

Introduction

- Two **models** were developed to solve Natural Language Inference (NLI), the task of assessing whether a premise semantically supports a hypothesis
- Method B focused on using RNNs whilst C finetuned pretrained Transformer architectures

Methods

- Method B:**
 - Trained a **Bi-LSTM** with frozen XLNET embeddings
 - Utilised **learning rate scheduling** to approach a global optimum
 - LR was reduced on metric plateau
 - Employed **subtractive & multiplicative sentence fusion, & attention** to enhance sentence representations
- Method C:**
 - Finetuned the base **ROBERTA** Transformer model
 - Employed **data augmentation** to bolster the training data:
 - Synonym replacements & insertions
 - Word deletions & swaps
 - Used **early stopping** and LR scheduling to reduce overfitting
 - LR warmup and then decreasing was used

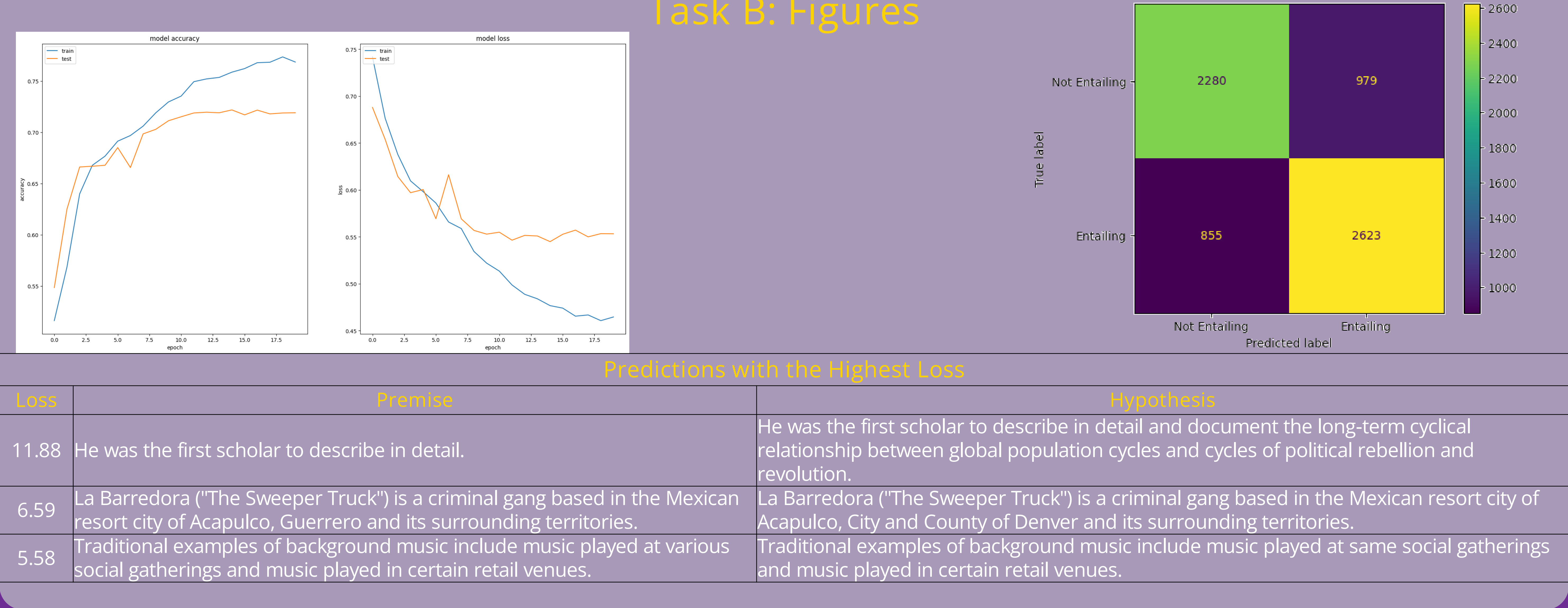
Conclusions

- Pretrained word embeddings** are key to higher performance
- Transformers** achieve a **higher performance** but also loss, compared with RNNs
- Likely due to low output scores from RNNs
- Basic textual augmentations do not significantly improve performance

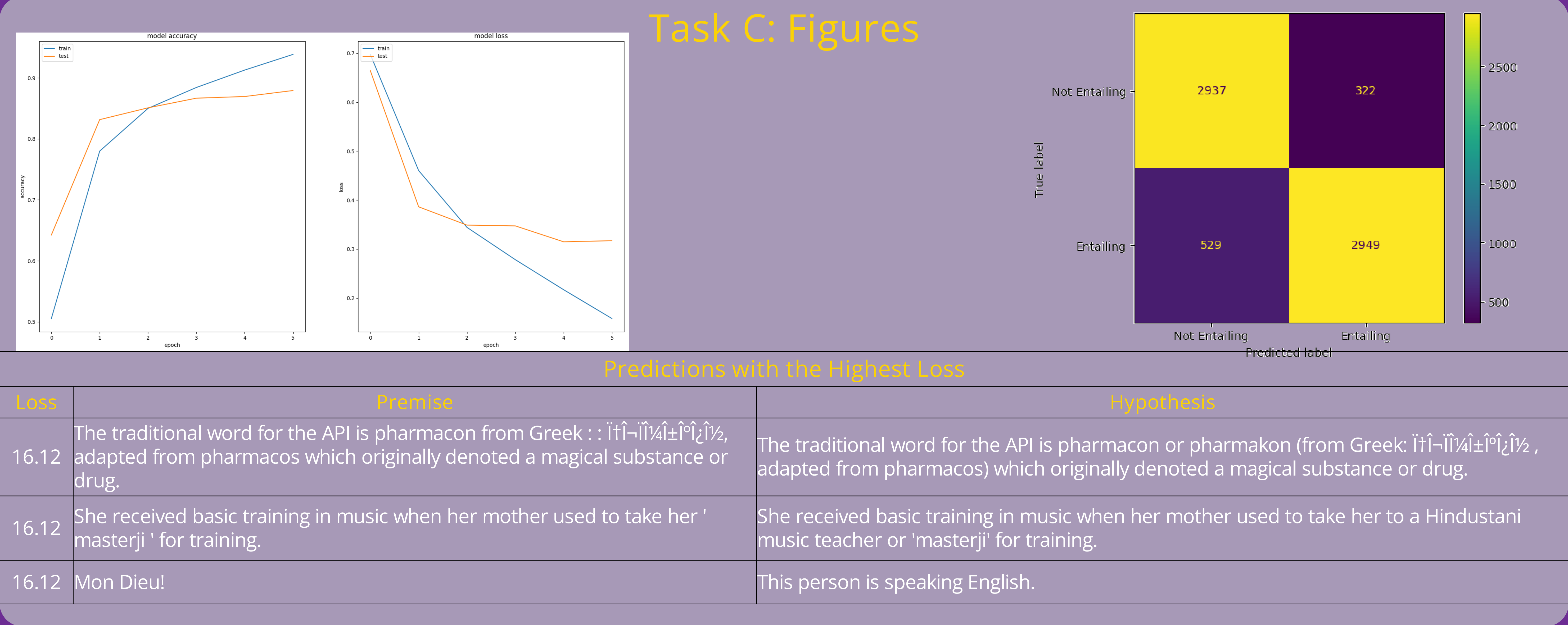
References

- S. Zhang, S. Liu and M. Liu, "Natural language inference using LSTM model with sentence fusion," 2017 36th Chinese Control Conference (CCC), Dalian, China, 2017, pp. 11081-11085, doi: 10.23919/ChiCC.2017.8029126
- Tarunesh, Ishan, Somak Aditya, and Monojit Choudhury. "Trusting roberta over bert: Insights from checklistng the natural language inference task." arXiv preprint arXiv:2107.07229 (2021).
- Sadat, Mobashir, and Cornelia Caragea. "Scinli: A corpus for natural language inference on scientific text." arXiv preprint arXiv:2203.06728 (2022).

Task B: Figures



Task C: Figures



Optimal Results

Method	Accuracy	Precision	Macro Precision	Weighted Macro Precision	Recall	Macro Recall	Weighted Macro Recall	F1-Score	Macro F1-Score	Weighted Macro F1-Score	MCC	Loss
B	0.728	0.728	0.728	0.728	0.754	0.727	0.728	0.741	0.727	0.728	0.455	0.535
C	0.874	0.902	0.874	0.875	0.848	0.875	0.874	0.874	0.874	0.874	0.749	1.849