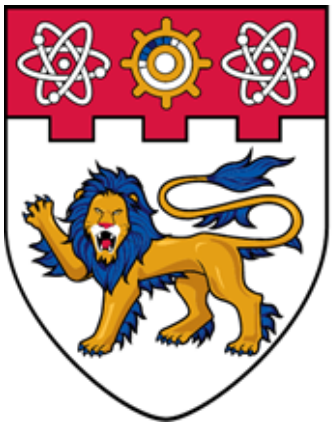

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ4042 - Neural network and Deep learning

Project Title: Evaluation of Transformers on Sentiment Analysis for Movie Reviews

AY23/34 Semester 1

Date of Submission: 8/11/23

Lee Yew Chuan Michael, U2021372J

Chua Wen Rong Jonathan, U2021875L

Foo Zhi Kai, U2022416G

Table of content

1. Introduction

The introduction of Transformers has immensely revolutionised the field of machine learning. Since its introduction by Google in the paper “Attention is all you need” [1], many other alternative Transformer architectures have been proposed. This includes making the models more lightweight, altering the training objective, and stacking more layers in an attempt to obtain better contextual understanding. These developments have led to many groundbreaking developments, not just in the field of Natural Language Processing (NLP), but also in the realms of Computer Vision. Thus, it is reasonable to conclude that Transformers are and will continue to be the leading force in machine learning for the years to come. With the multitude of Transformer architectures available, there seems to be an abundance of solutions for users to choose from. However, we argue that this is a misconception. Despite the myriad of solutions available, careful consideration must be given to model selection when solving a given task. Whilst one model might be more suited for Text Sentiment Analysis (TSA), the same model might not be suited for Text Summarisation. Additionally, even within a task itself, the dataset used also plays a part in architecture selection. To put it simply, for every application, there exists a best Transformer architecture that is most suited for it. Beyond the issue of model selection, we also highlight two common pitfalls that people often do not consider. This includes the lack of explainability of models and the vulnerability of Deep Neural Networks (DNNs) to adversarial examples [2].

In this project, we aim to evaluate different Transformer architectures to identify the best architecture for TSA. In particular, TSA in the context of movie reviews. For an accurate representation of sentiment analysis for movie reviews, we utilise the IMDb movie review dataset [3]. To facilitate a comprehensive and comparative analysis, we evaluate the different architectures using the following metrics: Training time, Evaluation time, Accuracy, F1-Score and Total number of trainable parameters. Through analysis and interpretation of these metrics, this project endeavours to provide pivotal insights into the appropriateness and applicability of various Transformer architectures for TSA within the domain of movie reviews.

In addition to this evaluation, we design an ensemble model architecture that combines BERT [4] with Bidirectional Long Short-Term Memory (BiLSTM) to enhance results. This combination leverages BERT's proficiency in learning contextual embeddings, which is crucial for understanding the nuanced language often found in sentiment-laden texts. Following this, we introduce the notion of explainability which helps to avoid the oversimplification of seeing models as merely black boxes. Through implementing explainability, we can make the model's decision understandable to humans, helping to build trust between humans. We also introduce the concept of adversarial examples on DNNs to highlight the dangers of such attacks and the need for DNNs to be defended. Doing so prevents adversaries from achieving attack success. Through this project, we aim to contribute to the academic discourse surrounding these models and assist practitioners and businesses in making informed decisions when implementing TSA tools.

2. Literature Review

2.1. Transformers

The key driver behind the Transformer architecture is the use of attention for modelling long-range dependencies in data [1]. Attention is a mechanism which allows a model to make predictions by attending to a set of training data. Transformers use self-attention to perform predictions for one part of the data sample based on observations from different parts of the same data sample. This provides Transformers with the ability to preserve the whole context of the data sample. Furthermore, Transformers are more efficient to train due to their parallel nature, as compared to other deep learning architectures.

2.1.1. Transformer Architectures

We evaluate six different Transformer models for TSA on IMDb movie reviews. This includes BERT [4], DistilBERT [5], ELECTRA [6], BART [7], XLNet [8], and GPT-2 [9]. BERT and DistilBERT provide a bidirectional contextual understanding, with DistilBERT offering a more efficient alternative. ELECTRA introduces an innovative training method akin to a Generative Adversarial Network (GAN), potentially enhancing learning efficiency. BART's hybrid structure combines the best of both worlds, bidirectional and autoregressive, making it versatile for various tasks. XLNet's permutation-based training and handling of longer sequences add another dimension to the analysis. Lastly, GPT-2's autoregressive nature and prowess in text generation offer a distinct perspective. This mix not only covers a broad spectrum of Transformer-based approaches but also ensures a comprehensive analysis of how different architectures perform in TSA, considering their unique mechanisms of understanding and generating text.

i. BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) [4] is a powerful pre-trained model with innovative features. Instead of encoding inputs unidirectionally, BERT encodes them bidirectionally, capturing contextual information from both directions. This enhances BERT's contextual understanding, due to its learned highly contextualised representations. In addition to bidirectional learning, BERT was also trained based on the "masked language mode" (MLM) objective and the next sentence prediction task. MLM involves randomly masking input tokens and requiring the model to predict the masked words using only surrounding context, encouraging better contextual representations. These powerful pre-trained representations makes BERT a suitable choice for a host of tasks. This includes TSA and text summarisation.

ii. DistilBERT

Despite the great performances of Transformers, these models tend to be computationally intensive. This is a problem when we want to deploy these models in Internet of Things (IoT) devices which tend to be resource-constrained. DistilBERT [5] is a distilled version of BERT which leverages on knowledge distillation during the pre-training phase. Knowledge distillation is a technique used to transfer knowledge from a larger, more complex model (often referred to as the "teacher" model) to a smaller, more efficient model (known as the "student" model). The goal is to retain as much of the performance of the larger model as possible while reducing the computational resources required by the smaller model. Such a lightweight model will enable us to alleviate resource constraint issues in IoT devices. DistilBERT is an example of this technique applied to the BERT model. Despite its smaller size, DistilBERT retains about 95% of BERT's performance on several benchmark tasks. This is a testament to the effectiveness of knowledge distillation in preserving the essential knowledge from the teacher model.

iii. ELECTRA

Unlike BERT, which is trained based on the MLM objective, ELECTRA [6] introduces a new approach which borrows concepts from Generative Adversarial Networks (GANs). Namely, ELECTRA trains two Transformer models, a generator and a discriminator. Instead of masking words, ELECTRA uses the generator to substitute tokens in a sequence with reasonable alternatives. Subsequently, the role of the discriminator is to identify which tokens were swapped out by the generator in the sequence. By adopting this approach, ELECTRA achieves greater pre-training efficiency compared to MLM. Unlike the masking approach of MLM, ELECTRA's task encompasses all input tokens. Consequently, this innovative training framework results in more contextually rich representations than those obtained with BERT.

iv. BART

Similar to BERT, the encoder in BART [7] processes the input text in a bidirectional manner. It takes the entire sequence of words and computes representations by considering the context from both sides of each word. However, it uses an autoregressive decoder for generating sentiment predictions. The autoregressive decoder generates output one token at a time, using the previously generated tokens as context. In TSA, this can be leveraged to generate coherent and contextually relevant sentiment predictions. The sequential nature of the decoder helps in handling ambiguous cases in TSA, where the sentiment might not be clear until the end of a sentence or paragraph.

v. XLNet

XLNet [8] employs permutation-based training to understand context. XLNet's permutation-based training allows it to predict each word in a sentence based on all possible positions, not just the previous ones. This approach enables XLNet to capture a more comprehensive and nuanced understanding of context, which is crucial in TSA where the sentiment can be influenced by words located anywhere in the text.

Moreover, XLNet integrates the Transformer-XL [10] mechanism, which is adept at handling long-range dependencies. This feature is particularly useful in TSA when dealing with lengthy text, such as detailed movie reviews or blog posts, where the sentiment might be spread across several sentences or paragraphs. The ability to process long sequences allows XLNet to capture subtle sentiments that might be expressed over extended text. It can effectively connect sentiments expressed at the beginning of the text with those at the end, ensuring a comprehensive sentiment analysis.

vi. GPT2

GPT-2 (Generative Pre-trained Transformer 2) [9] is an autoregressive language mode. GPT-2 predicts each token based on the sequence of tokens that came before it. This autoregressive nature allows GPT-2 to understand the flow and progression of sentiments in a text, which is crucial in TSA where the context and order of words can significantly influence the sentiment conveyed. While GPT-2 is more suited for the task of text generation, its ability at language comprehension enables it to understand language nuances, potentially beneficial for detecting varied sentiment expressions in the task of TSA. Additionally, its autoregressive nature allows for sequential contextualization, which is crucial for understanding sentiments that unfold over texts.

2.1.2. Differences Between the Architectures

i. Training and Learning Approaches

BERT and DistilBERT uses MLM, ELECTRA uses a GAN-like approach, and XLNet employs permutation-based training. GPT-2's autoregressive nature sets it apart in terms of text generation.

ii. Model Size and Computational Efficiency

DistilBERT stands out for its efficiency and smaller size. In contrast, models like BART, GPT-2 and XLNet are much larger and require more computational resources.

iii. Bidirectionality vs. Autoregressiveness

BERT, DistilBERT, ELECTRA, and BART are bidirectional, providing a more comprehensive understanding of context. GPT-2 and XLNet, while also handling context effectively, are autoregressive and sequential in their approach.

iv. Versatility and Task Suitability

BART's hybrid nature makes it versatile for various tasks. In contrast, GPT-2's strength lies in text generation due to its autoregressive nature.

v. Contextual Understanding and Sequence Handling

Each model has a unique way of handling context and sequence length. XLNet's integration of Transformer-XL helps it manage longer sequences effectively, which might be beneficial in analysing lengthy movie reviews.

2.2. Explainability

While DNNs have achieved state-of-the-art (SOTA) results, a key factor that is still lacking is the explainability of DNNs. To establish trust between humans and Artificial Intelligence (AI)-driven systems, it is crucial to dispel the misconception of treating DNNs as black boxes. This is particularly vital when integrating AI into critical systems in the banking or cybersecurity sectors. To make the decisions of DNNs understandable to humans, much research has been done in the field of explainable AI (XAI) [11]–[13]. These methods often involve calculating the contribution of each input feature to the prediction. Comprehending the individual contributions of input features to the DNN's predictions and its decision-making process fosters a deeper level of trust between humans and AI systems. This enhanced trust will give individuals more confidence to deploy such systems in real-world applications.

2.3. Adversarial Examples

Despite the tremendous capabilities that DNNs display, DNNs are vulnerable to the imperceptible perturbations of inputs. Such attacks aim to cause DNNs to misbehave and make wrong decisions, leading to catastrophic consequences. While this vulnerability was originally demonstrated for image classifiers, Transformers have also been found to be vulnerable to adversarial examples [14]–[16]. When applied to the context of TSA, such attacks can cause the model to misclassify 'Positive" reviews as "Negative" and vice versa. This can cause many disruptions if undefended models are used in production. Thus, when building a DNN to fulfil a certain task, one should always take into consideration the threats posed by adversarial examples and develop the appropriate defences to counter them.

3. Method

3.1. Dataset

In this project, the IMDb movie review dataset [3] was used. This is a dataset tailored for the task of binary sentiment analysis. It features 50,000 reviews from the Internet Movie Database (IMDb) which are labelled either as ‘Positive’ or ‘Negative’. In this dataset, the number of positive and negative reviews are 25,000 each. This makes it ideal for our task of evaluating Transformers. A balanced dataset allows us to achieve better results when training the models, preventing the models from becoming too biased towards a certain class. This allows us to make a fairer comparison when evaluating the different Transformers via the metrics. In addition to a balanced dataset, further processing was done to ensure that a maximum of 30 reviews were included for each movie. This ensured that no biasness would be unnecessarily introduced to the dataset.

3.2. Training Configurations

Table 1: Hyperparameters used for Training the Transformers

Hyperparameters	Values	Descriptions
Epochs	3	We trained each of the Transformers for 3 epochs. We found this to be the optimal setting. Training for more epochs leads to overtraining.
Train-Test Split	7:3	To train and test each Transformer model, we split the dataset into 70% for training and 30% for testing. Splitting the dataset helps ensure the model works well not just on the data it learned from but also on new data it has not seen before.
Tokenization	150	We set the maximum input token length for each Transformer model to be 150. This ensures that important information is retained and at the same time adheres to the constraint of limited memory.
Batch Size	16	In our experiments, we found batch size 16 to be the optimal setting, as it finds a good compromise between training stability and computational limitations. It is not too small as to cause instability during learning, and also not too big such that too much computer memory is utilised.

3.3. Evaluation Metrics

Evaluation Time

This measures the time taken by the model to evaluate and make predictions on new and unseen data. A model with a shorter evaluation time is often preferred, especially in applications requiring real-time analysis, such as monitoring social media sentiment or customer feedback.

Accuracy

Accuracy is the ratio of correctly predicted instances to the total instances. High accuracy is indicative of a model's ability to correctly classify the sentiment of texts. However, it is important to

note that accuracy alone might not always be the best metric, especially in imbalanced datasets, where one class significantly outnumbers the other.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100\%$$

F1-score

The F1-score is particularly important in classification tasks as it provides a balance between precision and recall:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

This metric is crucial in scenarios where both false positives (FP) and false negatives (FN) have significant implications. It's especially useful in dealing with imbalanced datasets, a common occurrence in TSA tasks.

3.4. Explainability

To perform explainability, we use Local Interpretable Model-Agnostic Explanations (LIME) [13]. LIME is a model-agnostic technique that helps to explain the predictions made by machine learning models. Model agnostic methods are more useful as they are applicable regardless of the model architectures used. The goal of LIME is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier. By applying such techniques, we can unpack the inner workings of the model to a human level of understanding and interpretability.

In the case of TSA, LIME highlights words (features) that contribute to each class label (positive or negative). With LIME, we can visualise, explain, and interpret the decision-making process of our model.

3.5. Adversarial Attacks

In our experiments, we utilise the Attacking to Training (A2T) attack [16], more specifically the A2T-MLM variant. This method introduces a more efficient technique for generating adversarial examples compared to other attacks. Traditional methods such as TextFooler [15] generate adversarial examples by first iteratively replacing words to determine the relative order of word importance (deletion-based ordering) which is computationally expensive and slow. On the other hand, A2T-MLM constructs adversarial examples by first ordering words by importance using the gradient of the loss. Using gradient-ordering greatly reduces the search space, allowing for faster generation of adversarial examples. To conduct the attack, we borrow the implementation from *TextAttack* [17] to generate our adversarial examples. To demonstrate this attack, we select 2500 samples from our test dataset to be generated as adversarial examples.

4. Experiments

4.1. Evaluation on Transformers

To compare the performance of different Transformer architectures for the task of TSA for the IMDb movie review dataset, we utilise pre-trained models from the *HuggingFace* library to perform pre-training using the hyperparameters defined in Section 3.2. This was done for each of the Transformer models as described in Section 2.1.1. Following the training phase, we then evaluate each model's performance using the metrics described in Section 3.3.

4.1.1. Results and Analysis

We report the evaluation metrics obtained from our experiments in Table 2.

Table 2: Metrics Obtained from Evaluation

Model	Evaluation Time (s)	Accuracy (%)	F1-score	Trainable Parameters
BERT	208.02	88.0	0.88	109,483,778
DistilBERT	**105.27**	86.0	0.88	**66,955,779**
ELECTRA	207.69	**90.0**	**0.90**	109,483,778
BART	209.96	89.0	0.89	**139,421,185**
XLNet	284.67	**90.0**	**0.90**	117,311,235
GPT2	172.95	87.0	0.86	124,441,344

From a glance, XLNet and ELECTRA are the best-performing models based on both F1-score and Accuracy. Both models achieved the highest measure out of the rest of the models evaluated. On the other hand, GPT2 was the worst-performing model based on F1-score.

Performing a comparison based on evaluation time, we observe that DistilBERT was the fastest-performing model, being able to perform evaluation on the test dataset in just 105.27 seconds. It is consistently faster than the rest of the models and can perform evaluation almost twice as fast compared to ELECTRA, BERT and BART. This phenomenon is consistent with the fact that DistilBERT was built with an emphasis on lightweightness, as evident by it having the lowest number of trainable parameters (66,955,779) compared to the rest of the models. The slowest model of the models evaluated was XLNet, which took almost three times the amount of time it took DistilBERT to perform evaluation. However, despite XLNet being the slowest model (284.67s), it was not the model with the highest number of trainable parameters. The model with the highest number of trainable parameters was BART. Thus, the high evaluation time of XLNet can be attributed to the training objective that XLNet adopts. XLNet learns through permutations to obtain better contextual representations. While this could have allowed it to obtain better prediction performance, it also likely contributed to the increased evaluation time. This observation showcases the presence of a tradeoff between evaluation time and the F1-score. A similar tradeoff can also be observed in the case of DistilBERT. Whilst DistilBERT proves to be the best-performing model in terms of evaluation time, it ranks third in terms of F1-score.

Besides the presence of such a tradeoff, we also note that besides the number of training parameters, the training objective of the respective model architectures also contributes significantly to the efficiency of the model. Whilst XLNet was not the model with the highest number of training parameters, its training objective likely introduced large overheads that led to it being the slowest-performing model in terms of evaluation time. From this, we deduce that whilst a more robust training objective to enable better contextual understandings can be used, this can lead to increased evaluation time, making the model less efficient. On the other hand, more relaxed training objectives can lead to faster evaluation time, with a sacrifice on contextual understandings and less accurate representations. From these findings, we conclude that model selection necessitates considerations of the training objective of the model, F1-score, and evaluation time of the model. Careful consideration of these factors and the weighing of these factors against each other can allow one to arrive at the optimal model architecture for the given task. For our task of TSA on the IMDb movie review dataset, we infer that DistilBERT is the most optimal model for this use case. Whilst it was not the best performer in terms of F1-score, it was just 0.02 off the best-performing model. Furthermore, the efficiency that DistilBERT offers far outweighs the slight difference in F1-score (0.02).

However, despite the great performance obtained with ELECTRA and XLNET, we believe that further improvements can be achieved. A higher performance was sought. To do so, we turned towards building an ensemble model, which will be further elaborated on in the next subsection.

4.2. BERT-BiLSTM

After testing the efficacy of the BERT model for TSA, it became evident that while BERT provides a strong foundation, there is still room for improvement. To attain further improvement in performance, we introduce an ensemble model that combines BERT with Bidirectional Long Short-Term Memory (BiLSTM). Figure 1 shows an overview of our ensemble model. This combination harnesses BERT's proficiency in generating contextual embeddings, which is crucial for grasping the nuanced language often found in sentiment-laden texts. Concurrently, BiLSTM was chosen to learn the output of the BERT layer to enhance the fitting effect of network features and generalisation to unseen datasets [18]. This amalgamation aims to provide a more comprehensive analysis of sentiments, effectively addressing scenarios where sentiments are intricately woven into the language or unfold over the course of the text.

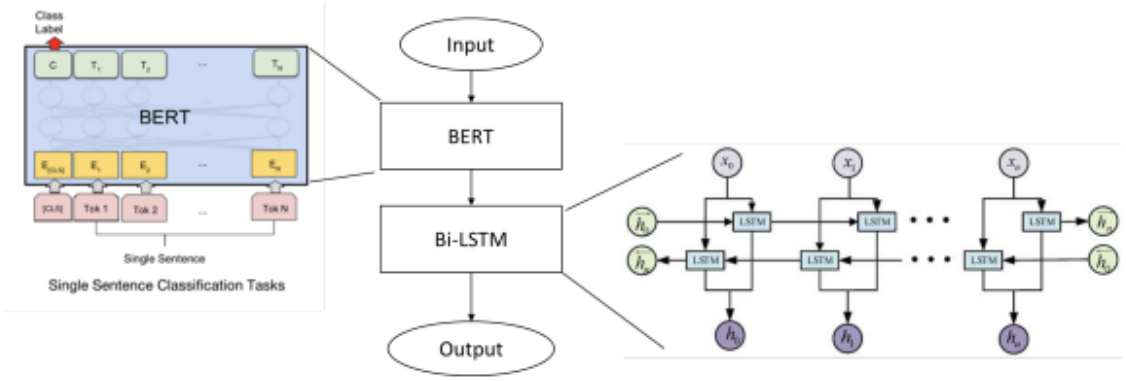

Figure 1: Overview of BERT with BI-BiLSTM [18]

4.2.1. Results and Analysis

Table 3: *Metrics Obtained from Evaluation of BERT-BiLSTM*

Model	Evaluation Time (s)	Accuracy (%)	F1-score	Trainable Parameters
DistilBERT	**105.27**	86.0	0.88	66,955,779
BERT	208.02	88.0	0.88	109,483,778
BERT-BiLSTM	220.24	**96.0**	**0.96**	**111,584,514**

From Table 3, BERT-BiLSTM was found to perform significantly better in terms of both accuracy and F1-score when compared to the other models. While BERT and DistilBERT yielded the same F1-score, BERT-BiLSTM performed 8% better. We attribute this phenomenon to the fact that BERT-BiLSTM is an ensemble model. Ensemble models tend to perform better, as they enable us to bring together the strengths of the models involved. In our experiments, integrating BERT with BiLSTM could have enabled us to capitalise on BERT's ability to generate contextual embeddings and BiLSTM's proficiency in capturing sequential dependencies. This could have contributed to a more comprehensive understanding of sentiments in the movie reviews.

Despite achieving the highest F1-score, the evaluation time and the number of trainable parameters for BERT-BiLSTM is the longest and the largest respectively when compared to BERT and DistilBERT. Whilst this difference is rather significant in the case of DistilBERT, the difference when compared with BERT is insignificant. From this, we conclude that BERT-BiLSTM outshines BERT as the significant improvement in F1-score by BERT-BiLSTM far outweighs the slight decrease in evaluation time that BERT offers.

However, when comparing DistilBERT and BERT-BiLSTM it is not so straightforward. In scenarios where high prediction accuracy is valued more than speed of prediction, we conclude that BERT-BiLSTM is the better model to use. However, in the case that the prediction needs to be fast and made in real-time, DistilBERT will be the far better choice. DistilBERT’s ability to perform almost twice as fast as BERT-BiLSTM gives it the edge in this regard.

4.2.2. Explainability

To demonstrate how explainability can be incorporated, we use LIME to explain the predictions from our BERT-BiLSTM model. We applied it to two movie reviews; one with positive and one with negative sentiment. We display the output from LIME in Figure 2. From the output of the negative review, we can observe that LIME deemed words such as “falls”, “disappointment” and “uninspiring” as crucial in leading our BERT-BiLSTM model to convey negative sentiments. In the case of the positive review, LIME interpreted words such as “breathtaking”, “dazzles” and “must-see” as words that contributed to our model predicting the review as having a positive sentiment.

The words identified largely correspond to what humans deem as having negative and positive connotations. This illustrates that techniques such as LIME can indeed enable us to get humanly interpretable explanations that explain how our BERT-BiLSTM model arrived at the corresponding predictions for the reviews. It is thus worthwhile to consider such explainability techniques when deploying models in real-world applications.

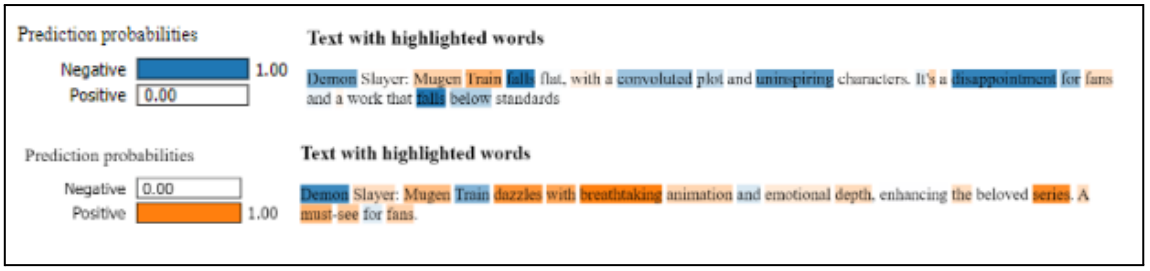

Figure 2: Applying LIME explainability on example movie reviews

4.2.3. Adversarial Examples

To highlight the importance of considering the vulnerability of DNNs to adversarial examples and the need for defences against them, we perform the A2T-MLM attacks on our undefended BERT-BiLSTM model. To reduce computational time, we select only 2500 samples from our test dataset and perform the attack on them. We report the success rate of the attack in Table 4. From this, we can see that the attack is able to fool our BERT-BiLSTM model about half the time (51.96%). While this might seem low, we deem impacting the model to misbehave half the time in a real-world use case to be a significant enough disruption. This serves to show how vulnerable an undefended DNN is and highlights the need for defences to be in place when using DNNs in real-world applications.

To counter adversarial examples, we recommend applying defences such as adversarial training. Adversarial training is a defensive framework which uses adversarial examples in tandem with clean samples as training data to train a model. In doing so, the trained model will become more robust towards adversarially perturbed samples. While this defence in itself might not be sufficient to entirely secure DNNs, it can alleviate the vulnerability of DNNs to adversarial examples. In our use case, this will mean that our BERT-BiLSTM model would become more resistant to adversarial examples, and would be able to still predict the correct sentiment even when it is presented with a perturbed sentence by an adversary.

Table 4: Adversarial Effectiveness of A2T-MLM

Successful (%)	Failed (%)	Skipped (%)
51.96	38.56	9.48

5. Conclusion

In this project, we highlighted the importance of giving consideration to which Transformer architecture to adopt when presented with a given task. In particular, we considered this for the task of TSA on the IMDb movie review dataset. We evaluated six different Transformer models to find out which model was more suitable for this particular task. After which, to improve results, we presented an ensemble model termed BERT-BiLSTM which outperformed all the other Transformers previously evaluated. Following this, we illustrated the importance of considering the factors of explainability and the vulnerability of DNNs to adversarial examples when DNNs are to be used in real-world applications. This was done by showing how explainability can be achieved and how it can aid our model. To illustrate the vulnerability of DNNs, we ran the A2T-MLM attack on our undefended model to demonstrate how vulnerable it was. We followed this up by suggesting adversarial training as a defence to counter such attacks.

References

[1] A. Vaswani *et al.*, ‘Attention is All you Need’.

[2] C. Szegedy *et al.*, ‘Intriguing properties of neural networks’. arXiv, Feb. 19, 2014. Accessed: Feb. 11, 2023. [Online]. Available: http://arxiv.org/abs/1312.6199

[3] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, ‘Learning Word Vectors for Sentiment Analysis’, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds., Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. Accessed: Nov. 06, 2023. [Online]. Available: https://aclanthology.org/P11-1015

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’. arXiv, May 24, 2019. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/1810.04805

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, ‘DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter’. arXiv, Feb. 29, 2020. doi: 10.48550/arXiv.1910.01108.

[6] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, ‘ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators’. arXiv, Mar. 23, 2020. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/2003.10555

[7] M. Lewis *et al.*, ‘BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension’. arXiv, Oct. 29, 2019. doi: 10.48550/arXiv.1910.13461.

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, ‘XLNet: Generalized Autoregressive Pretraining for Language Understanding’. arXiv, Jan. 02, 2020. doi: 10.48550/arXiv.1906.08237.

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, ‘Language Models are Unsupervised Multitask Learners’.

[10] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, ‘Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context’. arXiv, Jun. 02, 2019. doi: 10.48550/arXiv.1901.02860.

[11] N. Kokhlikyan *et al.*, ‘Captum: A unified and generic model interpretability

library for PyTorch'. arXiv, Sep. 16, 2020. doi: 10.48550/arXiv.2009.07896.

[12] S. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions'. arXiv, Nov. 24, 2017. doi: 10.48550/arXiv.1705.07874.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, '"Why Should I Trust You?": Explaining the Predictions of Any Classifier'. arXiv, Aug. 09, 2016. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/1602.04938

[14] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, 'BERT-ATTACK: Adversarial Attack Against BERT Using BERT'. arXiv, Oct. 01, 2020. doi: 10.48550/arXiv.2004.09984.

[15] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, 'Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment'. arXiv, Apr. 08, 2020. doi: 10.48550/arXiv.1907.11932.

[16] J. Y. Yoo and Y. Qi, 'Towards Improving Adversarial Training of NLP Models'. arXiv, Sep. 11, 2021. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/2109.00544

[17] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, 'TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP'. arXiv, Oct. 04, 2020. doi: 10.48550/arXiv.2005.05909.

[18] X. Li, Y. Lei, and S. Ji, 'BERT- and BiLSTM-Based Sentiment Analysis of Online Chinese Buzzwords', *Future Internet*, vol. 14, no. 11, Art. no. 11, Nov. 2022, doi: 10.3390/fi14110332.