Team #1- AI D. Detectives
Machine Generated Text Detection

Yi Tian; 40280140; yi.tian@mail.concordia.ca
and
Naveen Rayapudi; 40291526; rayapudinaveen777@gmail.com

December 19, 2024

Abstract. In this era where artificial intelligence is changing every domain from communication to content creation, distinguishing between human-generated and machine-generated text has become crucial. The ability to identify such content is essential for preserving the integrity of digital platforms. This report evaluates Long Short-Term Memory (LSTMs) and RoBERTa for the critical task of detecting machine-generated English text. The study reveals that RoBERTa consistently outperforms LSTM models, achieving higher accuracy and macro-F1 scores, particularly as the training data increases. RoBERTa's ability to use **"Attention"** mechanism and **Long-Term Dependencies** to capture complex language patterns , addressing the information bottleneck,and feeding the words in a parallel fashion gives it the upper hand over LSTM. In contrast, LSTM models struggle with processing long-term dependencies in text. The findings over the training data in sequence classification tasks highlight that RoBERTa outperforms LSTM models and stands as a more effective model for addressing the challenges of AI-generated text detection.

Links to Implementations and Dataset:

Resource	**Link**
LSTM	LSTM Implementation
RoBERTa	RoBERTa Implementation
Dataset	Dataset Information

Contents

1 Goal of the project

This project's main goal is to create a detection system that can distinguish between English texts that were written by humans and those that were gen- erated by machines. And to compare the performance of two machine learning models, RoBERTa and LSTM, in detecting machine-generated English text. Additionally, we aim to analyze how the LSTMs models handle different data sizes and change in parameters. With the growing capability to produce machine-generated text, this detection system shows a trustworthy way to identify such content, which is important for applications such as academic integrity, content verification, and disinformation prevention.

2 Methodology

2.1 Dataset

- The experiments were conducted using the official English dataset provided by COLING-2025-Workshop-on-MGT-Detection-Task1.
- The dataset has two labels: Human (0) and Machine-generated (1). The data format is as follows (using the first training sample as an example):

Listing 1: Example of Dataset Format

```
{
  "id": "f05034ca-d1da-445d-a6a2-5869ade0dfc3",
  "source": "m4gt",
  "sub_source": "reddit",
  "lang": "en",
  "model": "llama3-8b",
  "label": 1,
  "text": "Hitler's plans for the succession and
      power structure after his death are shrouded in
       mystery, as he never explicitly wrote down his
       intentions. However, ..."
}
```

- The distribution of the dataset is shown in Figure 1 and Figure 2.
- It can be observed that the training set contains a total of 610,767 samples, including 228,922 Human-generated texts and 381,845 Machine-generated texts. The training set is imbalanced, and the class proportions differ from those in the development set.

- During training, a subset of the provided training set was used, and the entire training set was also evaluated later. However, no attempts were made to balance the dataset. This approach is not the best practice, as discussed in the limitations section.

2.2 Data Preprocessing

- Preprocess data (including making everything lowercase and removing special characters) to reduce the vocabulary.
- `<UNK>` were used for out of vocabulary words, and `<PAD>` tokens were introduced to pad sequences to a uniform length (512), ensuring compatibility with batch training.
- Text data was tokenized at the word level and transformed into sequences by mapping tokens to indices, followed by truncation or padding.

2.3 Model Architecture

- The LSTM classifier experiment with both unidirectional and bidirectional architectures:
  - Unidirectional LSTM: The input sequence is fed into the RNN in a left-to-right order. The hidden state of the last time step from the final LSTM layer was used as input for the classification head.
  - Bidirectional LSTM: Model structure is shown in Figure 3. Processes the input sequence in both forward and backward directions. The hidden states from the last forward and backward steps were concatenated, resulting in a representation twice the size of the LSTM hidden vector size.
- Both architectures passed the extracted hidden state(s) through a fully connected neural network (FFNN) with a sigmoid activation function for binary classification.
- The RoBERTa classifier was included as a baseline model:
  - The roberta-base pretrained model was loaded and fine-tuned globally on the training set.

- RoBERTa was chosen as the baseline instead of BERT because we believe it is more likely to perform better on this task, primarily due to the differences between RoBERTa and BERT, as outlined in Table 1.
- Early stopping (using a subset of the development set) was enabled to prevent overfitting during training.

2.4 Key Implementation Details

- The models were implemented using PyTorch's `nn.LSTM` module.
- Bidirectionality was enabled by setting `bidirectional=True`.
- Experiments included hyperparameter tuning to analyze the effect of embedding size, dropout rate, and hidden vector size on model performance.

2.5 Training and Hyperparameter Tuning

- The LSTM models were trained on the entire training set and evaluated on the development test set. For hyperparameter tuning, a subset of 100,000 samples from the training set was used due to computational limitations.
- Hyperparameter Settings:
  - Input sequence length: 512.
  - Embedding size: 300 (for base experiments) with variations (200, 400, 600) tested for hyperparameter tuning.
  - Hidden vector size: 128 (for base experiments) with variations (256, 512) tested for hyperparameter tuning.
  - Number of LSTM layers: 2.
  - Dropout rate: 0.5 (varied between 0.01 and 0.5 during tuning).
  - Number of classification layers: 2.
- **Training Settings:**
  - Cost function: Binary Cross Entropy Loss.
  - Model Parameter Intialization: Shown in Table 2 (which may not be the optimal way, as discussed in the limitations section).

- Optimizer: Adam optimizer with a learning rate of 0.001.
- Batch size: 64.
- Training epochs: 10.

3 Evaluation

3.1 Results

- The performance of the models was evaluated on the test set using Macro-F1 Score and Accuracy as metrics.
- Table 3 shows the performance of the baseline RoBERTa classifier, trained on a 1/10 subset of the training set (61,076 samples) and evaluated on the entire development test set.
- Figure 5 shows the performance of the LSTM classifier with different training set sizes.
- Table 4 shows the effect of embedding size on the performance of the LSTM classifier.
- Table 5 shows the effect of hidden vector size on the performance of the LSTM classifier.
- Figure 6 shows the training loss after 10 epochs for different dropout rates, while Figure 7 shows the effect of dropout rate on the performance of the LSTM classifier.

3.2 Analysis

- **Effect of Dataset Sizes:**
  - The performance of the LSTM classifier improves as the dataset size increases from 1,000 to 100,000.
  - The Bidirectional LSTM achieves slightly better scores on larger datasets due to its ability to handle complex dependencies and capture contextual information from both directions.
- **Effect of Embedding Sizes:**
  - Varying the embedding size from 200 to 600 shows inconsistent performance variations across the Uni-Directional and Bi-Directional architectures.

  - Uni-Directional LSTM achieves its best performance with an embedding size of 200, attaining a Macro-F1 score of 0.64005, and accuracy of 0.65851. This suggests that reducing the embedding size can enhance performance for Uni-Directional LSTMs, possibly due to better generalization with fewer parameters when the training data is insufficient.
  - Bi-Directional LSTM achieves its best performance with an embedding size of 300, obtaining a Macro-F1 score of 0.64571 and accuracy of 0.66020. This indicates that Bi-Directional LSTMs benefit more from moderate embedding dimensions, as larger embedding sizes like 600 do not result in better performance and require increased computational resources.
  - Interestingly, both architectures perform worse with an embedding size of 600, reinforcing the notion that excessively large embeddings may lead to overfitting or inefficiencies.

- **Effect of Hidden Vector Sizes:**
  - Increasing the hidden vector size improves the performance of both models in terms of **Macro-F1, Accuracy, and Loss** (see Table [3]).
  - **Bi-Directional LSTM:**
    * Optimal hidden vector size is **256**, achieving the highest **Macro-F1 (0.6391)** and **Accuracy (0.6559)** scores.
  - **Uni-Directional LSTM:**
    * Shows a slight increase in performance as the hidden vector size grows.

- **Effect of Dropout Rate on Training Loss:**
  - Results from figure [5] highlight the following:
    * The Bi-Directional LSTM consistently outperforms the Uni-Directional LSTM across all dropout rates, especially at higher regularization levels **0.5 and 0.01**.
    * The Uni-Directional LSTM performs better at the lowest dropout rate **(0.001)**, suggesting it benefits from less regularization.
    * The **0.5** dropout rate is an effective regularization level for the Bi-Directional LSTM.

  * The **0.01** dropout rate strikes a good balance for both models.

4 Role of each team member (if applicable)

Yi-Tian: Took responsibility for selecting the models and designing the architecture. Also responsible for the implementation of the baseline classifier RoBERTa. Completed the sections on Methodology, Limitations, and Differences with the Original Proposal in the report.

Naveen Rayapudi: Contributed to the implementation of Uni-Directional and Bi-Directional LSTM classifiers, as well as hyperparameter tuning. Completed the remaining sections of the report.

Both of us contributed to the completion of the project poster.

5 Limitations

5.1 RoBERTa Classifier

- **Freezing Parameters**: The current approach globally fine-tunes parameters, which is more computationally expensive compared to adding a classification head, and may not result in significant performance improvement.

5.2 LSTM-based Classifier

- **Imbalanced Dataset**: In the training set, the ratio of human-generated text to machine-generated text is approximately 1:1.67. This imbalance causes the trained model to favor predicting text as machine-generated. A more balanced sampling approach could help mitigate this systematic issue.

- **Sequence Truncation**: Similar to RoBERTa, truncating sequences to 512 tokens impacted the LSTM's ability to capture long-range dependencies, further reducing its performance compared to models with contextual embeddings.

- **Output Representation**: Only the last hidden state of the LSTM is used for classification. This approach may overlook important information from earlier timesteps, particularly in longer sequences. Alternative strategies, such as mean or max pooling across all hidden states, could provide richer sequence-level representations.

- **Out-of-Vocabulary (OOV) Issues**: The LSTM implementation relies on word-level tokenization, which results in significant OOV issues. Words not present in the vocabulary are mapped to a single `<UNK>` token, leading to a loss of semantic information. Switching to subword tokenization techniques like Byte Pair Encoding (BPE) could mitigate this limitation.
- **Model Parameter Initialization**: The current implementation uses PyTorch's default parameter initialization for LSTM, including setting the forget gate bias to 0. However, in practice, setting it to 1 is often more effective in alleviating gradient vanishing issues and accelerating model training.
- **Batch Training Challenges**: The current training setup for the LSTM lacks advanced techniques such as masking, which could handle variable-length sequences more effectively and improve gradient flow during training.

6 Difference with your original proposal

Initially, we planned to create an ensemble of RoBERTa, vanilla RNN, and LSTM models. However, upon further research, we learned that vanilla RNN cells are largely deprecated due to their susceptibility to gradient vanishing and explosion issues. LSTM cells, which are specifically designed to address these limitations, are now commonly used as a replacement. Given that LSTM models are more robust and easier to train compared to vanilla RNNs, we determined that ensembling these two approaches would not provide meaningful advantages. Consequently, we chose to focus on optimizing the performance of the LSTM model and comparing it against the baseline RoBERTa model.

7 Conclusions

This project focused on the critical task of distinguishing between human-generated and machine-generated English text using machine learning models, specifically LSTM classifiers. The study revealed several key findings:

- The RoBERTa classifier consistently outperformed the LSTM models, achieving higher Macro-F1 scores. This highlights the effectiveness of pre-trained transformer-based models in handling complex text classification tasks.
- Uni-Directional and Bi-Directional LSTMs exhibited varying performance based on hyperparameters such as embedding size, hidden vector size, and dropout rate. Notably, Bi-Directional LSTM generally achieved better results than Uni-Directional LSTM, due to its ability to capture bidirectional contextual information.
- The embedding size and hidden vector size played significant roles in determining model performance. While a smaller embedding size (200) worked best for Uni-Directional LSTMs, Bi-Directional LSTMs benefited from a moderate embedding size (300).
- Dropout rates impacted the regularization and overall performance of the models. A dropout rate of 0.5 provided optimal results for Bi-Directional LSTMs, while Uni-Directional LSTMs preferred lower regularization.

Despite these findings, the project faced limitations such as dataset imbalance, sequence truncation, and out-of-vocabulary issues, which likely constrained the performance of LSTM models. Future work can address these limitations by exploring advanced tokenization techniques, balancing sampling strategies, and incorporating alternative architectures or ensemble methods.

In conclusion, while both models demonstrated their strengths, RoBERTa proved to be a more robust and scalable solution for detecting machine-generated text. This work underscores the importance of fine-tuning model parameters and leveraging pre-trained language models to address challenges in AI-generated text detection effectively.

8 References

1 Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* arXiv preprint arXiv:1907.11692.

2 Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). *Transformers: State-of-the-Art Natural Language Processing.* In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45).

3 Improving Sentiment Classification Using a RoBERTa-Based Hybrid Model. (2024). Retrieved from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10733963/`

4 Joshy, A., & Sundar, S. (2022). *Analyzing the Performance of Sentiment Analysis Using BERT, DistilBERT, and RoBERTa.* In Proceedings of the 2022 IEEE International Power and Renewable Energy Conference (IPRECON). Piscataway, NJ. `https://ieeexplore.ieee.org/document/10059542`

5 Petropoulos, P., & Petropoulos, V. (2024). *RoBERTa and Bi-LSTM for Human vs AI Generated Text Detection.* Working Notes of CLEF.

6 Bahad, S., Bhaskar, Y., & Krishnamurthy, P. (2024, June). *Fine-Tuning Language Models for AI vs Human Generated Text Detection.* In Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024) (pp. 918-921).

A Appendix

		Training Set			Development Set		
Source	Sub-source	Human	Machine	Total	Human	Machine	Total
hc3	finance	2579	3189	5768	1113	1301	2414
	medicine	886	883	1769	352	380	732
	open$_{q}a$	823	2339	3162	364	1015	1379
	reddit$_{e}li$5	34329	11680	46009	14781	4959	19740
	wiki$_{c}sai$	523	580	1103	245	262	507
m4gt	arxiv	22484	30684	53168	9487	13003	22490
	outfox	2162	40973	43135	995	17390	18385
	peerread	3300	16169	19469	1398	6749	8147
	reddit	20353	32609	52962	8663	14076	22739
	wikihow	19454	35305	54759	8532	15168	23700
	wikipedia	19029	25341	44370	8145	10881	19026
mage	cmv	6020	16592	22612	2618	7026	9644
	cnn	265	0	265	131	0	131
	dialogsum	210	0	210	98	0	98
	eli5	15347	21849	37196	6451	9340	15791
	hswag	6806	19169	25975	2903	8085	10988
	imdb	269	0	269	107	0	107
	pubmed	273	0	273	105	0	105
	roct	6916	20008	26924	2930	8439	11369
	sci$_{g}en$	6613	14390	21003	2891	6145	9036
	squad	14519	14875	29394	6333	6330	12663
	tldr	5558	15808	21366	2329	6930	9259
	wp	7919	21215	29134	3393	9390	12783
	xsum	6992	22129	29121	2925	9621	12546
	yelp	25293	16058	41351	11039	6940	17979
Grand Total		228922	381845	610767	98328	163430	261758

Figure 1: Dataset distribution by sub-source

Performance of LTMS over 610,767 samples dataset:

Model	Loss	Accuracy	Macro-F1
UniLSTM	0.42096	0.70182	0.68182
BiLSTM	0.30765	0.70114	0.67718

Feature	BERT	RoBERTa
Training Data Size	16GB	160GB
Masking Strategy	Static Masking	Dynamic Masking
Next Sentence Prediction	Includes NSP Task	Removes NSP Task
Training Time and Batch Size	Shorter time, smaller batch size	Longer time, larger batch size
Performance	Excellent	Superior

Table 1: Comparison of BERT and RoBERTa

Layer	Parameter Initialization
Embedding (`nn.Embedding`)	Initialized with a uniform distribution: $$\left[-\sqrt{\frac{1}{\text{embedding\_dim}}}, \sqrt{\frac{1}{\text{embedding\_dim}}}\right].$$
LSTM (`nn.LSTM`)	**Weight matrices:** Uniform distribution: $$\left[-\sqrt{\frac{1}{\text{hidden\_size}}}, \sqrt{\frac{1}{\text{hidden\_size}}}\right].$$ **Bias terms:** Initialized to zero (including Forget Gate Bias).
Fully-Connected (`nn.Linear`)	**Weight matrix (W):** Kaiming uniform: $$\left[-\sqrt{\frac{1}{\text{fan\_in}}}, \sqrt{\frac{1}{\text{fan\_in}}}\right].$$ **Bias (b):** Initialized to zero.

Table 2: Parameter initialization mechanisms for each layer in the model.

Model	Macro-F1	Accuracy
RoBERTa Classifier	0.69122	0.71250
Uni-directional LSTM	0.46513	0.52250
Bi-directional LSTM	0.51663	0.51750

Table 3: Baseline Model RoBERTa Peformance (with training set size of 61,076)

		Training Set		Development Set	
Source	Model	Human	Machine	Human	Machine
hc3	gpt-35	0	18671	0	7917
	human	39140	0	16855	0
m4gt	bloomz	0	21061	0	8991
	cohere	0	20808	0	8896
	davinci	0	19345	0	8210
	dolly	0	8932	0	3931
	dolly-v2-12b	0	1938	0	831
	gemma-7b-it	0	12162	0	5240
	gemma2-9b-it	0	8366	0	3629
	gpt-3.5-turbo	0	25856	0	11005
	gpt4	0	9956	0	4300
	gpt4o	0	10374	0	4247
	human	86782	0	37220	0
	llama3-70b	0	12333	0	5181
	llama3-8b	0	12057	0	5290
	mixtral-8x7b	0	15865	0	6623
	text-davinci-003	0	2028	0	893
mage	13B	0	5385	0	2367
	30B	0	5769	0	2380
	65B	0	5815	0	2404
	7B	0	5083	0	2166
	GLM130B	0	4398	0	1842
	bloom$_{7}b$	0	5151	0	2201
	flan$_{t}5_{b}ase$	0	6566	0	2887
	flan$_{t}5_{l}arge$	0	6500	0	2893
	flan$_{t}5_{s}mall$	0	6570	0	2811
	flan$_{t}5_{x}l$	0	6429	0	2739
	flan$_{t}5_{x}xl$	0	6532	0	2777
	gpt-3.5-turbo	0	15991	0	6682
	gpt$_{j}$	0	3468	0	1480
	gpt$_{n}eox$	0	4734	0	2021
	human	103000	0	44253	0
	opt$_{1}.3b$	0	5553	0	2351
	opt$_{1}25m$	0	5735	0	2469
	opt$_{1}3b$	0	4988	0	2296
	opt$_{2}.7b$	0	5736	0	2586
	opt$_{3}0b$	0	5637	0	2376
	opt$_{3}50m$	0	5128	0	2252
	opt$_{6}.7b$	0	5642	0	2378
	opt$_{i}ml_{3}0b$	0	6008	0	2619
	opt$_{i}ml_{m}ax_{1}.3b$	0	6176	0	2660
	t0$_{1}1b$	0	6309	0	2620
	t0$_{3}b$	0	6602	0	2849
	text-davinci-002	0	14884	0	6359
	text-davinci-003	0	15304	0	6781
Grand Total		228922	381845	98328	163430

Figure 2: Dataset distribution by model

Bi-Directional LSTM

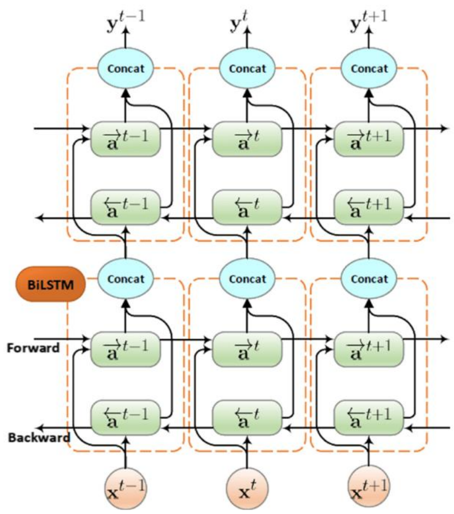

Figure 3: Double layer

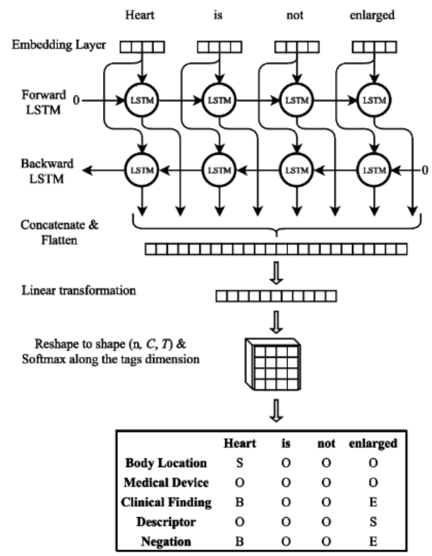

Figure 4: Single layer

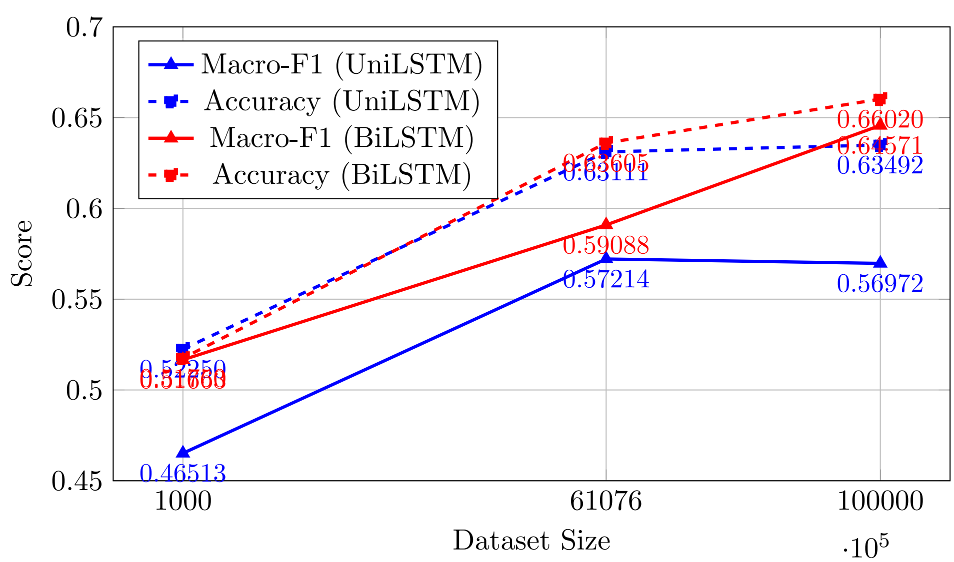

Figure 5: Performance of LSTM classifier on Different Dataset Sizes.

Embedding Size	LSTMS	Macro-F1	Accuracy
600	Uni-Directional	0.57821	0.64078
	Bi-Directional	0.62790	0.63931
300	Uni-Directional	0.56972	0.63492
	Bi-Directional	**0.64571**	0.66020
400	Uni-Directional	0.55752	0.63108
	Bi-Directional	0.63358	0.64257
200	Uni-Directional	0.64005	0.65851
	Bi-Directional	0.59183	0.62291

Table 4: Effect of Embedding Size on Model Performance (with training set size of 100,000)

Hidden Vector Size	LSTMS	Macro-F1	Accuracy	Loss
128	Uni-Directional	0.56972	0.63492	0.4903
	Bi-Directional	**0.64571**	0.66020	0.1354
256	Uni-Directional	0.62003	0.65399	0.3257
	Bi-Directional	0.63914	0.65590	0.0944
512	Uni-Directional	0.60246	0.64573	0.2453
	Bi-Directional	0.62955	0.64604	0.1134

Table 5: Effect of Hidden Vector Size on Model Performance (with training set size of 100,000 and embedding size of 300)

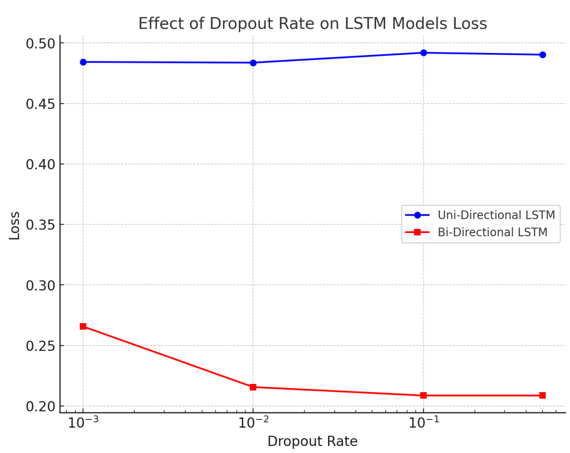

Figure 6: Training loss after 10 epochs for different dropout rates (with training set size of 100,000, embedding size of 300, and hidden vector size of 128)

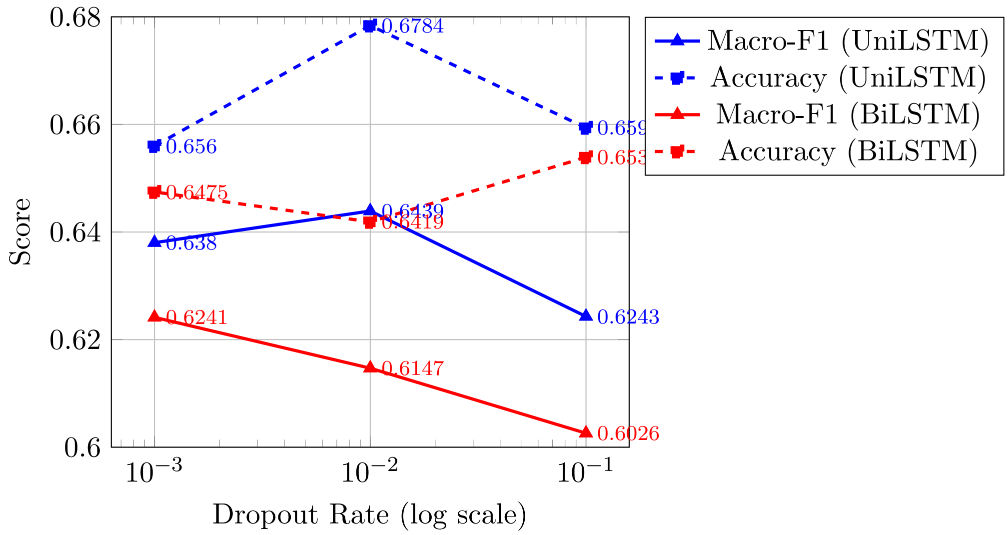

Figure 7: Effect of Dropout Rate on Macro-F1 and Accuracy