Medical Abstracts Text Classification with LSTM

Manas Manoj Bedekar
manas.bedekar@mail.mcgill.ca

Shwetali Shimangaud
shwetali.shimangaud@mail.mcgill.ca

Chaitanya Tekane
chaitanya.tekane@mail.mcgill.ca

Abstract

This project investigates the efficacy of pre-trained BERT and a novel LSTM-CNN with Attention model in tackling text classification using the Ohsumed dataset. The primary objective is to evaluate the performance of these models across various metrics including accuracy, CPU and memory utilization, and their ability to handle long-range dependencies. BERT excels in capturing long-range dependencies but falls short in local attention, while the proposed LSTM-CNN with Attention model leverages LSTM and Attention for long-range dependencies and CNN for local attention. Thus, the hypothesis posits that the LSTM-CNN with Attention model can achieve comparable accuracy to BERT while operating with reduced computational resources. This shift enables exploration of alternative, resource-efficient approaches to identify the most optimal solution. The outcomes of this comparative analysis promise valuable insights for natural language processing (NLP) and medical text classification, enriching these domains with practical findings.

1 Introduction

Clinical NLP and AI-based classification enhance healthcare by leveraging unstructured medical reports. Automated text classification systems address categories like diseases, symptoms, and drugs using available datasets. A major challenge is the increasing volume of medical texts due to the surge in online medical information, leading to massive electronic data generation.

The National Library of Medicine's (NLM) MEDLINE database, focusing on biomedicine, contains over 29 million references to journal articles in the life sciences, making it a premier bibliographic resource. Ohsumed, which includes medical abstracts from the MeSH categories of the year 1991, was used in a study [Joachims, 1997].

The primary aim of this project is to demonstrate that the LSTM model requires fewer resources than state-of-the-art (SOTA) models. By integrating it with GloVe word embedding and an attention mechanism, the LSTM models can yield comparable results for long-range text classification problems. The inclusion of an attention layer in the neural network allows the model to focus not only on the current hidden state but also to consider the previous hidden state based on the decoder's previous output. The GloVe word embedding matrix refers to the set of word vectors learned by the GloVe (Global Vectors for Word Representation) model during its training process. In GloVe, each word in the vocabulary is allocated a vector in a continuous vector space. Evaluation criteria encompass accuracy, CPU usage, and memory utilization. This study conducts a thorough comparison of the performance of various LSTM models against a standard pre-trained BERT model.

2 Related Work

The authors of (Yan et al., 2018), proposed a neural network for multi-label document classification based on LSTM.

Medical text classification has emerged as a popular research area in the NLP community. Various classification algorithms are presented in recent years. In this part, we offer a recent study that made use of the ohsumed datasets. Where, the authors in (Camacho-Collados and Pilehvar, 2017) to tackle the multi-label classification of the Ohsumed-20000 data sets, proposed a convolutional neural network (CNN) model, where they achieved 36.0 precision. Regarding the second model, which is a hybrid model of CNN and long-short-term memory (LSTM), the accuracy achieved by this model is 37.5.

In this study (Lin et al., 2021), the authors used 7,400 documents for the Ohsumed dataset. They proposed the BertGCN model, where the Bidirectional Encoder Representations from Transformers (BERT) model allows the vectorization of docu-

ments, whereas the convolutional graph networks (GCN) method for document classification.

In (Huang et al., 2019), the authors used graphical neural network (GNN) techniques for text classification. They proposed a new GNN-based method for text classification. Instead of building a single corpus level graph, they produce a text-level graph for each input text. For a text-level graph, they connected word nodes in a reasonably small window in the text rather than directly connecting all word nodes. Initialized by Glove word embedding, a better result is obtained with a precision of 69.4 ± 0.6. For the TF-IDF representation, the logistic regression model performed well on long text datasets like (20NG)3 and outperformed CNN with randomly initialized word embedding.

3 Dataset

Our study relies on the Ohsumed dataset, comprising 20,000 medical abstracts dividing them into 10,000 for training and 10,000 for testing, with the specific task of categorizing the 23 cardiovascular disease categories. As illustrated in Figure 1, the data distribution highlights significant imbalances, notably observed in the dominance of the 'Pathological Conditions, Signs and Symptoms' category, which contains the highest volume of samples. To enhance text representation, we performed essential refining steps such as eliminating stop words, punctuation, white spaces, and lemmatization.

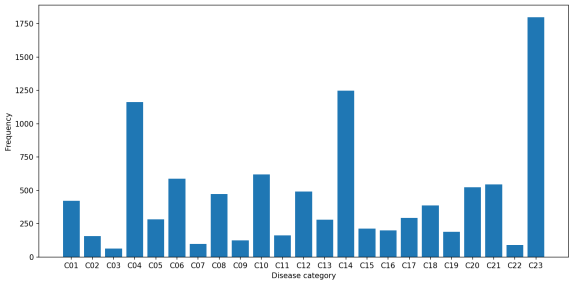

Figure 1: Data Distribution

4 Experiment setup

4.1 LSTM Implementation

We initiated our experiments with a basic LSTM configuration, comprising a single layer with 196 output states and recurrent dropout set to 0.2. The embedding dimension was established at 128.Subsequently, we enhanced the model by integrating pretrained GloVe word embeddings, replacing the

Bacterial Infections and Mycoses	C01
Virus Diseases	C02
Parasitic Diseases	C03
Neoplasms	C04
Musculoskeletal Diseases	C05
Digestive System Diseases	C06
Stomatognathic Diseases	C07
Respiratory Tract Diseases	C08
Otorhinolaryngologic Diseases	C09
Nervous System Diseases	C10
Eye Diseases	C11
Urologic and Male Genital Diseases	C12
Female Genital Diseases and Pregnancy Complications	C13
Cardiovascular Diseases	C14
Hemic and Lymphatic Diseases	C15
Neonatal Diseases and Abnormalities	C16
Skin and Connective Tissue Diseases	C17
Nutritional and Metabolic Diseases	C18
Endocrine Diseases	C19
Immunologic Diseases	C20
Disorders of Environmental Origin	C21
Animal Diseases	C22
Pathological Conditions, Signs and Symptoms	C23

Figure 2: Diseases categories

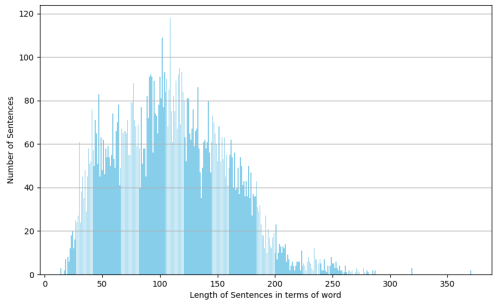

Figure 3: Distribution of number of words

initial embedding layer. In this iteration, the LSTM layer featured 296 output states, recurrent dropout remained at 0.2, and the embedding dimension defaulted to 300.Consistently, both experiments employed a batch size of 32, and convergence of the loss function was achieved after 4 epochs.

4.2 LSTM with Attention Mechanism

In the subsequent phase, our focus was on enhancing LSTM models by incorporating Attention mechanisms. Four variations were explored: augmenting the basic LSTM models with Attention resulted in the LSTM with Attention and LSTM with Attention combined with Glove word embedding models. Following this, the LSTM layer was replaced with BiLSTM in both scenarios, yielding the BiLSTM with Attention and BiLSTM with Attention using word embeddings. Consistency was maintained across all models by utilizing the same hyperparameters from earlier experiments. The Attention mechanism was implemented with a 'sigmoid' activation function.

LSTM	LSTM-WE	LSTM Attention	BiLSTM Attention	LSTM Attention WE	BiLSTM Attention WE	LSTM-CNN	BiLSTM-CNN Attention Glove	BiLSTM-CNN Attention Word2Vec
Embedding layer	**Glove Embedding layer**	Embedding layer	Embedding layer	**Glove Embedding layer**	Glove Embedding layer	Embedding layer	**Embedding layer Glove**	**Embedding layer Word2Vec**
Dropout(0.5)	Dropout(0.5)	Dropout (0.5)	Dropout (0.5)	Dropout(0.5)	Dropout(0.5)	Dropout(0.5)	Dropout(0.5)	Dropout(0.5)
LSTM	LSTM	LSTM	**BiLSTM**	LSTM	**BiLSTM**	**Conv1D 8*8**	Conv1D 8*8	Conv1D 8*8
softmax	softmax	**Attention layer**	Attention layer	Attention layer	Attention layer	max pooling	max pooling	max pooling
		max pooling	max pooling	max pooling	max pooling	LSTM	**BiLSTM**	BiLSTM
		softmax	softmax	softmax	softmax	softmax	**Attention**	Attention
							average pooling	average pooling
							softmax	softmax

Figure 4: Architecture

CNN-LSTM Attention	LSTM-CNN Attention	CNN & LSTM in parallel Attention	
Embedding layer	Embedding layer	Embedding layer	Embedding layer
Dropout(0.5)	Dropout(0.5)	Dropout(0.5)	Dropout(0.5)
Conv1D 3*3	BiLSTM	Conv1D 3*3	BiLSTM
max pooling	BiLSTM	max pooling	
Dropout(0.5)	Conv1D 3*3	Dropout(0.5)	
Conv1D 3*3	max pooling	Conv1D 3*3	BiLSTM
max pooling	Dropout(0.5)	max pooling	
BiLSTM	Conv1D 3*3	Conv1DTranspose 5*5	
BiLSTM	max pooling	Conv1D 1*1	
Attention	Attention	**Concatenation layer**	
average pooling	average pooling	Attention	
softmax	softmax	average pooling	
		softmax	

Figure 5: CNN LSTM architecture

Model	Train	Valid	Test
CNN-LSTM	45%	41%	43%
LSTM-CNN	32%	33%	31%
CNN LSTM in Parallel	38%	37%	37%

Table 1: Accuracy for different CNN LSTM architectures

4.3 LSTM with CNN and Attention

In this study, we explored the integration of CNN and LSTM through three distinct architectural configurations, as illustrated in Figure 5. Our comparative analysis of these architectures revealed superior performance in the first configuration, where LSTM layers were sequentially stacked following the CNN layers. Subsequently, the output of the LSTM layer was utilized as input for an Attention mechanism.Table 1 presents the accuracy metrics obtained for each model, demonstrating the superiority of the initial configuration, which outperformed the alternative two models. As a result, the selected architecture formed the basis for subsequent experimentation and analysis.

In the final phase, we combined CNN with LSTM to harness local pattern recognition from CNN and the long-range dependency learning of LSTM. Three models were developed: The first model, LSTM-CNN, incorporated a conv1D layer with 128 filters (8x8) preceding the LSTM layer.The second model replaced the embedding layer with Glove embeddings, swapped LSTM with BiLSTM, and added an Attention mechanism post-BiLSTM, resulting in BiLSTM-CNN Attention with Glove word embeddings.For the third model, the Glove word embeddings were replaced with Word2Vec embeddings. To train the Word2Vec model, word vector size was 2000 with a window size of 4 was used. Consistency in hyperparameters was maintained across all these models to ensure fair comparisons.

In all of the above experiments Adam optimizer was used with categorical cross entropy loss. Learning rate was set to default. The architecture for each model is described in Figure 4.

4.4 BERT Implementation

In this phase, we introduced BERT into our research using the 'bert-base-uncased' pretrained model. Initially, we fine-tuned this model on our Obsumed dataset. However, due to BERT's character limit of 512 characters, we explored LongFormer as an alternative. LongFormer's training time exceeded 12 hours on GPU, requiring a batch size of 1, which was beyond our resource capabilities. Consequently, we employed a workaround by chunking the original sentences into segments, each restricted to a maximum length of 512 characters. Labels were adjusted accordingly, and classification was performed on this chunked dataset using the original BERT model. The BERT model was then fine-tuned using the AdamW optimizer and cross-entropy loss.

5 Results

5.1 CPU and Memory Utilization

One crucial evaluation parameter in this study involved examining the resource utilization of each

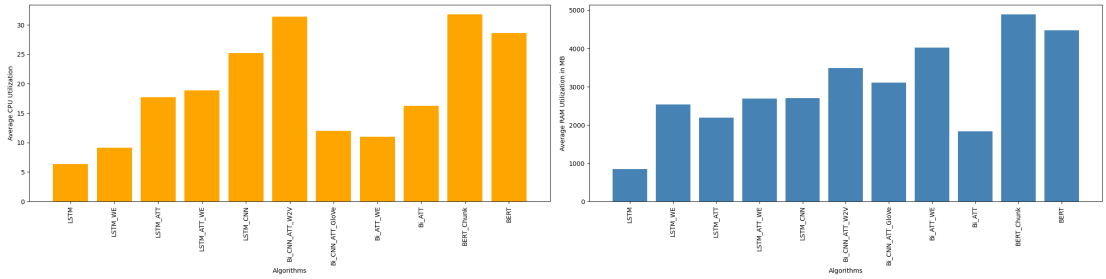

Figure 6: CPU and RAM Utilization

Models	Train Accuracy	Test Accuracy	CPU Use	Memory Use MB
Simple LSTM	40.52%	32.25%	6.32	845
LSTM-WE	40.74%	39.58%	9.11	2536
LSTM Attn	50.83%	39.83%	17.72	2186
LSTM Attn-WE	48.51%	45.13%	18.86	2696
CNN-LSTM	38.25%	38.24%	25.41	2702
BiLSTM Attn	32%	27.94%	16.27	1830
BiLSTM Attn WE	41.92%	42.33%	10.96	4020
CNN-BiLSTM Attn-WE	39.79%	33.73%	12.02	3110
CNN-BiLSTM Attn-WE(Word2Vec)	40.2%	37.13%	31.36	3483
BERT	52%	7%	28.8	4480
BERT with Chunking	60%	9%	31.79	4895

Table 2: Model Results

Models	Train Acc	Test Acc
Simple LSTM	34.3%	31.5%
LSTM-WE	38.2%	36.7%
LSTM Attn	35.1%	31.9%
LSTM Attn-WE	37.4%	34.6%
CNN-LSTM	11.3%	10.1%
BiLSTM Attn	32.5%	31.1%
BiLSTM Attn WE	35.6%	35.3%
CNN-BiL. Attn-WE	44.9%	40%
CNN-BiL. Attn-W2V	35.8%	35.1%
BERT	25%	23%

Table 3: Model Results on Long text

algorithm. The findings from Figure 6 and Table 2 suggest that LSTM models without pre-trained word embeddings utilized the fewest resources, yet yielded comparable results. Conversely, pre-trained SOTA models consumed substantial resources but did not perform well during the testing phase. On average SOTA models consumed more tha 4500 MB memory and blocked around 30% CPU. Similar to SOTA models, LSTM with Attention and a word embedding layer also utilized significant resources; however, in contrast, they improved the outcomes of the existing model.

5.2 Performance on long text

The primary objective of this study was to assess the performance of various LSTM variants against the SOTA BERT model on lengthy texts. LSTM models, known to encounter the vanishing gradient problem and possess limited information retention capabilities, can potentially handle extended texts when coupled with an Attention mechanism. To validate this hypothesis, we conducted an evaluation using approximately 1000 of the longest sentences extracted from both our training and testing datasets. The aim was to compare the accuracy of each model on this filtered dataset.The results revealed that our 'CNN BiLSTM with word embedding and Attention' model achieved an accuracy of 44.9%. In contrast, the pretrained BERT model attained a 25% accuracy score. This outcome strongly suggests that the LSTM model integrated with Attention mechanism and CNN outperforms the BERT model when dealing with longer texts. This underscores the effectiveness of leveraging LSTM's augmented with Attention mechanisms in processing extensive textual information, thereby showcasing superior performance over the BERT model in this specific context. The detailed results are shown in Table 3.

5.3 Impact of Word Embedding

From the results Table 2, it's observed that word embedding matrix layer enhanced the accuracy of LSTM, LSTM-Attention, and BiLSTM models which implies that pre-trained word embedding representation improves the overall accuracy with handling out of vocabulary word with some extent and converges faster than one-hot encoding.

5.4 Impact of Attention Layer

The results Table 2 clearly indicate that attention mechanism allowed the model to focus more on relevant parts of the input sequence, addressing issues of information loss and improved the model's ability to capture long-range dependencies.

5.5 Impact of Bidirectional LSTM

The utilization of a Bidirectional LSTM model did not yield an enhancement in performance, as confirmed across different use cases. This phenomenon may be attributed to the dataset's characteristics, where the word dependencies are not extensively long, and the LSTM model suffices to retain pertinent information. Additionally, the intricate nature of both Attention mechanisms and Bidirectional LSTMs, designed for handling long-range dependencies, implies that they may capture analogous features or patterns in textual data. Consequently, substituting BiLSTM with LSTM might not yield a significant performance improvement. For the details refer Table 2.

6 Discussion

The exploration into text classification using the Ohsumed dataset showcased compelling outcomes. Integrating BiLSTM with Attention mechanisms effectively tackled the vanishing gradient issue, notably boosting model performance. The addition of CNN acted as a local attention layer, capturing intricate patterns in the data. Furthermore, Self-Attention aided in comprehending global text semantics, resulting in a holistic understanding. Employing pretrained Word Embeddings optimized data representation for Neural Network models, markedly enhancing performance. Remarkably, the CNN-LSTM model, incorporating attention and word embeddings, rivaled pretrained BERT models in text classification tasks, offering a resource-efficient alternative.This study prioritized evaluating computational resource utilization. The CNN-LSTM model with attention and glove word embeddings notably consumed fewer CPU and RAM resources than the intensive BERT model. The practical advantage of the LSTM variant over fine-tuning BERT models becomes evident in resource-constrained environments.In analyzing longer texts, the LSTM variant outperformed the BERT model, achieving 45.13% accuracy. Tailoring these models to specific tasks remains crucial due to variations in performance across datasets. The study emphasizes the importance of combining LSTM, CNN, and attention mechanisms to handle long-range dependencies while capturing word semantics contextually. Further, we can explore models with higher computational complexity like SPADE (Zuo et al., 2022), Structured State Space Model (Gu et al., 2022) for better accuracy results. The model's applicability is constrained to scenarios lacking access to advanced GPU resources for training or fine-tuning, and when specialized pretrained models tailored for the specific field are unavailable.

7 Conclusion

This study substantiates the efficacy of the LSTM-CNN Attention variant model in achieving comparable accuracy to state-of-the-art models while consuming fewer computational resources, specifically tailored for the Ohsumed dataset. The amalgamation of LSTM and CNN with attention mechanisms presents a promising avenue for text classification tasks, emphasizing the importance of model architecture selection and fine-tuning for optimal performance across diverse datasets and tasks.

8 Statement of Contribution

This study's contribution lies in Manas Bedekar's meticulous experimentation across diverse LSTM variants and his development of code to measure CPU and RAM utilization. Simultaneously, Shwetali Shimangaud extensively explored BiLSTM models, delved into the LSTM-CNN architecture, and evaluated these models' handling of long-range text. The collective efforts of Manas and Shwetali culminate in the comprehensive findings detailed within this report.

References

Jose Camacho-Collados and Mohammad Taher Pilehvar. 2017. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *arXiv preprint arXiv:1707.01780*.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces.

Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. *arXiv preprint arXiv:1910.02356*.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727*.

Yan Yan, Ying Wang, Wen-Chao Gao, Bo-Wen Zhang, Chun Yang, and Xu-Cheng Yin. 2018. Lstmˆ 2 2: Multi-label ranking for document classification. *Neural Processing Letters*, 47:117–138.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Denis Charles, Eren Manavoglu, Tuo Zhao, and Jianfeng Gao. 2022. Efficient long sequence modeling via state space augmented transformer. *arXiv preprint arXiv:2212.08136*.