BRAINTEASER: A Novel Task Defying Common Sense Sentence Puzzle subtrack

Yash Yadav 2021117 | Harshvardhan Singh 2021052 | Pulkit Nargotra 2021273 | Shantanu Prakash 2021285

April 25, 2024

1 Introduction

In natural language processing (NLP), the development of language models has created an interest in solving human cognition and reasoning problem. Traditional NLP tasks focus on processes, which adhere to logical, sequential, and rule-based reasoning, however, lateral thinking has been relatively overlooked in NLP.

Inspired by the CODALAB competition, we worked on the BrainTeaser Question Answering (QA) task which unlike conventional QA tasks engage in lateral thinking and defy default commonsense associations. We wanted to push the boundaries and think beyond conventional wisdom and explore unconventional solutions to complex problems.

Question-Answering task is driven by a desire to bridge the gap between human-like language understanding and machine capabilities. It is a complex problem and requires models to decipher context, infer meaning and generate precise responses.

In summary, the implementation of the BrainTeaser project presents an exciting opportunity to explore the intersection of NLP and cognitive psychology. By encouraging models to engage in lateral thinking and challenge default assumptions, BrainTeaser aims to foster the development of more human-like and versatile NLP systems capable of navigating the complexities of real-world language understanding tasks.

2 Related Work

Natural Language Processing (NLP) has been revolutionised with the coming of transformer models like BERT (Devlin et al., 2019 [BMR+19]) and ChatGPT (Brown et al., 2020 [DCLT19]) including question-answering. Their architecture captures semantic and contextual nuances (Zhang et al., 2020 [ZWZ+20]), proving exceptionally effective in comprehending and responding to complex inquiries. By training on extensive text corpora, they develop a deep understanding, enabling them to identify the most plausible answers among multiple choices.

From the works we found, we could see how question answering and natural language processing interacts and intersects (Hirschman and Gaizauskas, 2001 [HG01]) and a comparison on approaches based on natural language processing, information retrieval and question templates, analyzing the differences among the QA approaches, their accuracy and applicability (Andrenucci and Sneiders, 2005 [AS05], Prager, 2007 [Pea07]).

We found a paper (Abdelhak Kelious and Mounir Okirim, 2024 [KO24]) that generated overall score of 0.98 in the BrainTeaser competition. It paired the questions with the given options and passed them to the transformer model and provided the output to the softmax function, thus getting an answer among the options. It excels particularly in thinking challenges where the puzzle, often contrary to common sense, is based on sentence excerpts.

Another paper (Ioannis Panagiotopoulos et al., 2024 [PFLS24]) that worked on the sample problem and did leverage transfer learning techniques starting from smaller models based on masked language modelling, such as BERT (Devlin et al., 2019 [BMR+19]) and consequent BERT-based encoders, proceeding with similar techniques on LLMs for parameter updating to querying the model's prior knowledge via prompting.

3 Dataset

The BrainTeaser task at SemEval-2024 (Jiang et al., 2023b, 2024b) is a multiple-choice questions answer dataset. We used the SP dataset for our models. Each instance of the dataset contains a question with four options. Out of the four options, one is correct, two distractors(incorrect) and one option which is 'None of the above'. Also, each question has three variants in total - Original, Semantic Reconstruction and Context Reconstruction. The dataset contains 169 unique questions with each of them having the three variants. Thus totalling to 169*3 = 507 entries. Of these 507 entries, we made the split as follows: Train: 405 entries, Validation: 51 entries,Test: 51 entries. These splits were created so that no variant of a question goes into any other split.

4 Methodology

4.1 Textual Entailment

We have made a model based on transformers for multiple-choice questions. We took two sentences where sentence1 is the question and sentence2 is the option for that question and we do this for each question-answer pair, thus creating 4 independent entries for each question. The pairs are passed to the T5 small model which outputs 'entailment' and 'option' on predicted correct and incorrect option respectively.

Using the above trained model, we predicted the label of each sentence pair entry of the test data and passed it to a softmax function which gives us the predicted correct option out of the four options and assigned 'entailment' to it and 'other' to the rest.

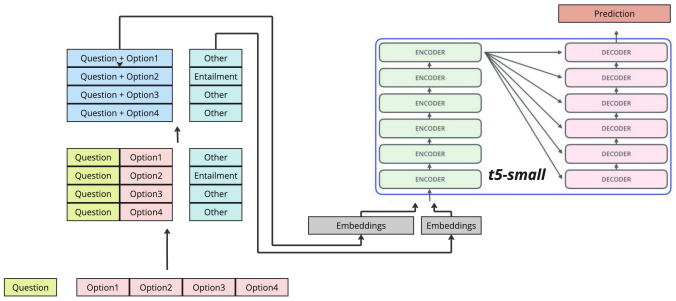

Figure 1: Architecture for Textual Entailment

4.2 Base State-of-art Models

For this BRAINTEASER QA Task, we selected T5(Text to Text Transformer) as our base model since T5 is a model designed as a unified framework to formulate all the text based problems into a text-to-text format. We leverage the broad understanding of language of the model obtained from its pretraining on a mixture of supervised and unsupervised learning tasks. Hence, its understanding of human language turns out to be useful for tasks which involve innovative thinking and problem solving.

There are other state of the art models such as GPT2, BERT, BART, SpanBERT, RoBERTA which could have been considered for the task. GPT is a decoder model which is capable of understanding and manipulating instructions and more suitable for generation tasks based on instructions rather than interpretation tasks. It can interpret obviously upto some extent but is not considered effective. [RS18] RoBERTa is an optimized version of the BERT model. Both BERT and RoBERTa specializes in classical tasks like question answering, sentiment analysis, language entailment tasks but not much effective for creative reasoning Q/A tasks. [DT19]

We use a variant of the original T5 model which is 'Flan-T5' particularly adapted for tasks that require understanding and generating instructions. It is a strong candidate for complex nuanced language tasks. It can deal with tasks involving instructions and reasoning. We required a task to creatively

answer the questions in addition to understand the text which is fulfilled by flan-t5 model[RO22]. Here T5-small was suitable for our task since the dataset we had was very small. It would have led to overfitting if we had considered more complicated architectures like T5-base or T5-large. We used T5EncoderModel for our task instead of T5Model and T5ModelForCon- ditionalGeneration since T5EncoderModel architecture was the most suitable for our task.

4.3 Mutliple Choice Answering

4.3.1 Flan-T5 model

Initially, we took a very basic architecture approach where we used the basic T5 encoder model variant: 'google/flan-t5-small' and on top of it a linear classifier layer which takes in input the output of the t5 encoder model, i.e, the last hidden states which are then passed on to the classifier layer to give output as logits. For each question, we prepare four encodings i.e question concatenated with each of the four choices separated by a special token '[SEP]' and each of these 4 embeddings is supplied separately to the model architecture and the logits obtained from the four of these are then passed to a softmax layer to predict the correct answer by the model.

We deploy a linear layer to the encoded outputs obtained from flan-T5 model to reduce the high dimensionality of the outputs and obtained logits which were simply passed on to the softmax layer. The Softmax layer receives logits from all the four question-option embeddings and outputs softmax probabilities and using argmax we obtain the answer as the one with highest softmax probability. We encode the labels as a one-hot encoded vector of length 4 here and compute the accuracy and F1 score respectively.

4.3.2 Modified Flan-T5 model

In previous architecture, the role of the linear layer was primarily to focus and capture the features most relevant to each of the embeddings which help us in distinguishing between different question-answer pairs and supplying all the four question answer pairs helps us in comparative reasoning between the encodings. Thus, by directly comparing the different embeddings, the model learns to predict the answer with a creative reasoning i.e thinking outside the box, not typically following the logical reasoning or the textual entailment task here.

But we observed that for the previous model architecture the model was slightly prone to overfitting given the very high accuracy we observed from starting epochs itself so we made modifications to the previous architecture.

We passed the outputs of the classifier layer to GELU (Gaussian Error Linear Unit) activation function. It was done to introduce non-Linearity to the outputs which was very essential for learning the complex and creative patterns of the question-answer pairs. It helped in better learning and interpretation of the question-answer pairs which helped in preventing overfitting and capturing the nuances of the data better. [4] Then, we used the Dropout technique which is a very popular technique to prevent overfitting of neural networks. It helps in generalizing to better unseen data. To further increase the model's capacity to capture the features further in data and to capture the possible new features that can be captured after non-linearity introduced through GELU activation, we pass the encodings through a final fully connected layer i.e Linear Layer to enhance the embeddings and features captured further.

Therefore the addition of the three layers helped in processing more complex relationships in the embeddings, preventing overfitting significantly and mapping all the high level features to predictions.

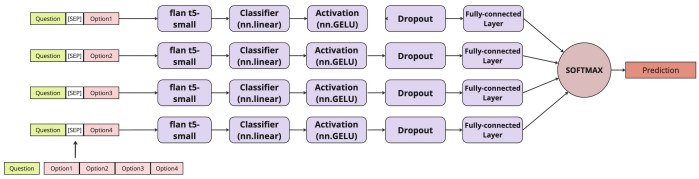

Figure 2: Final architecture for Multiple Choice

5 Results

Model	Accuracy
flan t5-small (Multiple Choice Answering)	0.7745
t5-small (Multiple Choice Answering)	0.7450
t5-small (Textual Entailment)	0.6862

In our project assessment, we examined the performance of three models across distinct tasks. The accuracies, presented in descending order, showcase the effectiveness of each model. Flan T5-Small demonstrated the highest accuracy of 0.7745 in the Multiple Choice Answering task, highlighting its robustness in handling such tasks. Following closely, T5-Small achieved an accuracy of 0.7450 in the same Multiple Choice Answering task, indicating its competitive performance. Lastly, in the Textual Entailment task, T5-Small exhibited an accuracy of 0.6862, showcasing its capability in this specific domain. These accuracy metrics provide valuable insights into the strengths and capabilities of each model, aiding in informed decision-making for future tasks and projects.

6 Observations

Our predictive accuracy for questions featuring "None of the above" as an answer option was notably lower, indicating a challenge in accurately forecasting outcomes for these specific question types.

On increasing the batch size to 8 in our flan-t5 model, we saw a decrease in the accuracy of the model on the validation set.

7 Conclusion

We observed that T5 Models and its variants performed signifcantly good in predicting answers with creative reasoning. We experimented with multiple activation layers and the results with GELU surpassed all the other activation functions.

We also observed that 'None of the Above' option was predicted better by the textual entailment task than the multiple choice answering task. If on multiple choice answering task, we had used a base T5 model which was trained on fact awareness dataset we could have significantly improved the accuracy of the models in multiple choice answering task.

We also observed that alternatively we could have use the pretrained model of T5 on finetuning on certain tasks which are similar to ours like analogy-methaphors task, Lateral puzzle thinking, creative writing, story completion, what if reasoning task and this would have given significantly better results if used as this base model.

8 Github

The github repo contains the dataset, the code and the model checkpoints.

References

[4] Museum 7432 semeval brainteaser task submission.

[AS05] A. Andrenucci and E. Sneiders. Automated question answering: Review of the main approaches. *IEEE*, 1:275–300, 2005.

[BMR+19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. A simple method for commonsense reasoning. 2019.

[DT19] Ming-Wei Chang Kenton Lee Devlin, Jacob and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[HG01] L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. *Nat. Lang. Eng.*, 7(4):275–300, 2001.

[KO24] Abdelhak Kelious and Mounir Okirim. Abdelhak at semeval-2024 task 9 : Decoding brain-teasers, the efficacy of dedicated models versus chatgpt. 2024.

[Pea07] J. Prager and et al. Open-domain question-answering. *Foundations Trends Inf. Retrieval*, 1(2):91–231, 2007.

[PFLS24] Ioannis Panagiotopoulos, Giorgos Filandrianos, Maria Lymperaiou, and Giorgos Stamou. Ails-ntua at semeval-2024 task 9: Cracking brain teasers: Transformer models for lateral thinking puzzles. 2024.

[RO22] Jordan Menick Aäron van den Oord Oriol Vinyals Rae, Jack W. and Simon Osindero. Scaling instruction-finetuned language models. 2022.

[RS18] Karthik Narasimhan Tim Salimans Radford, Alec and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[ZWZ+20] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. Semantics-aware bert for language understanding. *AAAI*, 2020.