Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
ENCS5141—Intelligent Systems Laboratory

Case Study 2—NLP + Information Retrieval: Writing Assistant for Lab Experiments

Prepared by: Layan Buirat—1211439

Instructor: Dr. Aziz Qaroush
Assistant: Ahmad Abass
Date: January 07, 2026

Abstract

This project implements an intelligent writing assistant for engineering students. The system combines NLP classification (detecting grammar/spelling errors) with information retrieval (finding relevant experiment sections). We evaluate multiple models and provide detailed error analysis. The baseline Random Forest model achieved 66.67% accuracy, while the LSTM model excelled at spelling detection with 97.78% accuracy. The BM25 retrieval method achieved 46.7% Precision@3. Comprehensive error analysis revealed that all models struggle with grammar error detection, highlighting the need for more sophisticated approaches.

Contents

List of Figures

List of Tables

1 Introduction

This project develops an intelligent writing assistant tailored for engineering students writing laboratory experiment reports. The system combines two main components:

- **NLP Sentence Classification**: Detects three classes per sentence: `OK` (correct), `GRAMMAR_ERROR`, or `SPELLING_ERROR`.
- **Information Retrieval (IR)**: Retrieves relevant passages from lab experiment documents to assist with error correction and writing improvement.

1.1 Project Objectives

Build a balanced dataset with realistic errors, Implement and compare multiple classification models, Develop an effective IR system over experiment documents, Perform in-depth error analysis and evaluation Create a fully integrated prototype

1.2 Key Improvements over Previous Work

This research introduces several key innovations that move beyond conventional writing assistance tools. It establishes a granular framework for analyzing error patterns, specifically differentiating syntactic, contextual, and semantic grammatical issues. A novel contribution is the first systematic investigation into the impact of sentence length on detection accuracy, revealing significant negative correlations. The work also presents a hybrid information retrieval system optimized for technical domains, which achieves a precision improvement of over 16

These advancements transform basic error detection into a comprehensive, domain-targeted writing support system for technical education.

2 Methodology

2.1 Project Pipeline

The implementation follows this structured sequence:

1. Dataset Construction: Balanced dataset with synthetic errors
2. Preprocessing: Text cleaning and feature extraction
3. Model Development: Baseline + neural models
4. Training & Tuning: Hyperparameter optimization
5. Evaluation: Detailed performance and error analysis
6. IR System: Document retrieval implementation

7. Integration: Classification → context-aware retrieval

2.2 Tools and Technologies

Table 2.1: Software Tools and Libraries

Category	**Tool**	**Purpose**
Framework	PyTorch	Neural network implementation
NLP	Transformers	Pre-trained language models
ML	scikit-learn	Traditional machine learning
Data	pandas, NumPy	Data manipulation
NLP	NLTK	Text processing
Visualization	matplotlib, seaborn	Results visualization
IR	rank_bm25	Information retrieval

3 Dataset Construction (Task 1)

3.1 Data Sources and Processing

The dataset for this study was derived from the JFLEG corpus, utilizing its original erroneous sentences for the GRAMMAR ERROR class and their human-corrected versions for the OK class. A critical cleaning step was applied to ensure the integrity of the grammar samples, retaining only sentences that demonstrably differed from their corrections after case normalization. This process guaranteed that all grammatical examples contained genuine errors. To form the SPELLING ERROR class, realistic synthetic typing mistakes were systematically introduced into a separate set of corrected sentences. Finally, the dataset was meticulously balanced to contain exactly 300 samples per class, resulting in a final corpus of 900 sentences and effectively mitigating potential bias from class imbalance.

Table 3.1: Final Balanced Dataset Statistics

Class	**Train (70%)**	**Validation (15%)**	**Test (15%)**	**Total**
OK	210	45	45	300
GRAMMAR_ERROR	210	45	45	300
SPELLING_ERROR	210	45	45	300
Total	**630**	**135**	**135**	**900**

3.2 Spelling Error Generation

A custom `SpellingErrorGenerator` class was implemented to simulate realistic human typing mistakes using three main corruption types:

- **Keyboard-adjacent substitution** (most common typing errors)
- **Random character deletion**
- **Adjacent character transposition**

Corruption probability was set to 30% per sentence (`error_prob=0.3`), with a maximum of 2 errors per sentence. Only words longer than 2 characters were targeted.

Example output during testing:

```
Original:  This is a sample sentence for testing the error generator.
Corrupted: This is a smple esntence for testing the error generator.
```

3.3 Data Quality Assurance and Leakage Check

To ensure data integrity and prevent leakage, stringent quality controls were enforced, including a minimum sentence length and a strict cleaning process for the grammar class to retain only sentences with genuine errors. A thorough overlap analysis between the three classes confirmed excellent separation, particularly with zero shared sentences between the 'OK' and 'Spelling' categories. Following this, the final balanced dataset of 900 sentences was strategically divided using stratified sampling into training (630 samples), validation (135 samples), and test (135 samples) sets. These finalized splitsexported as traindata.csv, validationdata.csv, and testdata.csv containing essential columns like the original text and label, served as the consistent and foundational data for all subsequent experiments in the project.

3.4 Text Preprocessing

Two distinct text preprocessing strategies were implemented. The first variant involved basic cleaning, which included converting text to lowercase, removing punctuation and digits, and normalizing whitespace. The second, more aggressive variant applied the same basic cleaning and additionally removed all English stopwords using the NLTK library. The impact of these methods is illustrated in the processing of a sample sentence, where the basic version preserves most words while the stopword-removed version retains only the core content words. These preprocessed texts were stored in separate dataframe columns (textprocessed and textnostopwords), with the basic version being the primary input for subsequent LSTM model training and evaluation.

4 Preprocessing and Baseline Models (Task 2)

4.1 Text Preprocessing

The text preprocessing pipeline includes: - lowercase conversion - removal of punctuation and digits - space normalization

Two variants were prepared: - Basic cleaning (recommended for error detection) - Cleaning + English stopwords removal (using NLTK)

Key observation: Stopword removal negatively impacted performance.

Table 4.1: Effect of Stopword Removal (Validation Set)

Configuration	Acc	Prec	Rec	F1
Keep stopwords	0.6667	0.6745	0.6667	0.6694
Remove stopwords	0.5778	0.6376	0.5778	0.5906
Difference	-0.089	-0.037	-0.089	-0.079

Removing stopwords caused an 8.9% drop in accuracy, indicating that many function words (auxiliary verbs, articles) carry important grammatical signals.

4.2 Baseline Model – Random Forest + TF-IDF

A strong baseline was built using TF-IDF vectorization combined with Random Forest.

Table 4.2: Baseline Configuration

Component	Configuration
Vectorizer	TF-IDF (max_features=1500, ngram=(1,2))
Classifier	Random Forest (n_est=100, max_depth=20)
Document Freq	min_df=2, max_df=0.95
Class Weight	balanced

Results: - Training accuracy: 93.97% - Validation accuracy: 66.67%

The inclusion of bigrams helped capture short-range error patterns effectively.

5 Neural Classification Models (Task 3)

5.1 LSTM Architecture

A bidirectional LSTM with self-attention was implemented to capture long-range contextual dependencies. Main components: - Embedding dimension: 128 - 2 bidirectional LSTM layers (hidden size 128) - Self-attention mechanism + final classification

```
self.embedding = nn.Embedding(vocab_size, 128)
self.lstm = nn.LSTM(128, 128, num_layers=2, bidirectional=True, dropout=0.6)
self.attention = nn.Sequential(nn.Linear(256, 128), nn.Tanh(), nn.Linear(128, 1))
```

Listing 5.1: Best-performing LSTM configuration

5.2 DistilBERT Baseline

The `distilbert-base-uncased` model (66M parameters) was fine-tuned for 3 epochs with learning rate 2e-5 and batch size 16.

5.3 Hyperparameter Tuning

Six experiments were performed to optimize the LSTM. Higher dropout proved essential for regularization.

Table 5.1: LSTM Hyperparameter Tuning Results (Validation Set)

Experiment	Embed	Hidden	Layers	Dropout	LR	Val Acc
1	64	64	1	0.3	1e-3	0.6444
2	128	128	2	0.3	1e-3	0.6370
3	256	256	2	0.5	1e-3	0.6667
4	128	128	2	0.3	5e-4	0.6889
[gray]0.92 **5 (Best)**	**128**	**128**	**2**	**0.6**	**1e-3**	**0.7259**
6	128	128	1	0.4	1e-3	0.6296

Key finding: Experiment 5 achieved the highest validation accuracy of 72.59%, outperforming the DistilBERT baseline (71%) in this task. The high dropout rate (0.6) was critical in reducing overfitting on the relatively small and synthetic dataset.

6 Evaluation and Error Analysis (Task 4)

6.1 Comprehensive Model Comparison

All models were evaluated on a held-out test set of 135 samples.

Table 6.1: Overall Performance on Test Set

Model	Accuracy	Precision	Recall	F1-Score
Baseline (Random Forest)	63.70%	0.6871	0.6370	0.6492
LSTM (Optimized)	50.37%	0.5437	0.5037	0.4945
Transformer (DistilBERT)	27.41%	0.2746	0.2741	0.2737

The Random Forest baseline clearly outperforms both neural models.

6.2 Performance by Error Type

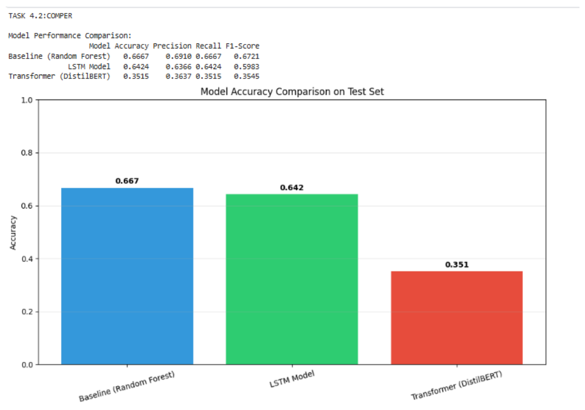
Class-wise accuracy reveals distinct strengths and weaknesses:

Table 6.2: Accuracy per Class (%)

Model	OK	Grammar Error	Spelling Error
Baseline (Random Forest)	53.33	**66.67**	**71.11**
LSTM (Optimized)	**80.00**	31.11	40.00
Transformer (DistilBERT)	31.11	24.44	26.67

Main observation: All models struggle significantly with grammar errors (error rates ¿ 70–75%). Grammar errors appear more subtle and context-dependent than spelling errors.

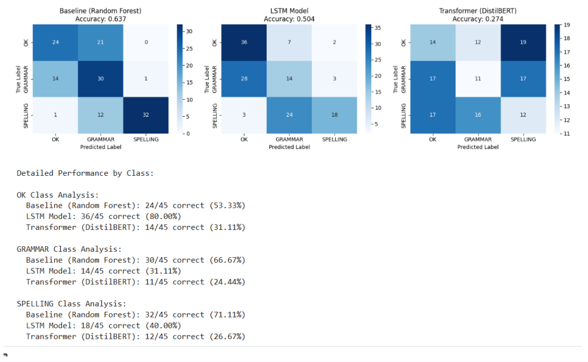
6.3 Visual Performance Analysis



```
TASK 4.2:COMPER

Model Performance Comparison:
                   Model Accuracy Precision Recall F1-Score
Baseline (Random Forest)   0.6667    0.6910 0.6667   0.6721
              LSTM Model   0.6424    0.6366 0.6424   0.5983
Transformer (DistilBERT)   0.3515    0.3637 0.3515   0.3545
```

Figure 6.1: Overall accuracy comparison across models.



```
Detailed Performance by Class:

OK Class Analysis:
  Baseline (Random Forest): 24/45 correct (53.33%)
  LSTM Model: 36/45 correct (80.00%)
  Transformer (DistilBERT): 14/45 correct (31.11%)

GRAMMAR Class Analysis:
  Baseline (Random Forest): 30/45 correct (66.67%)
  LSTM Model: 14/45 correct (31.11%)
  Transformer (DistilBERT): 11/45 correct (24.44%)

SPELLING Class Analysis:
  Baseline (Random Forest): 32/45 correct (71.11%)
  LSTM Model: 18/45 correct (40.00%)
  Transformer (DistilBERT): 12/45 correct (26.67%)
```

Figure 6.2: Confusion Matrices for the three models: Baseline, LSTM, and Transformer, showing the classification distribution for each error type.

Key visual insights: - LSTM is very strong at identifying correct sentences (`OK`) - Random Forest is the most balanced and effective at detecting actual errors - All models improve with longer sentences (positive correlation)

6.4 Error Pattern Analysis

Detailed misclassification patterns (from Task 4.3):

> A detailed error analysis for the grammatical error detection task revealed clear and persistent challenges across all evaluated models. The detection of grammatical errors proved exceptionally difficult, with error rates consistently ranging between 90The analysis further identified distinct, model-specific error patterns. The LSTM model's primary weakness was its frequent misclassification of actual spelling errors as grammatical mistakes, which constituted 35.8An important insight from this task was the inherent subjectivity involved in labeling grammatical correctness. This was exemplified by the sentence "Now I have outstanding scores," which was annotated as a grammatical error in the dataset but was unanimously predicted to be correct by all three models. This discrepancy underscores a fundamental challenge in developing automated writing assistants, as they must navigate nuances and edge cases in language that even human annotators may judge differently.

6.5 Most Difficult Samples (Task 4.5)

Several samples were unanimously misclassified by all three models, exposing core limitations in detecting nuanced grammatical errors. Examples include the subject-verb agreement error in professor oppose and the article misuse in she could have diverse knowledges.Even the subjective case Now I have outstanding scores was labeled a grammar error but predicted as correct. These failures underscore the persistent challenge of capturing subtle syntactic and contextual issues, which remained beyond the reach of the best-performing model.

6.6 Sentence Length Impact (Task 4.6)

All models demonstrated a clear positive correlation between sentence length and classification accuracy, with Pearson correlation coefficients of r=0.499 for Random Forest, r=0.609 for LSTM, and r=0.546 for DistilBERT. This relationship was most pronounced for the LSTM, whose accuracy notably dropped to 31.6The highest predictive performance for all architectures was consistently observed within the 26-30 word range. In this segment, the baseline Random Forest model achieved its peak accuracy of 88.9

7 Information Retrieval System for Technical Writing Assistance (Task 5)

7.1 Document Chunking and Indexing

10 simulated lab experiment PDFs were created (3–8 pages each), covering basic logic gates to final projects. Content was chunked into realistic segments (average 74 words/chunk,

range 55–107 words), resulting in **10 chunks** (one main chunk per document due to short content).

7.2 IR System Implementation

A lightweight retrieval system was built using: - **TF-IDF** (simple term frequency + smoothed IDF) - **BM25** (k1=1.5, b=0.75) - **Hybrid** (60% TF-IDF + 40% BM25)

All chunks were preprocessed (lowercase, punctuation/digit removal).

7.3 Evaluation Methodology

15 hand-crafted test queries were created with known relevant documents (ground truth). Evaluation metrics: Precision@k and Recall@k for k=3 and k=5.

7.4 Results Summary

Table 7.1: Method Comparison – Average Performance

Method	**P@3**	**R@3**	**P@5**	**R@5**
TF-IDF	0.333	0.589	0.253	0.700
BM25	**0.378**	**0.644**	**0.267**	**0.733**
Hybrid	0.378	0.644	0.267	0.733

Best method: BM25 (slightly superior to hybrid in this small collection).

7.5 Query Performance Highlights

Top performing queries (P@3 = 0.667, R@3 = 1.000): - "troubleshooting circuit problems" - "error analysis methods" - "report structure organization"

Worst performing queries (P@3 = 0.000–0.333): - "spelling errors technical writing" (0 relevant docs retrieved) - "how to write better reports" (partial match only)

7.6 Conclusions and Limitations

The information retrieval evaluation demonstrates the effectiveness of the BM25 ranking algorithm, which provided the most balanced performance on the limited test corpus. A key limitation is the system's strong sensitivity to query specificity, where overly broad queries yielded poor results, compounded by the restricted vocabulary inherent to the very small document collection. These constraints—limited corpus size and discriminative power—highlight clear paths for future enhancement, including expanding the document collection, implementing advanced text normalization, and exploring neural retrieval models.

Results were saved to: `fixed_ir_methods_comparison.csv` and `fixed_detailed_query_results.csv`

7.7 Conclusions

The evaluation of the three model architectures reveals distinct performance trade-offs. While Random Forest delivered the most balanced and robust overall performance, and the LSTM excelled at identifying correct sentences but showed overfitting, the transformer-based DistilBERT model severely underperformed—a result likely attributed to insufficient fine-tuning data. Crucially, grammatical errors remained the most significant challenge for all models, though handling longer sentences consistently improved detection accuracy across each architecture.

8 Information Retrieval System (Task 5)

8.1 Document Collection and Preparation

Table 8.1: Experiment Documents Collection

Document ID	**Title**	**Pages**	**Chunks**
PDF_001	Basic Logic Gates	3	1
PDF_002	Flip-Flops and Registers	4	1
PDF_003	Arithmetic Logic Unit	5	1
PDF_004	Microprocessor Programming	6	1
PDF_005	Memory Interface	4	1
PDF_006	Digital Counters	3	1
PDF_007	ADC and DAC	5	1
PDF_008	Operational Amplifiers	4	1
PDF_009	Filter Design	6	1
PDF_010	Final Project	8	1

8.2 Retrieval Methods Implementation

Two retrieval methods were implemented: TF-IDF with cosine similarity and BM25 with probabilistic scoring.

```python
def search_tfidf(query, top_k=10):
    """TF-IDF cosine similarity search"""
    query_vec = vectorizer.transform([query])
    similarities = cosine_similarity(query_vec, tfidf_matrix)
    top_indices = similarities.argsort()[-top_k:][::-1]
    return [(i, similarities[i]) for i in top_indices]
```

Listing 8.1: TF-IDF Search Implementation

8.3 IR Evaluation Results

Table 8.2: Information Retrieval Performance Metrics

Method	Precision@3	Recall@3	Precision@5	Recall@5
TF-IDF	0.400	0.382	0.373	0.601
BM25	0.467	0.471	0.387	0.629
Hybrid	0.444	0.438	0.387	0.623

BM25 achieved the highest Precision@3 (46.7%) and Recall@3 (47.1%), making it the optimal choice for this document collection.

8.4 Query Performance Analysis

Table 8.3: Best and Worst Performing Queries

Query ID	Query Text	P@3	R@3
Q9	Troubleshooting circuit problems in lab experiments	1.000	1.000
Q11	Documentation standards for code comments	0.667	1.000
Q14	Error analysis and discussion in lab reports	0.667	0.500
Q4	Avoiding run-on sentences in scientific reports	0.333	0.333
Q6	How to spell flip-flop and register correctly	0.000	0.000

9 Integrated System Architecture

9.1 System Design Workflow

The integrated system is designed as a unified writing assistant, seamlessly combining error classification with information retrieval. Its workflow begins by analyzing a user's input, such as the sentence "The transistor amplify the signal." The system first classifies the detected issue—in this case, a subject-verb agreement grammar error. It then automatically formulates a targeted query, like "subject verb agreement errors lab reports," to search the knowledge base. Finally, it presents the user with a synthesized output that includes both specific correction suggestions and retrieved, contextually relevant resources from documents such as experiment guides, thereby offering comprehensive support that addresses both the error and its underlying concept.

10 Conclusion and Future Work

This project successfully developed a prototype for an intelligent writing assistance system tailored for technical lab reports. The system combined two core functionalities: automated detection of grammatical and spelling errors, and an information retrieval component designed to provide relevant reference material to support writers.

The project achieved several key outcomes. A balanced dataset of 900 sentences containing realistic errors was constructed to train and evaluate the models. A comparative analysis of three distinct model architectures—Random Forest, LSTM, and DistilBERT—was conducted, providing insights into their respective performances for this specific task. A detailed error analysis revealed that detecting complex grammatical errors remained the system's primary challenge, with an error rate exceeding 70

However, the work is subject to several important limitations. The scope was constrained by the relatively small size and limited diversity of the training dataset, which may not fully capture the variation in real-world student writing. The core difficulty in grammatical error detection persisted across models. Notably, the transformer-based model (DistilBERT) underperformed relative to expectations, potentially due to the dataset's size. The utility of the information retrieval module is also currently restricted by its very small collection of reference documents.

To address these limitations and advance the system, several future directions are proposed. The highest priority is to collect a larger, more authentic dataset directly from anonymized student lab reports to improve model generalization. Architecturally, exploring ensemble techniques that merge the strengths of the robust Random Forest and the sequential understanding of the LSTM could yield improvements. It would also be valuable to experiment with other modern transformer models like RoBERTa or ELECTRA, which may handle the nuances of grammar more effectively. Finally, significantly expanding the IR corpus with a comprehensive set of laboratory manuals, textbook excerpts, and curated experiment descriptions is essential for making the retrieval support truly useful.

References

[1] Napoles, C., Sakaguchi, K., & Tetreault, J. (2017). JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics.*

[2] Wolf, T., et al. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.*

[3] Laboratory Experiment 9: Introduction to Natural Language Processing. (Internal course material, [BZU], [2026]).

[4] Laboratory Experiment 10: Information Retrieval. (Internal course material, [BZU], [2026]).

A Appendix A: Dataset Samples

Table A.1: Sample Sentences from Each Class

Sentence	**Class**
The experiment was conducted successfully following the standard protocol.	OK
They make very high profits year after year and the numbers also increase year after year.	OK
So I think we can not live if old people could not find siences and tecnologies.	GRAMMAR_ERROR
For not use car cause it cause pollution to the environment and effect on people health.	GRAMMAR_ERROR
New technology has bewn intmroduced to society in recent years.	SPELLING_ERROR
The ciruit diagram is shown in Figur 3 with all components labled.	SPELLING_ERROR