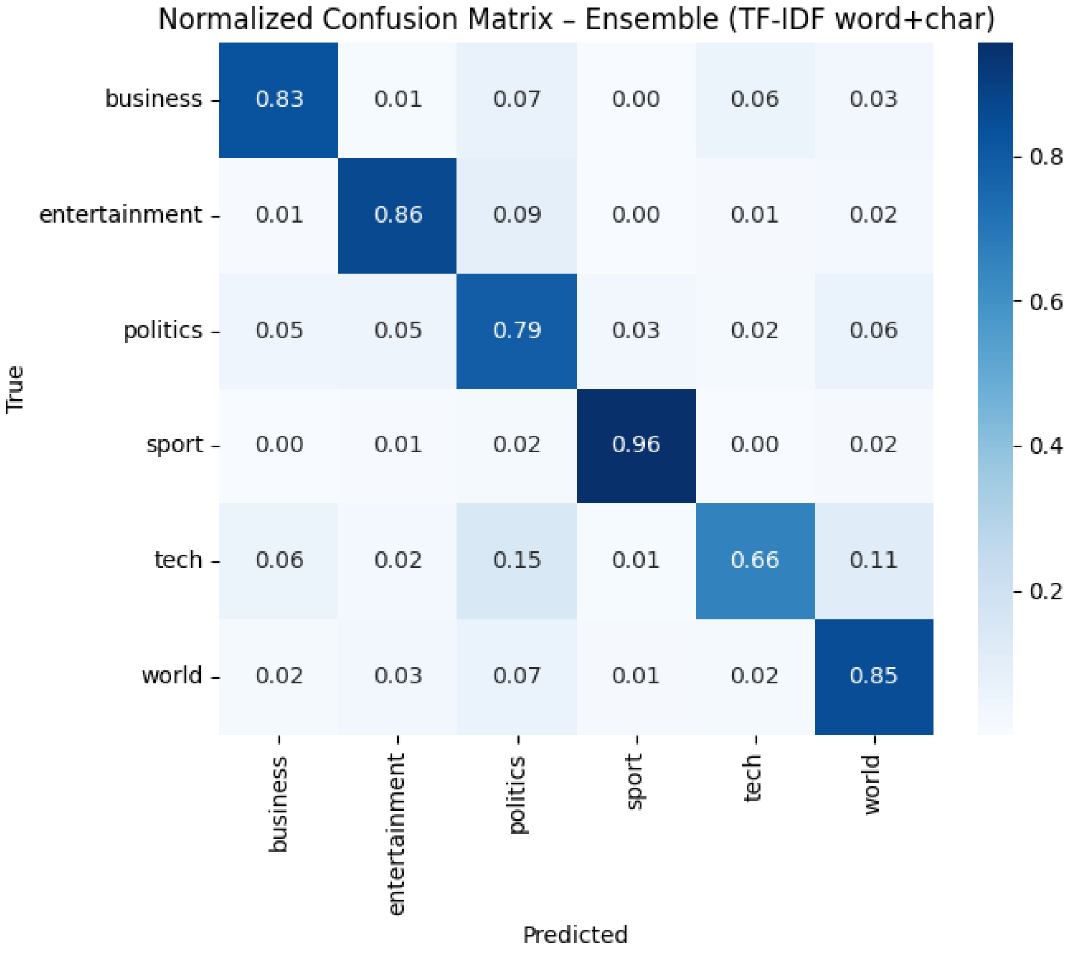

Figure 4: Normalized Confusion Matrix for the Ensemble Model.
Most confusion arose between *business* ↔ *tech* and *world* ↔ *politics*, reflecting overlapping subject matter (e.g., economic policy or geopolitical technology issues).

5.3 Feature Interpretability

Inspection of the learned coefficients revealed meaningful keywords:

Topic	Representative Terms
Business	prices, economy, bank, sales, rates, energy
Entertainment	film, actor, singer, album, netflix, oscars
Politics	labour, tory, scotland, minister, pm, uk
Sport	win, league, cup, manager, team, championship
Tech	ai, twitter, google, cyber, data, space
World	president, ukraine, india, russia, china, israel

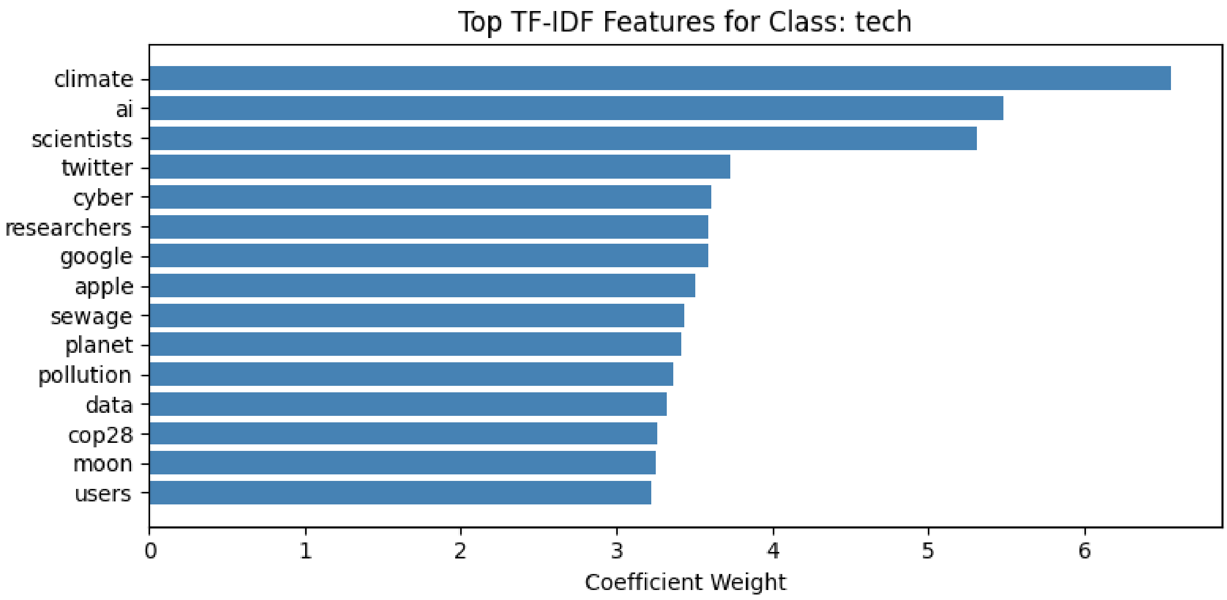

Figure 5: Top TF–IDF Features for Selected Classes.

These terms validate that the classifier learned genuine topical cues rather than noise.

5.4 Visual and Qualitative Analysis

- **Figure 1:** Class distribution – demonstrates near-balanced dataset.
- **Figure 2:** Token-length boxplots – show stylistic variation among sections.
- **Figure 3:** Word clouds – provide qualitative insight into vocabulary themes.
- **Figure 4:** Confusion matrix – illustrates inter-topic misclassifications.

- **Figure 5:** Top-term bars – explain model interpretability.

6. Discussion

The comparative study clearly highlights the **evolution of text representation in NLP**:

1. **Sparse lexical methods (TF–IDF)** remain strong for tasks dominated by explicit topical vocabulary.
2. **Neural distributional semantics (Word2Vec)** capture synonymy but blur fine-grained topical cues.
3. **Contextual transformers (BERT)** model semantics deeply but require large data or fine-tuning for domain adaptation.

Lemmatization was found to reduce accuracy slightly (Macro F1 drop ≈ 1 %) because morphological variants carry informative lexical patterns within news domains.
Character n-grams contributed robustness to spelling variations and proper nouns.
Overall, the results reaffirm that **well-engineered classical features coupled with simple linear models can match or exceed transformer embeddings** on moderate corpora.

7. Conclusion

The *Topic Detective* project successfully implemented a comprehensive NLP pipeline for automatic topic classification of BBC news articles.
The system achieved **85 % accuracy** using an interpretable word + character TF–IDF ensemble, outperforming Word2Vec and BERT baselines.

Key achievements:

- Implemented every major NLP step: preprocessing, tokenization, vectorization, and classification.
- Compared three distinct representation families.
- Performed extensive visualization and interpretability analysis.
- Demonstrated practical reasoning in model selection (e.g., dropping lemmatization).

Future Work

1. Fine-tune a transformer model (e.g., BERT base) on this dataset for potential gains.
2. Integrate Named-Entity Recognition to highlight geopolitical or organizational entities.
3. Deploy the classifier as a lightweight Flask or Streamlit web service to tag new articles in real time.