Integrating Extra Linguistic Meaning into the BERT Framework

Stanford CS224N Custom Project

Riley Carlson
Department of Symbolic Systems
Stanford University
rileydc@stanford.edu

Bradley Moon
Department of Mathematics
Stanford University
bradmoon@stanford.edu

Ishaan Singh
Department of Mathematics
Stanford University
ishaanks@stanford.edu

Abstract

In this project, we aim to improve BERT's performance on advanced Natural Language Inference (NLI) tasks by integrating additional linguistic meanings that humans naturally interpret from the input text. We propose a method that generates presuppositions and implicatures from the input text and combines them with BERT's encodings using three different combination techniques: concatenation, adding, and SVD decomposition. We finetuned the T5 language model on the IMPPRES dataset to generate these extra linguistic meanings. We evaluate our model on three downstream tasks that would benefit from more nuanced linguistic meaning including Inference Classification, Irony/Sarcasm Detection, and Sentiment Analysis. Downstream task evaluation show modest improvements with the additive and concatenation techniques when incorporating the encodings from the presuppositions and implicatures, revealing the need for a more robust generation process.

1 Key Information to include

Mentor: Soumya Chatterjee; **External Collaborators**: N/A; **Sharing project:** N/A

2 Introduction

Problem: Traditional encoders like BERT (Devlin et al., 2018) struggle with understanding the meanings of sentences that are not explicitly conveyed, which is the basis of Natural Language Inference (NLI) tasks (Jeretic et al., 2020). These implicit meanings often rely on the context in which the sentence is uttered, logical implications of the sentence, and common knowledge shared between the speaker and the listener. While humans naturally infer the intended meaning behind words, BERT's performance is limited on tasks that require this deeper level of understanding.

Incorporating extra linguistic meaning is a difficult task, that even humans sometimes struggle with, because it requires the model to have access to a vast amount of world knowledge and the ability to reason about the context in which the sentence is uttered. Adding to this complexity is the fact that the interpretation of extra linguistic meaning is often not deterministic, as there may be multiple valid interpretations depending on the context and the individual's background knowledge, making it challenging to develop a model that consistently arrives at the intended meaning.

Background on Extra Linguistic Meaning: Two types of extra linguistic meaning we will focus on are implicatures and presuppositions. They are two fundamental types of extra linguistic meaning that significantly contribute to the overall understanding of a sentence, and incorporating them into BERT encodings has the potential to greatly improve the model's performance on tasks that require inferential reasoning. Implicatures arise when a speaker intentionally implies something beyond what is explicitly stated, often relying on the listener's ability to infer the intended meaning based on

the context and the Gricean Maxims of Conversation ((Grice, 1975), (Potts, 2015)). For example, when asked if she enjoyed the party, Riley replied, 'The music was loud,' implying that she did not enjoy the party without explicitly stating it. Presuppositions are a type of extra linguistic meaning that refers to the implicit assumptions or background information that a speaker takes for granted and assumes to be true when making a statement ((Levinson, 2000), (Partee et al., 1995)) . For example, when someone says, 'The King of Mexico is tall,' they are presupposing that Mexico has a king.

Overview of Approach: Our project focuses on incorporating extra linguistic meaning into BERT encodings to improve its performance on inference tasks. We propose generating presuppositional and implicature meanings using an LLM finetuned on the IMPPRES dataset (Jeretic et al., 2020). By combining the BERT-generated encodings of the original sentence with the encodings of the generated extra meanings, we aim to create a more comprehensive representation of the sentence's meaning which will enhance the model's ability to capture the intended meaning of sentences, leading to improved performance on NLI tasks.

3 Related Work

Jeretic et al. (2020) demonstrated the shortcomings of BERT in NLI tasks, which involve determining whether a given premise sentence entails, contradicts, or is neutral to a hypothesis sentence. To address this issue, they developed the IMPPRES dataset, consisting of 135K sentence pairs labeled with logical relationships (contradiction, entailment, and neutral) and whether the second sentence is a presupposition or implicature. The authors showed that incorporating knowledge of extra linguistic meaning by finetuning BERT on the IMPPRES dataset led to improved performance on downstream NLI tasks.

Our work builds upon the findings of Jeretic et al. (2020) but takes a different approach to incorporate extra linguistic meaning into BERT. Instead of finetuning BERT directly on the IMPPRES dataset we propose an architecture that generates separate encodings for the original sentence, presupposition, and implicature. We use the IMPPRES dataset as a training corpus to generate the presuppositions and implicatures will then be encoded by BERT. The generated encodings are then combined to create a comprehensive representation of the sentence's meaning, which includes both the original meaning and the extra linguistic meaning.

By explicitly considering presuppositions and implicatures separately from the original sentence, our approach aims to create a more balanced representation of the overall meaning. This method differs from the finetuning approach used by Jeretic et al. (2020), as it does not rely on the model learning to implicitly capture extra linguistic meaning during the finetuning process.

The explicit consideration of linguistic structure is shown to be successful by Bai et al. (2021). They demonstrate that generating syntactic information from the input sentence and combining it with the "naive" encodings from BERT leads to improved performance on Natural Language Understanding tasks. While Bai et al. (2021) use an attention mechanism to combine the syntax tree information with the naive encodings, we elect to use a simpler and more deterministic combination mechanism, such as SVD, concatenation, or element-wise addition. The reasoning for this is because presuppositions and implicatures are also represented as text, their encodings are comparable to those of the original sentence.

RoBERTa by Liu et al. (2019) is an optimized version of BERT that achieves state-of-the-art performance on various natural language understanding tasks. While RoBERTa does not explicitly incorporate extra linguistic information, it demonstrates the importance of improved training techniques and larger training data. RoBERTa is trained on a much larger corpus for more iterations compared to BERT, and it employs dynamic masking. These optimizations result in improved performance on downstream tasks, highlighting the potential benefits of enhancing the training process and increasing the amount of training data. However, the computational resources and training time required for RoBERTa are substantial. In our project, we aim to improve the performance of BERT on NLI tasks by incorporating extra linguistic meaning, such as presuppositions and implicatures, without relying on the extensive compute power and training effort used in RoBERTa. We hope that our approach can provide a more efficient way to enhance the performance of pretrained language models on these tasks.

4 Approach

4.1 Baselines

Our baseline model is BERT from the open-source HuggingFace library. We selected BERT as our baseline because it is a component of our proposed model, which will be discussed later. Using BERT as a baseline allows us to directly compare its performance with our proposed model to assess whether incorporating additional linguistic content improves performance in downstream tasks. We use the BERT tokenizer to generate encodings for the dataset of the specific task in question. These encodings are then used in further downstream tasks for evaluation. Unlike our primary models, the baseline does not incorporate the additional linguistic meanings described above; it solely processes the dataset associated with the given downstream task.

4.2 Paradigm

Our workflow can be described via the following diagram:

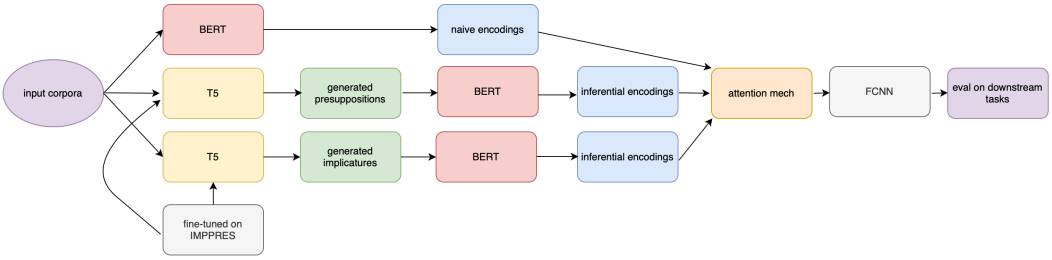

Figure 1: Our proposed model to integrate extra linguistic meaning into the BERT framework

As shown in Figure 1, the enhanced model is comprised of a three layer mechanism. The first layer naively takes in the input corpora (variable length sentences from the dataset associated with the downstream task) and generates encodings via BERT. The second layer takes in the input and, using T5 finetuned on the IMPPRES dataset, generates presuppositions, which are then passed into BERT to generate encodings. The third layer also takes in the input data and generates implicatures using the finetuned T5 model. The encodings from the three layers are then combined using various combination techniques, which we elaborate on later. These combined encodings are then used for evaluation on a variety of downstream tasks, by passing them into a two-layer fully connected neural network, to generate the labels. These downstream tasks are Irony/Sarcasm Detection, Sentiment Analysis, and Inference Classification. All of the above are tasks we believe our enhanced model will show improved performance ondue to the extra linguistic meaning now baked in.

Notice that our baseline makes use of the naive encodings. We also use a second, slightly simpler model, which combines the naive encodings with the encodings from the presuppositions. Because presuppositions are easier to identify than implicatures due to their weaker reliance on context, we believed it would be pertinent to include a model that solely incorporates presuppositions without implicatures. Our final model combines all three embeddings: naive encodings, presupposition encodings, and implicature encodings.

5 Experiments

In this section, we explain the various data used in our workflow, the evaluation metrics, and the training paradigm specifics.

5.1 Data

5.1.1 Data for LLM Finetuning

To finetune the LLM for generating presuppositions and implicatures, we used the IMPPRES dataset from Jeretic et al. (2020) as training data. We specifically selected the positive instances where the second sentence was an implicature or presupposition of the first sentence, resulting in a total of 4,500 sentence-presupposition pairs and 1,200 sentence-implicature pairs. We chose not to use the negative instances during finetuning because the T5 model, being an encoder-decoder model trained on a text-to-text format, is well-suited to learning from positive examples alone Raffel et al. (2023). By providing only positive examples, we aim to simplify the training process and allow the model to focus on learning the generative patterns of presuppositions and implicatures directly, without the need for explicitly distinguishing between valid and invalid instances. The T5 model's architecture and training objective enable it to capture the input-output relationships and generate the desired output accordingly. The data was split into an 70/10/20 train/val/test set.

5.1.2 Data for Downstream Tasks

Inference Classification: NLI is the task of determining the inferential relationship between two sentences. The dataset used for this task was from the Stanford Natural Language Inference CorpusBowman et al. (2015), which is hand-labeled based on what annotators believed the sentences to be in relation to one another and was in jsonl format. During preprocessing, all examples with a "negative" gold-label (meaning there was no majority consensus) were thrown out, leaving us with around 550,000 pairs of sentences for the training set and 10,000 for the testing set.

Irony/Sarcasm Detection: The objective of Sarcasm Detection is to classify a comment as to whether it is sarcastic or not. In this task, we obtained a balanced dataset of Reddit comments from Princeton NLP SAR (2017), which required conversion from jsonl into csv format. The final dataset contains 250,000 training examples and 60,000 testing examples. We believe that the style of comments would be more conversational in nature, and as such that this task would benefit from having extra linguistic meaning added.

Sentiment Analysis: Sentiment analysis is a well-known task where a review is classified as positive or negative. For our task, we used a Rotten Tomatoes Movie Review datasetPang and Lee (2005) available off of Hugging Face, which was balanced and contained 8,500 training examples and 1,000 testing examples. Due to the informal nature of the comments, we believed that adding extra linguistic meaning would make a difference here as well.

5.2 Evaluation method

5.2.1 LLM Evaluation

To evaluate the quality of LLM's presuppositions and implicature generations, we use BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) scores (specifically ROUGE1 and ROUGE2) against the model's generation and the true presupposition/implicature from the IMPPRES dataset. We will compare the results to those obtained using the pretrained model without any finetuning on the IMPPRES dataset.

5.2.2 Downstream Evaluation

Each of our downstream tasks (sentiment analysis, irony/sarcasm detection, and inference classification) are classification tasks (2-class, 2-class, and 3-class respectively). As such, our evaluation metric is given by percentage accuracy.

5.3 Experimental details

5.3.1 LLM Finetuning

We chose to use the T5 model by Raffel et al. (2023) to generate the implicatures and presuppositions of the input. Due to memory constraints, we used `t5-small` as the pretrained model for finetuning. We trained two separate models: one for generating presuppositions and another for generating implicatures. This decision was made because implicatures and presuppositions, despite seeming

similar at the surface level, require different linguistic tests to differentiate between them. Having a single model learn both presuppositions and implicatures simultaneously would put additional pressure on it to generate extra linguistic meaning while also learning to distinguish between the two concepts. Moreover, to simplify the use of the generated data in the rest of the model workflow and avoid the need for prompt engineering to format the model's output of extra linguistic meaning, we opted for separate models.

The presupposition model and the implicature model were both finetuned for 20 epochs with cross-entropy as the loss function, but with different learning rates. The presupposition model used a learning rate of 2×10^{-5}, while the implicature model used a learning rate of 1×10^{-4}. For both models, we used a batch size of 8 and applied a weight decay of 0.01 to regularize the training process and prevent overfitting.

5.3.2 Encoding Generation

As shown in Figure 1, there are three encodings produced in our model. Note that the encodings used are of the CLS token, in order to reduce variation based on the sequence length and to reduce the complexity of the model's output. As the CLS token can attend to the entire sequence and aggregate context about it, it is commonly used in downstream classification tasks, which is exactly what we use it for in this context.

Once the presuppositions and implicatures of the input sentence are generated by the LLM, these meanings are encoded via BERT. These two encodings are then combined with the encoding of the original sentence via BERT. We experiment with three various combination techniques.

The first and simplest method is simply concatenating the encodings, resulting in an encoding of size $\mathbb{R}^{3\times 768}$ where 768 is the encoding dimensions returned by the off-the-shelf BERT.

The second method involves element-wise addition of normalized encodings, resulting in a $\mathbb{R}^{768}$ encoding space. Each encoding $\mathbf{e}_i$ is normalized as

$$\hat{\mathbf{e}}_i = \frac{(\mathbf{e}_i - \mu(\mathbf{e}_i))}{\sigma(\mathbf{e}_i)}$$

We then sum $\hat{\mathbf{e}}_i + \hat{\mathbf{e}}_j + \hat{\mathbf{e}}_k$ to arrive at the final encoding, which we pass into our feedforward layer. In the third and last method, we stack the encodings into a matrix as shown below:

$$\mathcal{M} = \begin{bmatrix} \text{naive encodings} \\ \text{presupposition encodings} \\ \text{implicature encodings} \end{bmatrix}$$

We then compute the singular value decomposition (SVD) of this matrix. That is, we decompose $\mathcal{M}$ as the product of matrices $U\Sigma V^\top$ where U and V are orthogonal and Σ is diagonal. Such a decomposition exists for all real matrices. In particular, we can make our diagonal matrix such that its entries satisfy $\sigma_{11} \geq \sigma_{22} \geq \dots$. Now, we want the eigenvector corresponding to the largest eigenvalue, which in this context corresponds to taking the first left eigenvector (with eigenvalue σ_{11}), which is of size $\mathbb{R}^{768}$. This is what we use as our final encoding.

5.3.3 Downstream Fully-Connected Neural Network

After generation of these encodings, they are passed into a fully connected neural network, with ReLU activation between the layers. This activation is given by $\text{ReLU}(x) = \max(0, x)$. The output layer produces a vector with the number of entries equal to the number of classes for the task. We then apply a softmax function, given by

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

where z_i represents the logit for class i. This formula essentially converts each of the logits in the vector to a probability, with the elements in the vector necessarily summing to 1, such that $\text{Softmax}(z_i)$ is the probability that the input falls in class i given the model. The class with the highest probability is returned as the prediction class.

Furthermore for optimization, we used Adam by Duchi et al. (2011). As for our loss function, we used binary cross entropy loss (the standard loss function for problems of this nature), which is given by

$$\mathcal{L}_{\text{BCE}}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right)$$

where $\hat{y}_i$ is the predicted probability for a given class (as returned by the softmax), y_i is the actual label for class i (either 0 or 1), and N is the number of classes (dependent on the downstream task) (Zhang and Zhou, 2014).

As for the hyper-parameters, we prioritized consistency between the number of epochs based on the task used. In every task, we made use of the previously mentioned Adam optimizer. For the SNLI task, we used a batch size of 64 across all embedding types, with 20 epochs, and learning rates of between 1×10^{-2} and 1×10^{-3}. This task also used a three-layer neural network, as opposed to the two-layer networks used for the other two tasks. For the SARC task, we used a batch size of 1000 across all embedding types, with 125 epochs, and a learning rate of 1×10^{-3} for each task (the higher number of epochs mitigated the variability in initializing the learning rates). For the sentiment analysis task, we also used a batch size of 1000 across all embedding types, with 100 epochs, and learning rates of between 1×10^{-3} and 1×10^{-4}. Given the amount of data present, the training was performed on a cloud compute instance with an NVIDIA Tesla T4 GPU and on the built-in Metal Performance Shaders GPU on the M2 MacBook Pro.

5.4 Results

5.4.1 LLM

The train/val/test BLEU and ROUGE scores for each model and the models with no finetuning are reported in Table 1

Model	Train BLEU	Train ROUGE	Val BLEU	Val ROUGE	Test BLEU	Test ROUGE
Presupposition: Finetuned	0.8762	0.9011/0.8416	0.8914	0.9095/0.8559	0.8886	0.9109/0.8560
Presupposition: No-Finetuning	N/A	N/A	N/A	N/A	0.2177	0.4697/0.3246
Implicature: Finetuned	0.6153	0.7891/0.6131	0.6007	0.7863/0.5996	0.6043	0.7811/0.5955
Implicature: No-Finetuning	N/A	N/A	N/A	N/A	0.1135	0.4528/0.1884

Table 1: BLEU and ROUGE (ROUGE1 / ROUGE2) of LLMs with finetuning on IMPPRES and no finetuning

5.4.2 Downstream Tasks

Presented below are the classification accuracies on the test sets for each of the downstream tasks, rounded to the nearest integer.

SENTIMENT-ANALYSIS:

	no comb.	concatenation	SVD	addition
Baseline	81%			
w. Presup		82%	61%	86%
Presup + Implicatures		84%	59%	88%

SARC:

	no comb.	concatenation	SVD	addition
Baseline	68%			
w. Presup		67%	35%	72%
Presup + Implicatures		66%	33%	73%

SNLI:

	no comb.	concatenation	SVD	addition
Baseline	63%			
w. Presup		60%	32%	66%
Presup + Implicatures		63%	33%	67%

Across the board, we saw the best performance for the addition mechanism, as the results seemed to be significantly higher than the baselines. On the other hand, concatenation did not seem to significantly

improve the classification accuracy. We argue that the reason for this is a significant portion of the presuppositions and implicatures share words with the original sentences (explained later); thus, concatenation can cause the model to focus on redundant information. On the other hand, using the addition and normalization technique, the model is able to filter out redundant information while encapsulating novel information. As there is lower dimensionality here as well, the novel information added via the presupposition and implicature embeddings is more likely to play a larger role in the learning of the model. In the future, we would try to learn weights for adding in the encodings for implicatures and presuppositions using methods such as contrastive learning, as addition seemed the most promising.

However, the singular value decomposition method seemed to perform quite poorly on all tasks. In particular, on SARC, SVD performs worse than random guessing, which is quite concerning. We hypothesize that because there is an inherent dimensionality reduction here, SVD focuses on the strongest features and loses out on some nuance. Indeed, on the sentiment analysis task, the one with the least requirements for understanding nuance, SVD performed relatively the best. For SNLI, SVD was no better than guessing. SARC, arguably the hardest task due to the inherent difficulty in detecting sarcasm from online comments (and the one that humans themselves often struggle with), led to the poorest performance, at significantly worse than random guessing. This is because taking the comments at face value (which reducing down the meaning to the most significant element is equivalent to) would lead to a poor understanding of when sarcasm was being used. The whole purpose of our project is to allow the model to incorporate extra linguistic meaning. SVD seemingly abstracts away from this, by selecting only for the most prominent feature.

Another trend is that the inclusion of implicatures seemed to provide a slight performance enhancement beyond that granted by the addition of presuppositions alone. This makes sense because implicatures often bring more contextual information to the meaning of the sentence, compared to presuppositions.

6 Analysis

The generation of implicatures as a whole performed worse than the generation of presuppositions. Many sentences led to generation of valid presuppositions, but either repetitive or invalid implicatures, as it is generally harder to generate implicatures than presuppositions. Recall that the presuppositions and implicatures share many overlapping words with the original sentences. For example, in the sentiment analysis dataset, one of the training examples is "an utterly compelling 'who wrote it' in which the reputation of the most famous author who ever lived comes into question." The corresponding presupposition is "Exactly one person wrote it." While this is a true presupposition, the last two words are overlapping. When combining the embeddings, we seek to focus on the "exactly one" part of the sentence, as opposed to the writing part, which is why adding proved more effective than simple concatenation.

Examples of where the LLM does not work as intended arise in the SARC dataset, as many of the Reddit comments are too short or have a hidden meaning that the LLM could not learn from the IMPPRES dataset finetuning. For example, one of the (more clearly sarcastic) test set comments was "Religion must have the answer." The generated presupposition and the generated implicature both are "Relgion must have the answer" (the same as the original sentence), and all the models misclassify the comment as not sarcastic. The reason for this is that the presupposition and implicature do not add any meaning, but this meaning is necessary to detect the sarcasm in the sentence.

An example of where the model succeeds in detecting the sarcasm in the sentence can be found on the sarcastic example "Wow you read my mind." The model misclassifies this example on the baseline and the model with the baseline and presupposition (which is "Wow you read my mind"). Once you add in the generated implicature, which is "Wow you didn't read my mind," the concatenation and addition models are able to detect the implicit sarcasm as the contradiction is noted here by our model.

For some short comments, the model either returns the comment itself as the implicature and/or the presupposition, or returns the empty string if it cannot come up with anything. In particular, this happens when prompted with a question. The question "Do you want to grab lunch tomorrow?" evidently has the presupposition that tomorrow will come, and the implicature that meals are consumed. However, both the presupposition and the implicature returned are the empty strings. This likely is

due to the fact that the IMPPRES dataset does not contain questions. Further finetuning on a dataset with questions in it would benefit our model, but this prevented us from implementing a downstream question answering task, as our approach would likely not add anything meaningful to the task.

7 Conclusion

In our work, we experimented with incorporating extra linguistic meaning to address the issue of models failing to understand the nuanced meaning of sentences building off the work of Jeretic et al. (2020). Our model makes use of generation by finetuned large language models and then the combination of embeddings to pass into a simple multilayer perceptron for evaluation on downstream classification tasks. Our modest gains across most of the models highlight the challenges in accurately incorporating the extra linguistic meaning derived from these constructs, as generation of presuppositions and implicatures is difficult. However, our success cases show that when correctly generated, this extra linguistic meaning enhances model performance.

In our experiments, we varied between including presuppositions and including both presuppositions and implicatures in our framework to generate encodings via BERT. We also varied the combination technique used, between concatenation, addition, and the novel singular value decomposition approach. These were then passed into downstream tasks, namely Sarcasm Detection, Spoken Natural Language Inference, and Sentiment Analysis. The model that performed the best across each of the tasks was the three-layer model that incorporated both presuppositions and implicatures into the framework and used the additive normalization technique to combine the BERT-produced encodings. This model achieved 67% accuracy on the language inference task, 73% accuracy on the sarcasm detection task, and 88% accuracy on the sentiment analysis task.

We found that on examples where the generated presuppositions and implicatures matched human intuition, the model saw an uptick in performance, implying that this added linguistic meaning is difficult to generate but can serve as an effective technique to improve model performance. Future work would involve learning weights for the additive mechanism used in combining the embeddings, via evaluation on another related task. A more robust dataset containing more complex implicatures and presuppositions that are not necessarily triggered by a specific lexeme would benefit the overall framework, as would contrastive learning against examples that are neither implicatures nor presuppositions.

8 Ethics Statement

Given that we're using subtle and nuanced linguistic features, our model could potentially exacerbate biases already found in natural language (and in particular those that are in the dataset). Given every group of English speakers use text differently, we may be underrepresenting some groups in the training set (particularly groups that are from low-resource, low-income areas). We may be unintentionally propagating biases forward in our model just based on the corpora of text that we train on, which will exacerbate existing social inequities. To fix this, we could incorporate adversarial training into our model. We could also try to train on a more diverse dataset, or attempt to rebalance the dataset if a more diverse dataset is found. Another concern is the interpretablity of our model– oftentimes machine encodings can be difficult to understand, and the overall process can seem quite shrouded to the average consumer. If a user finds something done incorrectly downstream, it may be difficult to work backwards since the model is not highly interpretable. To work around this, we could include good documentation along the way, as well as working on a mechanism that explains each layers to users and demonstrates what the model has learned up to that point.

9 Contributions

Riley worked on finetuning the LLMs for use in generation of the presuppositions and implicatures. Ishaan worked on the sarcasm detection task, while Bradley worked on the SNLI task. Ishaan and Bradley worked together on the sentiment analysis task. Dataset preprocessing, generation of the implicatures and presuppositions, production of encodings via BERT, and combination of the encodings was shared equally between Ishaan, Riley, and Bradley. Writeup and analysis was shared between team members, with each team member contributing to the sections they implemented.

References

2017. A large self-annotated corpus for sarcasm.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-bert: Improving pre-trained transformers with syntax trees.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models imppressive? learning implicature and presupposition.

Stephen C Levinson. 2000. *Presumptive meanings: The theory of generalized conversational implicature*. MIT press.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Barbara Partee et al. 1995. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360.

Christopher Potts. 2015. Presupposition and implicature. *The handbook of contemporary semantic theory*, pages 168–202.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837.