Comp551 - Assignment 4

Members:
Adam Dufour (261193949)
Gabriel Caballero (261108565)

December 11, 2025

Abstract

In this work, the recurrent neural network (RNN) were studied in order to probe their efficacy and usefulness in a Seq2Vec problem. The study case for this report is the assignment of a field to a database of different articles coming from [4]. On top of building the long short-term memory (LSTM), a fine tuned transformer model (BERT) was analyzed and this work reports that for these models, the performance obtained is of 86.46% and 92.77% respectively on the WOS-11967 abstract dataset for the parent labels. As for the child labels, accuracies of 75.14% and 85.96% for the LSTM and BERT models respectively were obtained.

Introduction

An RNN is a powerful tool in machine learning to help with problems handling sequences. This is a particularly relevant problem case since it has been shown to be of great use. For example, the model built to predict protein folding named AlphaFold [2], the team that developed this tool has shared the 2024 Nobel Prize in Chemistry. AlphaFold shows that RNN are a powerful tool and its uses are relevant in today's world. However, these models contain fatal flaws (such as vanishing gradients and lack of long-term retention), this is why this work will focus on LSTM and BERT models, models that use memory and attention to overcome the limitations of an RNN. This work focuses on a Seq2Vec problem where the models are looking to categorize abstracts into different scientific fields. The findings of this work show that the transformer based model BERT showed better results across the different datasets. WOS-5736, the dataset containing the smallest amount of categories, showed the best results.

Datasets

The datasets that were used to train the models are three collections of abstracts from varying fields in the scientific literature [3]. Set 1 (WOS-11967), 2 (WOS-46985) and 3 (WOS-5736) can be better understood when visualizing table 1. These sets are known under the "Web of Science Dataset" and the fields the datasets span across are: Computer Science, Electrical Engineering, Psychology, Mechanical Engineering, Civil Engineering, Medical Science and Biochemistry. Other work citing these datasets can be observed in [4]. However, this work will focus on WOS-11967, but understanding the whole dataset gives a better understanding and a clearer picture about the dataset.

DataSets	Label types	Number of labels	Number of Abstracts
WOS-11967	Parent labels	7	11967
	Child labels	35	
WOS-46985	Parent labels	7	46985
	Child labels	134	
WOS-5736	Parent labels	3	5736
	Child labels	11	

Table 1: Datasets Composition

Results

Task 1 - Retrieving and cleaning the Dataset

After loading the dataset (WOS-11967), the abstract's sequences were tokenized and embedded in order to afford a more "machine friendly" information to represent the texts. In table 2, the different representations of the abstract can be visualized. As seen in this table, the phrases are completely broken down and transformed into integers to represent the sequence and prepare for the Seq2Vec problem.

Original Phrase	The aim of this study was to investigate (a) the behavioral cues that are displayed by
Tokenized representation	['the', 'aim', 'of', 'this', 'study', 'was', 'to', 'investigate', 'a', 'the', 'behavioral', 'cues', 'that', 'are', 'displayed', 'by']
LSTM encoding	[2, 374, 3, 13, 23, 18, 6, 380, 7, 2, 702, 1231, 11, 16, 2455, 15]
BERT encoding	[101, 1996, 6614, 1997, 2023, 2817, 2001, 2000, 8556, 1006, 1037, 1007, 1996, 14260, 23391, 2008]

Table 2: Comparison of tokenization and encoding

After the tokenization, the words constructing the abstracts are counted in order to build a vocabulary and then the sequences are padded or cut in order to obtain the same length for all abstracts. Finally, the dataset is split into training, testing and validating sets to report the model's performance.

Task 2 - Implementation of LSTM and BERT

The LSTM model is built according to what was seen in class. The class contains a forward pass and a fit/predict function. During the initial trials of the LSTM, a lot of overfitting was observed. To mitigate this issue, the team implemented factors such as L1/L2 regularization, dropout and an Adam optimizer. These allowed the reduction of the gap between testing and validating accuracies.

As for the BERT model, the model was pretrained BERT model. The class then fine tunes the model to the dataset and uses AdamW, a scheduler. The model also has early stopping implemented. The scheduler allows varying the learning rate (α) in order to optimize α at every epoch.

Embeddings on the model were done using GloVe, this embedding method uses global word co-occurrence statistics. The specific embedding that is used here is GloVe300d that consists of 220B tokens, 1.2M vocab, uncased and 300d vectors.

Hyperparameter Choice

In this section, the hyperparameter for the models were probed and the best performing hyperparameter were identified in order to build future models. Figure 1 shows the different hyperparameter for both types of models reported. It is of interest to note that the dropout and hidden states led to better results and that the learning rate showed increasing accuracies in some cases.

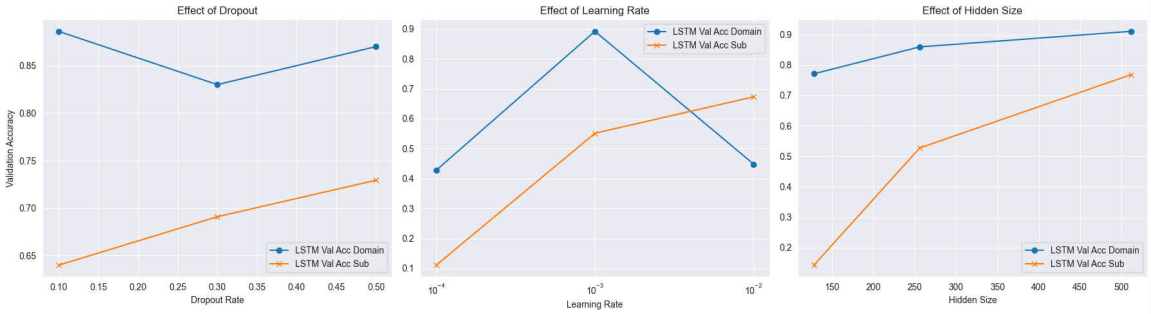

Figure 1: Effect of hyperparameter on LSTM

Task 3 - Experiments

In this section, various tests and figures are reported to visualize the performance of the models. Figure 3 shows the accuracy results for the LSTM and BERT models for the parent labels and the child labels. It seems that the most accurate model in both scenarios is the BERT model. Figure 4 shows the attention obtained through the sequence for the best correct and

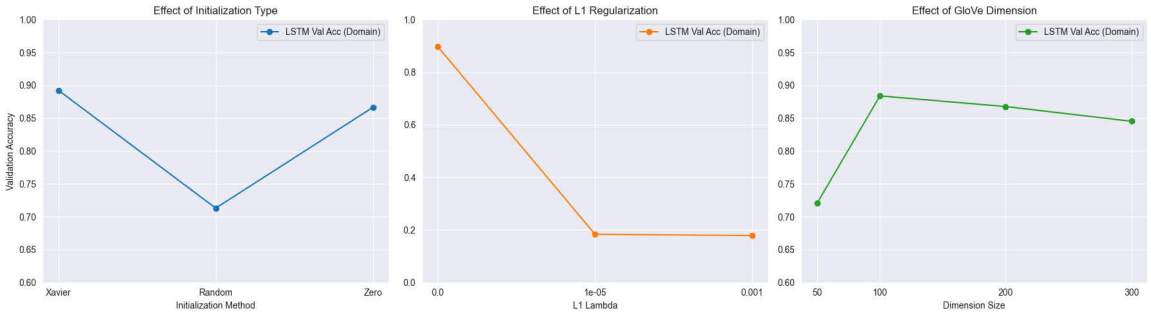

Figure 2: Further optimization of the LSTM model

wrongly predicted document. Out of all, the most accurate was the BERT model trained on WOS-11967 on the parent labels with an accuracy of 92.77%.

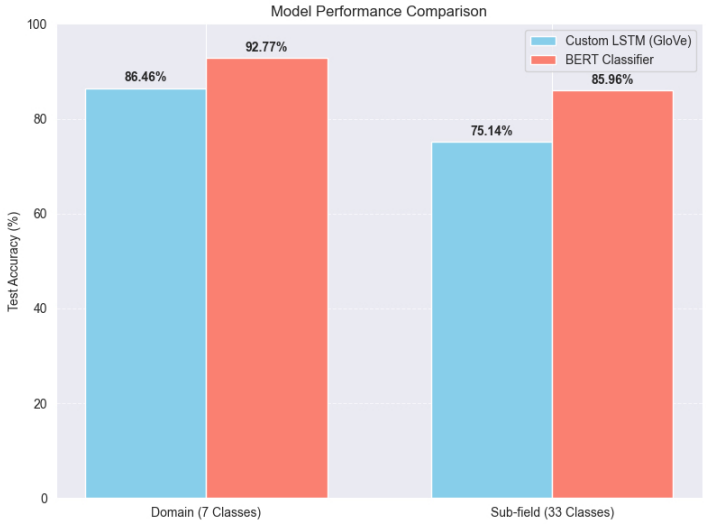

Figure 3: Model accuracies across various Datasets

Model	Task	Test Accuracy	Winner?
Custom LSTM (GloVe)	Domain (7 Classes)	86.46%	
BERT Classifier	Domain (7 Classes)	92.77%	←
Custom LSTM (GloVe)	Sub-field (33 Classes)	75.14%	
BERT Classifier	Sub-field (33 Classes)	85.96%	←

Table 3: Results Summary for the different models across the labeling

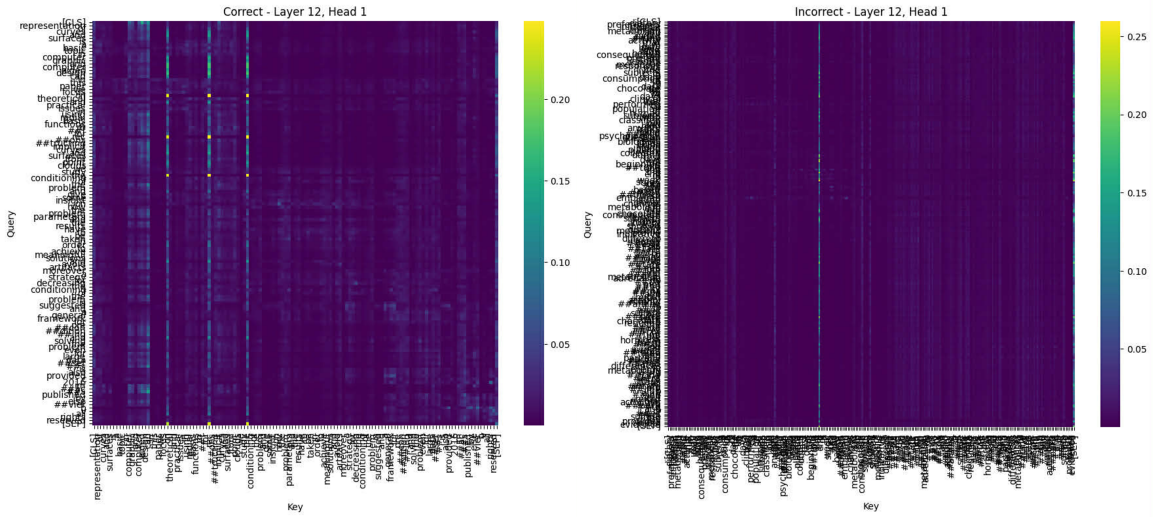

Figure 4: Attention map of Layer 12 Head 1

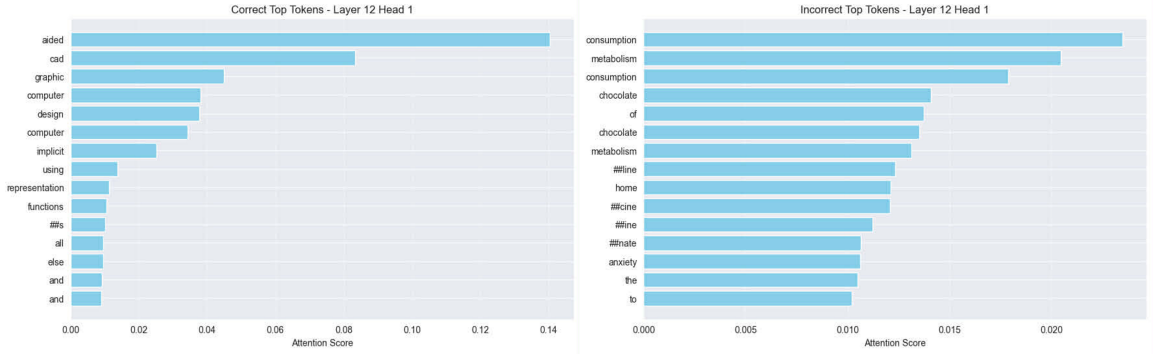

Figure 5: Highest scoring token for best and worst performing head

Looking at figure 6, it is possible to observe the false and true positive across all classes.

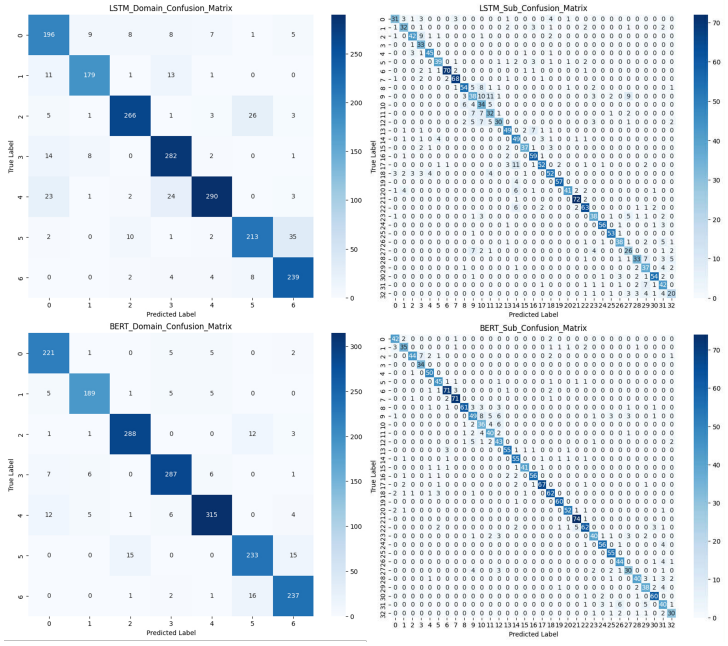

Figure 6: False and True positives across all classes for both models

Discussions and Conclusions

Overall, the main results have been observed from the previous sections. It seems that the BERT models were the most robust and accurate models. This makes sense, since BERT takes into its core Attention. This powerful mechanism allows the model to better understand the sequences, thus extracting more information out of it. Through the process, overfitting seems to have been an issue. It is possible to still see some signs of this problem, but some key components have been implemented to allow diminishing this issue such as dropout, learning rate modulation and size of hidden state. These allowed reducing the gap between the training error and the validation error, but it is not perfect. This factor could be that in the test sets that were used, since the abstracts between a field take a lot of inspiration across themselves, they could repeat the same sequences or close to it. This could lead to a bias and explain the risk of overfitting. Finally, the work presented above detailed the implementation and testing of an LSTM and BERT model across different datasets that were described above and the discrepancies across the performances of the models allowed understanding and commenting on the models' behavior. Looking at the attention matrix, it clearly shows that for the correctly predicted document, multiple tokens being important seem to have led to better results; it means that inferring importance on multiple words in the sequences leads to better results. The wrongly predicted document, shown in the same figure, illustrates this takeaway, where according importance to a smaller amount of tokens leads to worse results. In fact, figure 5 shows that for the accurate prediction, longer and more meaningful words have a higher attention score. As for the incorrect prediction, the words weighing in more seem to be less information rich; words like 'of' or 'the' do not carry the sentences. The scores are also higher on average. Finally, figure 6 shows the false and true positives. It seems for the LSTM, classes 5 and 6 are the most confused, which makes sense since they are Mechanical and engineering labels. For BERT, the same classes are confused. For the subcategories, it is quite impressive that the false classifying rate seems low. The categories should look a lot more alike, but it is still making sense as it forces the model to find more subtile differences in the abstract to get the subdomains. The main takeaway of this work is in the refinement and implementation of a machine learning model to solve a Seq2Vec problem case. The complexity of the problem is shown in this report in the form of the complexity of the implementation of the models themselves.

Additional work

Looking on to initialization, figure 2 shows the results across different initialization. The default model uses a 'Xavier' initialization that consists of keeping the variance of the activations roughly the same across layers; this then avoids exploding/vanishing gradients. Looking on to random initialization, the activations are picked randomly from a normal distribution and the zero initialization sets all weights initially to zero and the model learns from there. The best performing is Xavier with zero close behind. On the same figure, it is also possible to denote L1 regularization implementation that seems to make the model worse; this result makes sense as L1 seems to aggressively force the weights to be zero and the model responds badly. On the plot, GloVe dimension also can be observed. These results clearly highlight the plateau that the model hits and as the dimension increases, the model can fall into overfitting, thus lowering the accuracy. Figure 7 visualizes the learned feature space of the LSTM and BERT models using t-SNE (a non-linear dimensionality reduction technique) [1]. The clear separation of clusters demonstrates that the models were able to differentiate the scientific fields.

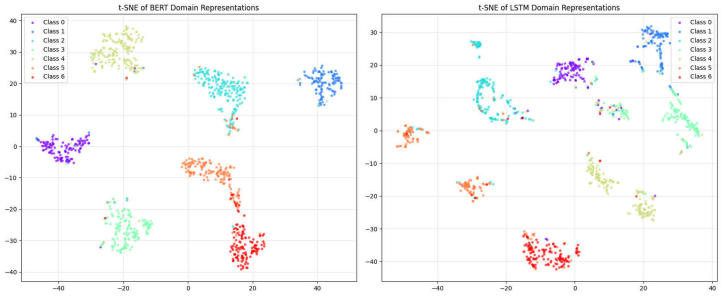

Figure 7: Clustering of classes for the two models

Statement of Contribution

Gabriel Caballero: Data retrieving and cleaning, task 1, 2 and 3 coding, model implementation and optimization, additional work section implementation, figures and tables in the report.

Adam Dufour: Retrieving and cleaning of the datasets, model implementation and writing the report, abstract, intro, datasets, discussions and conclusions.

References

[1] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840, 2002.

[2] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[3] 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 *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE, 2017.

[4] 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)*, pages 364–371, 2017.