COMP 551 Assignment 4 Report

Author Jinghan Zheng, Yunjie Xiao, Tailai Li

Abstract

In this project, we compare a Long Short-Term Memory network (implemented from scratch) with a pretrained BERT model on the Web of Science (WOS-11967) scientific text classification tasks. Both models are evaluated on predicting the main scientific field (7 classes) and the sub-field (34 classes) of each abstract. We also examined attention matrices from a selected transformer block for both correctly and incorrectly classified documents. In the creativity section, we briefly tested a few LSTM tweaks to see how small architectural changes impact performance and compare them with performance of BERT. We mainly found that BERT clearly outperforms LSTM on the scientific text classification task.

Keywords: Text classification; LSTM; BERT; Web of Science dataset; Fine-tuning; Attention analysis; Transfer learning.

1 Introduction

Text classification is an important task in natural language processing, where the goal is to assign meaningful labels to written documents. The previous paper [2] proposes HDLTex. This hierarchical deep-learning framework stacks different neural architectures (RNN, CNN, DNN) to classify documents first into broad fields and then into specialized sub-fields. Using the Web of Science dataset, the authors show that HDLTex achieves significantly higher accuracy than traditional multi-class methods such as Naïve Bayes and SVM, especially as the number of classes grows.

Our tasks involve implementing an LSTM model from scratch, fine-tuning a pretrained BERT model, and comparing their performance across both classification levels. We also analyze attention patterns from BERT to understand why certain predictions succeed or fail. In the creativity section, we experimented with dropout removal, batch-norm removal, ReLU attention, and forget-gate bias initialization to see how architectural and training tweaks affect LSTM performance.

We apply both approaches to the Web of Science (WOS-11967) dataset [1], which is divided into datasets of various sizes and contains scientific abstracts labeled with a main field (7 classes) and a more detailed sub-field (34 classes). Our experiments show that BERT consistently outperforms all LSTM variants in both L1 and L2 classification, with attention analysis and model ablations confirming that pretrained transformers handle scientific text far more effectively than sequence-based models. Further experiments show that although LSTMs can be improved through architectural tuning, pretrained models like BERT remain substantially more effective for scientific text classification overall.

2 Datasets

The experiments use the Web of Science (WOS-11967) dataset [1], which contains 11,967 scientific abstracts paired with two levels of labels: a 7-class primary scientific field and a 34-class sub-field (derived from five subclasses under each L1 parent).

After loading, we split the dataset into 70% training (8,376 samples), 15% validation (1,795 samples), and 15% test (1,796 samples) using stratified sampling to preserve class balance.

For LSTM, we implemented word-level encoding: (1) tokenization via lowercasing and regex removal, (2) vocabulary filtering keeping only words with frequency ≥ 5, (3) word-to-index mapping using Python's `collections.Counter` to count word frequencies, indexed by insertion order, (4) conversion to fixed-length sequences of 200 integers with padding/truncation. We choose this encoding as the threshold of 5 removes noisy rare words while retaining sufficient examples for meaningful embeddings, improving generalization. A sequence length of 200 captures most text content while maintaining efficient memory usage for LSTM training. Word-level tokenization preserves order for sequential learning, and simple regex preprocessing avoids errors from language-specific rules. Moreover, using a direct index mapping (rather than subword models or hashing) provides constant-time lookup and minimal preprocessing overhead, reducing the total data-loading time and simplifying batching during training.

For the BERT model, we tokenized the text using the `bert-base-uncased` tokenizer from the Hugging Face library. Each text was converted into a fixed-length sequence of token IDs by applying truncation and padding to a maximum length of 256 tokens. The tokenization process also included replacing unknown words with a special `[UNK]` token. This ensured that all input sequences were of uniform length, facilitating efficient batch processing during training.

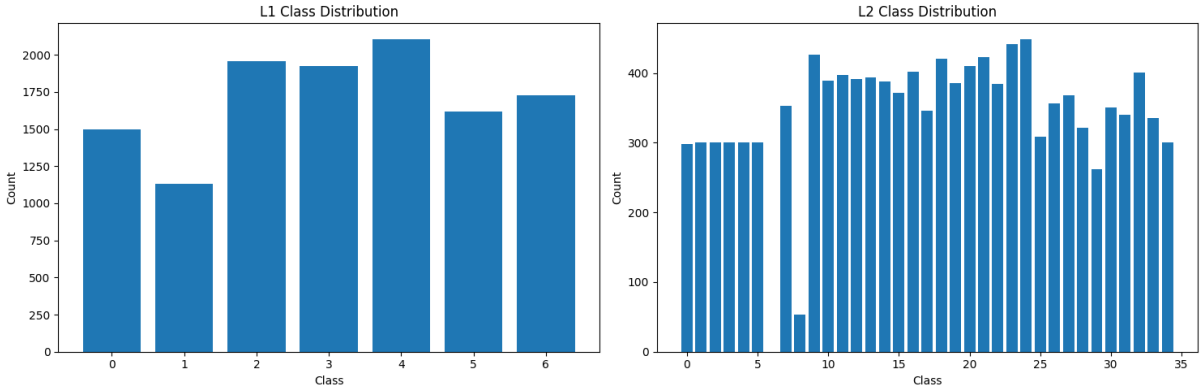


Figure 1: L1 and L2 class distributions in the WOS-11967 dataset.

To better understand the dataset, we examined the class distributions for both label levels. The L1 labels span seven classes with roughly 1,600–2,100 samples each, while the L2 labels cover five subclasses under each parent (34 classes total) with about 300–400 samples each. Both levels are fairly balanced, so no resampling or class weighting was needed. Overall, the dataset is well-structured and appropriate for testing classical and deep-learning text classifiers.

To create 34 unique classes, we map the L2 labels to new L2 labels by combining the L1 and L2 labels using the formula New L2 $=$ L1 $\times 5 +$ L2, where each L2 is a subclass of an L1 category.

3 Results

3.1 Performance of LSTM and BERT Model (Accuracy)

Model	Acc L1	Acc L2
LSTM	0.899777	0.761136
BERT	**0.932071**	**0.847439**

Table 1: Model performance on L1 and L2 classification tasks. Higher accuracies are bolded.

The results show a clear performance gap between the two approaches. The custom LSTM multi-task model achieved solid accuracy on both label levels (0.8998 for L1 and 0.7611 for L2), indicating that the scratch-built sequence model was able to learn meaningful patterns from the text despite using a relatively simple architecture and limited vocabulary-based embeddings. The fine-tuned BERT model, on the other hand, outperformed the LSTM model with accuracies of 0.9321 for L1 and 0.8474 for L2. This suggests that pretrained language models like BERT capture more linguistic features due to large-scale pretraining on diverse corpora and subword tokenization.

3.2 Attention Matrix Examination

We choose layer 11 and head 0's attention matrix for analysis. Using L1 labels, we determine correctness and select 2 samples that are correctly classified and 1 sample that is incorrectly classified by BERT. For each sample, we show the top 10 tokens most attended to by `[CLS]`:

Correctly Classified Samples

- **Example 1:** True L1 = 4, Predicted L1 = 4

Top tokens attended by [CLS]:

Idx	Token	Attn
10	pollution	0.1355
166	.	0.0616
58	.	0.0600
76	.	0.0585
185	.	0.0348
125	.	0.0338
9	water	0.0236
158	sediment	0.0194
52	nutrient	0.0191
114	exchange	0.0163

Decoded text (truncated):

```
[CLS] environmental conservation
and management policy first
emphasized on water pollution
control in japan. however, this
kind of passive conservation
policy is gradually being
shifted to an active approach
such as satoumi, which includes
the restoration of biodiversity,
biological productivity, ha
...[SEP]
```

Analysis: High attention on "pollution," with supporting focus on "water," "sediment," and "nutrient," aligns with terminology for Civil Engineering (L1=4), explaining the correct classification.

- **Example 3:** True L1 = 0, Predicted L1 = 0

Top tokens attended by [CLS]:

Idx	Token	Attn
219	instrumentation	0.3368
218	optical	0.0446
209	.	0.0383
36	.	0.0372
71	.	0.0361
172	.	0.0352
130	.	0.0296
99	.	0.0293
63	quantum	0.0250
50	quantum	0.0223

Decoded text (truncated):

```
[CLS] we explain how to share
photons between two distant
parties using concatenated
entanglement swapping and assess
performance according to the two
- photon visibility as the figure
of merit. from this analysis, we
readily see the key generation
rate and the quantum bit error
rate as figures of m ...[SEP]
```

Analysis: Strong focus on "instrumentation" and "optical," with repeated attention to "quantum," which looks like Physics related at the first glance. However, the model correctly grabbed the essence of the context, which is about quantum communication techology, classified under Computer Science (L1=0).

Incorrectly Classified Samples

- **Example 1:** True L1 = 6, Predicted L1 = 5

Top tokens attended by [CLS]:

Idx	Token	Attn
38	allergic	0.0479
16	allergic	0.0408
222	asthma	0.0346
60	##gen	0.0184
58	all	0.0165
5	##mun	0.0164
59	##er	0.0146
202	dust	0.0127
17	respiratory	0.0126
4	im	0.0126

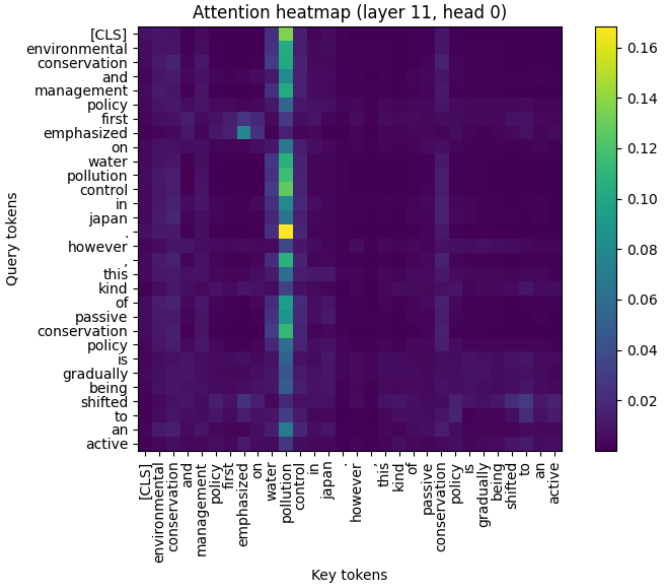
Decoded text (truncated):

```
[CLS] sublingual immunotherapy
( slit ) is a treatment for
allergic respiratory diseases
that has demonstrated efficacy
and safety. several
formulations of slit are now
available worldwide for
treatment of allergic rhinitis
( ar ). grass tablets
containing 15 to 25 µg of group
5 major allergen red ...[SEP]
```

Analysis: Emphasis on "allergic," "asthma," and "respiratory" which are common in both Medical Science and Biochemistry. The overlap likely caused confusion, leading to misclassification from L1=6 to L1=5.

Summary of Classified Samples 1. It's observed that symbols like periods (".") receive high attention scores in several correctly classified samples. This likely reflects their role in segmenting sentences, helping the model identify key phrases around punctuation. However, it takes away some attention to the content words which are more directly relevant to classification. 2. In misclassified samples, attention correctly focuses on domain-relevant terms, but overlapping vocabulary across classes leads to confusion. Which implicates that less attention should be given to those that cause ambiguity. 3. Overall, BERT's attention mechanism successfully highlights important tokens for most samples but still have possible improvements in dealing with ambiguous terms and non-informative tokens.

Heatmap Visualization



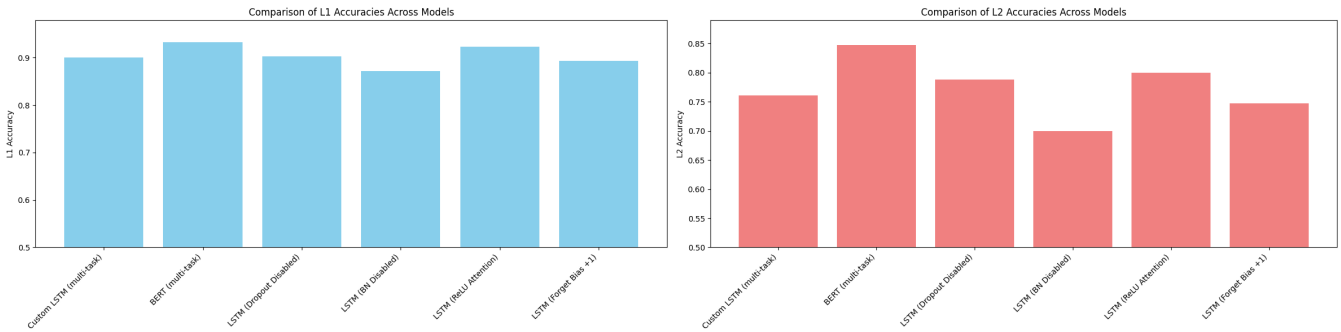
(a) Correctly classified sample.

Figure 2: Attention heatmap near output layer.

It is observed that in layer 11 (near the output), the attention heads become more specialized, amplifying tokens that are strongly relevant to class predictions. Most query tokens attend heavily to a small set of key tokens that are highly indicative of the document's class, while largely ignoring less relevant words. For `[CLS]`, the attention concentrates on these discriminative tokens, supporting the final classification.

3.3 Creativity & Discussion between the BERT and RNN results.

To enhance our project, we explored the impact of activation choice, initialization, regularization, and normalization techniques on LSTM performance.



(a) LSTM vs. BERT (L1 and L2).

(b) Alternative comparison view.

Figure 3: Performance comparison between LSTM and BERT across L1 and L2 tasks.

Model	Acc L1	Acc L2
LSTM	0.899777	0.761136
BERT	**0.932071**	**0.847439**
LSTM (Dropout Disabled)	0.903118	0.788419
LSTM (BN Disabled)	0.871938	0.699332
LSTM (ReLU Attention)	0.923163	0.799555
LSTM (Forget Bias +1)	0.893653	0.747216

Table 2: Performance across configurations for L1 and L2 classification. Best accuracies are bolded.

According to the results in the table above, we found that:

- Disabling dropout surprisingly improved LSTM accuracy, suggesting the model was not overfitting and benefited from retaining all learned features. This may be due to the current model not being very complex; more layers and training epochs may be needed to see the regularization effect.
- Disabling batch normalization before the output layer slightly decreased accuracy in L1 and more in L2, indicating BN helps stabilize training and improve generalization, especially for the more complex L2 task.
- Replacing the attention mechanism's `tanh` activation with ReLU boosted performance on both L1 and L2, more significantly on L1. This suggests that, for this dataset, ReLU's ability to model sparse, high-magnitude features is more effective than `tanh`'s bounded output.
- Initializing the forget gate bias to +1 did not influence performance much, indicating that small changes to gate initialization may not have a large impact on learning for this task.
- (Answering Questions in Task 3):

  Overall, BERT still outperformed all LSTM variants (performance comparison), highlighting the advantage of pretrained language models on text classification tasks (pretraining on an external corpus (like BERT) is good for the scientific text classification task). Subword tokenization in BERT likely contributed to its superior performance by better handling rare words and capturing deeper linguistic patterns, improving accuracy and data efficiency. BERT's transformer architecture models bidirectional context and long-range dependencies via self-attention, which helps with fine-grained L2 distinctions; our custom LSTM learns sequence dynamics but lacks pretrained knowledge and bidirectional global context, limiting its performance.

4 Discussion and Conclusions

Our results first show a clear performance difference between the two models: BERT consistently achieves higher accuracy than our LSTM on both L1 and L2 classification tasks. This confirms the advantage of pretrained language models, which benefit from large-scale pretraining and subword tokenization. The attention-matrix examination further supports this finding—BERT focuses strongly on class-indicative tokens when making correct predictions, while misclassifications often arise when attention is spread across overlapping domain terms. In our creativity experiments, modifying LSTM components such as dropout, normalization, attention activation, and gate initialization produced noticeable but limited improvements, with none surpassing BERT. BERT's transformer architecture captures bidirectional context and long-range relationships through self-attention, enabling it to handle fine-grained L2 distinctions more effectively. The LSTM captures sequence patterns but lacks global context and pretrained knowledge, limiting its accuracy. Overall, our study shows that while LSTMs can be enhanced through architectural tuning, pretrained transformers like BERT remain substantially more effective for scientific text classification. Future work may explore domain-specific pretraining, larger transformer variants, or interpretability-driven model designs.

5 Statement of Contributions

Jinghan and Yunjie are responsible for organizing the data, implementing the model, and running the experiments. Tailai is responsible for cleaning the data, gathering the results, and writing the report.

References

[1] Kamran Kowsari, Donald Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew Gerber, and Laura Barnes. Web of science dataset, 2018.

[2] Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, page 364–371. IEEE, December 2017.