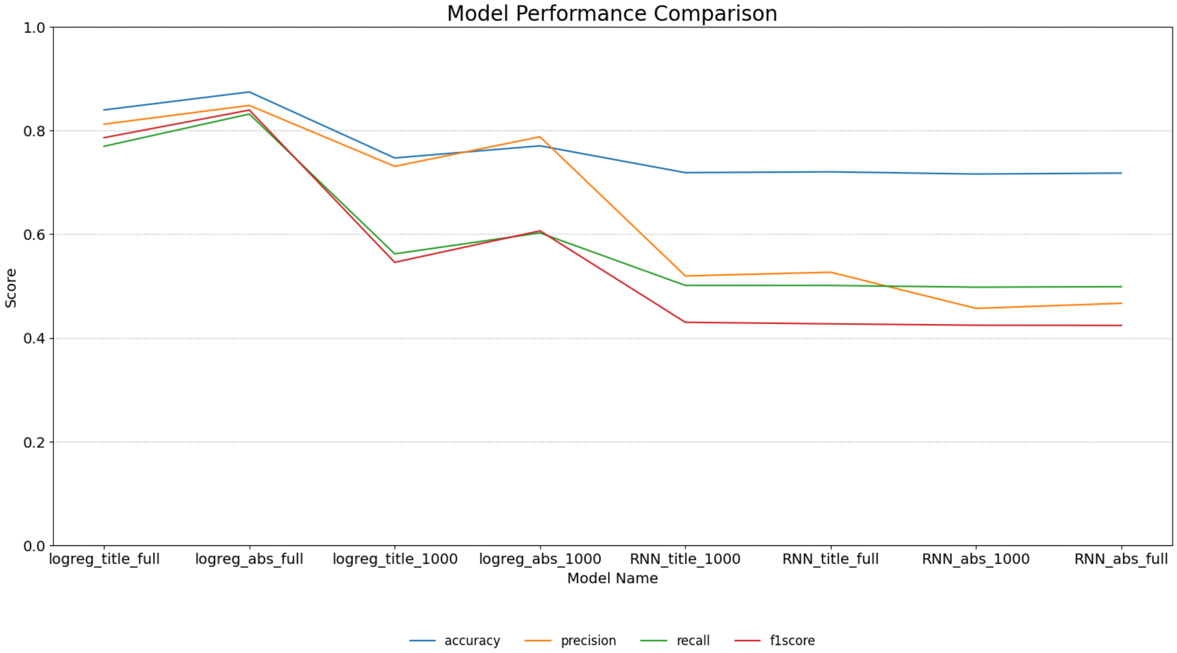
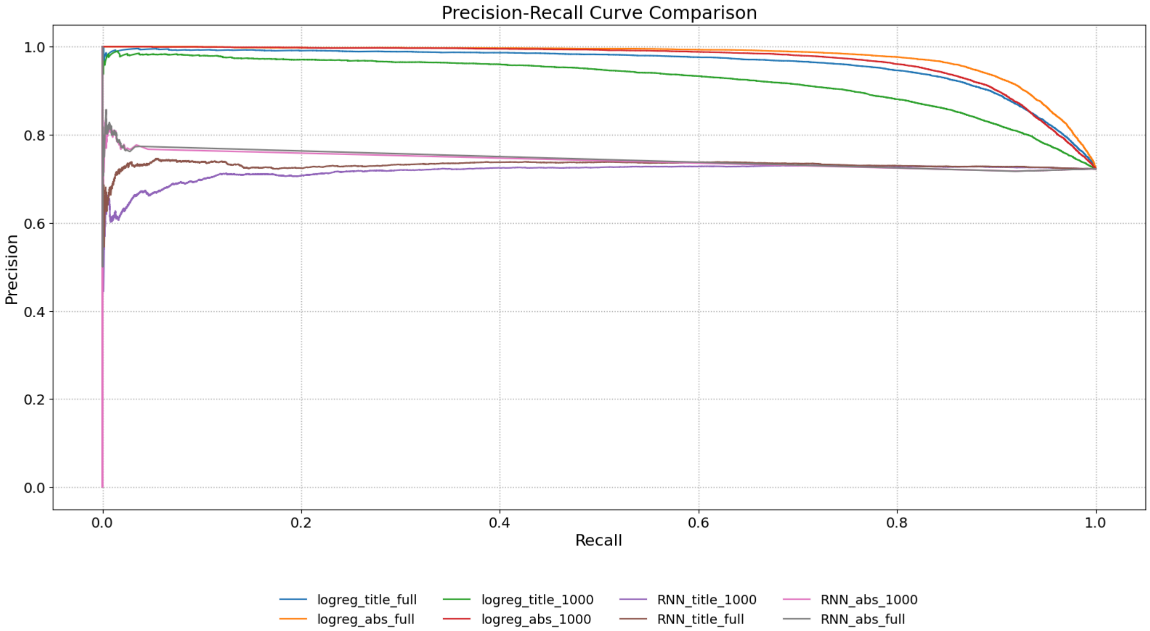
Part 1: Text Classification: *Due to Google Colab's runtime limits, I used only 10% of the original training data. In this context, "full training data" refers to this 10% subset. Note that only the training set was used for logistic regression, unlike the tutorial which combined training and dev data.*





How well did the two algorithms work under different data size conditions, when and why?

The performance gap between models trained on 1,000 samples and those trained on the full dataset is **significant**, especially for **RNN models**.

- **Logistic Regression (LogReg)** shows moderate performance drop when using only 1,000 samples compared to the full dataset, but still maintains **reasonable accuracy (≈75–78%)** and F1-scores above **0.55**.

- In contrast, **RNN models suffer severe degradation**, with all precision, recall, and F1-scores falling below 0.6—even when accuracy remains moderately stable around 75%. This indicates that RNNs struggle to capture meaningful patterns when data is scarce.

Insight: RNNs may benefit significantly from larger datasets with mindful vocab selection, while Logistic Regression is more resilient under low-resource conditions like first 1000 and full training data.

How did the model trained on title compare with the one trained on the abstracts?

Models trained on abstracts consistently outperformed those trained on titles across all metrics. Abstracts provided richer and more informative content, allowing both Logistic Regression and RNN models to achieve higher precision, recall, and F1-scores. This performance gap was evident regardless of the training size or algorithm used. In contrast, title-based models struggled due to limited context, leading to lower classification effectiveness.

Insight: Abstracts offer greater semantic depth, making them a more effective input for text classification.

What insights do the various metrics and plots give you?

1. Overall Performance Metrics (first plot):

- **Logistic Regression with Full Data** (*logreg_title_full*, *logreg_abs_full*): Strongest performers across all metrics (accuracy ≈ 0.85+, precision/recall/f1 ≈ 0.8–0.85).
- **Logistic Regression with 1000 Samples**: Noticeable drop, especially with *logreg_title_1000*, while *logreg_abs_1000* maintains better performance than the title version, which indicates abstracts seem more informative.
- **RNN Models (Title or Abstract, 1000 or Full)**: Poor performance across the board. F1-scores are below **0.6** even with full datasets. Indicates RNNs didn't generalize well or were undertrained compared with logistic regression

Key takeaway: Logistic Regression is more robust, especially in low-data scenarios. Abstracts are generally more informative than titles in both algorithm

2. Precision-Recall Curve (second plot): Curves closer to the top-right indicate better precision-recall

- *logreg_title_full* and *logreg_abs_full* have the best curves — confirming their strength from the first plot. *logreg_title_1000* and *logreg_abs_1000* drop but still outperform all RNN variants.
- **RNN curves** (especially *RNN_abs_1000*, *RNN_title_1000*) show early steep drops in precision — they misclassify too many positives.

Key takeaway:

- RNN models fail to sustain precision as recall increases, indicating overfitting or training instability. Logistic Regression models maintain strong and consistent precision-recall trade-offs.

Part 2: Topic Modeling Analysis and Interpretation

1. Introduction

This report explores topic modeling using LDA on a corpus of research article abstracts. The objective was to uncover hidden thematic groupings across articles and evaluate how well the LDA model captures meaningful topics. Two configurations were tested: unigram-only and bigram-enhanced tokenization with custom text-processing and document appearance threshold shown in the ipynb file, each applied to datasets of 1,000 and 20,000 articles.

2. Topic Groupings Identified

The models revealed recurring and interpretable topics across all configurations. Some of the most prominent themes include:

- **Machine Learning & Algorithms**: Topics featured keywords such as *model*, *learning*, *training*, and *algorithm*, indicating a strong cluster around machine learning methodologies.
- **Natural Language Processing (NLP)**: Frequent terms included *language*, *semantic*, *sentence*, and *representation*, suggesting a clear NLP-focused grouping.
- **Human-Computer Interaction (HCI)**: Topics highlighting *interaction*, *user*, *interface*, and *design* aligned with research in HCI.
- **Education & Learning Technologies**: Terms like *students*, *teaching*, and *educational* suggested discussions around technology in education.
- **Data Science & Analysis**: Including words like *data*, *analysis*, and *statistical*, often appearing in general-purpose or cross-cutting topics.

These groupings were coherent and aligned with the expected disciplines in the dataset. For instance:

- The article starts with **“Counterfactual (CF) explanations for machine learning models”** exemplifies the machine learning theme.
- **“How can multiple humans interact with multiple robots?”** aligns with the HCI topic.
- Another article discussing **“We study the concentration of random kernel matrices around their mean”** reflects the NLP theme.

Most topics contained well-formed and domain-relevant word clusters, especially when bigrams were used (e.g., *natural_language*, *user_interface*, *deep_learning*), enhancing interpretability.

3. Evaluation of Topic Modeling Effectiveness

The LDA models proved effective in identifying dominant themes within the corpus, though topic quality varied with configuration and dataset size.

Strengths:

- **Topic Coherence**: Most configurations (especially with 20,000 articles) yielded semantically coherent clusters.

- **Improved Clarity with Bigrams**: Bigrams captured domain-specific expressions (e.g., *large_language*, *deep_learning*), reducing ambiguity and increasing interpretability.
- **Depth from Abstracts**: Using abstract text provided richer contextual information than titles would have, enabling better topic separation.

Limitations:

- **Generic Overlap**: Unigram models often included broad terms like *data*, *system*, and *model* across multiple topics, reducing boundary clarity.
- **Noise in Smaller Datasets**: The 1,000-document datasets led to fragmented or vague topics due to limited vocabulary diversity and co-occurrence strength.

Despite these issues, LDA served its knowledge discovery purpose well—surfacing hidden structures and thematic divisions in academic literature, especially when bigrams and larger datasets were used.

4. Configuration and Dataset Size Comparison

Configuration	**Observations**
1000 articles, unigrams	Topics were general and lacked depth. Many top words were common across domains (e.g., data, system), leading to overlapping themes. The model struggled to distinguish nuanced research areas due to the small corpus size.
1000 articles, bigrams	Introducing bigrams brought modest improvements compared to that of unigrams, surfacing terms like *neural_network* and *user_interface*. However, due to the small dataset, some topics remained underdeveloped or inconsistent.
20,000 articles, unigrams	The expanded dataset allowed for more granular and diverse topic distributions. However, the lack of bigram context meant some topics still suffered from overlapping high-frequency unigrams (e.g., learning, model).
20,000 articles, bigrams	This setup produced the most coherent and specific topics. Well-separated clusters emerged around core concepts like natural_language, deep_learning, policy_learning, and system_architecture. Topics were aligned with distinct academic fields, reflecting a deeper semantic structure.

Summary Insight:

The 20,000-article bigram configuration clearly outperformed others, producing well-separated, domain-specific topics with minimal overlap. Salient terms like natural_language and deep_learning reflected meaningful research areas. In contrast, the 1,000-article models—particularly with unigrams—yielded vague, overlapping topics. These results highlight the importance of both data scale and bigram tokenization for effective topic modeling and I suggest potential for further exploration using n-grams or alternative modeling approaches.