CS-542 : Autocast Competition Midterm Report

Smriti Suresh (U83742024) , Jaisal Singh (U60118211)

1. Introduction

Forecasts of climate, geopolitical conflict, pandemics and economic indicators help shape policy and decision making. This report aims to give insights into our approach for competing in the Autocast Competition 2023 and providing a descriptive analysis of the large language models and methods that were successful as well as those that could be improved. We implement training the machine learning models on the Autocast dataset and display our findings and improvements to the Future World Events Prediction task.

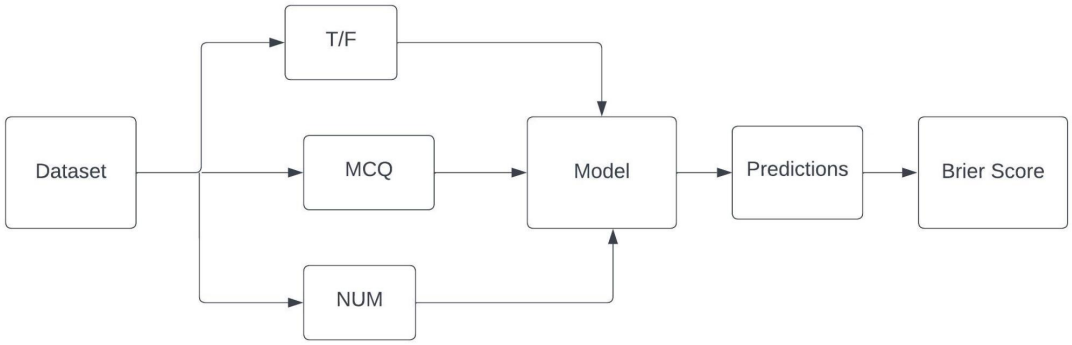
2. Data

The competition provided the Autocast Questions dataset in JSON format, that contains 4161 entries in the format: question: { 'id', 'question', 'background', 'qtype', 'status', 'choices', 'answer', 'crowd', 'publish_time', 'close_time', 'prediction_count', 'forecaster_count', 'tags', 'source_links'}

The dataset contains True and False Questions, Multiple Choice Questions and Numerical Questions which are identified by the *qtype* attribute. We have made use of the '*question'*, the '*qtype'*, the '*background'*, '*choices'*, '*tags'* as well as '*close_time'* for this work.

We preprocess the dataset by splitting into train and test sets, removing questions with no answers and truncating the training set to entries with close_time = 2021-09-30 as specified in the rules of the competition.

3. Architecture



4. Models

4.1 Random Baseline

The first baseline we tried out was a random baseline model which predicted a random outcome for each of the question types:

1) For True/False, it outputs numbers 0 or 1 randomly.

2) For Multiple choice, it outputs a random number between 1 and the number of the choices provided in the question.
3) For Numerical problems, it outputs a random number through Python's random() function.

4.2 Calibrated Random Baseline

This random baseline is an improvement over the random baseline as it performs hyperparameter tuning of the constants like epsilon which is a small fractional value that is added or subtracted from the label probabilities so there is a clear majority in the choices and ties can be avoided. We experimented with different epsilon values for the 3 different types of questions.

4.3 GPT-3.5

The GPT-3.5 API (Application Programming Interface) is a service provided by OpenAI that allows developers to integrate GPT-3's language generation capabilities into their own applications.

The GPT-3 API also includes various options and parameters that can be used to customize the generated output, such as setting the length of the generated text or controlling the level of creativity or coherence in the output. Additionally, the API provides various safety features, such as content filtering and bias detection, to help ensure that the generated language is appropriate and unbiased. It is trained on a diverse range of internet text, including websites, articles, books, and other text documents. We observe a huge difference in the training score v/s testing score for GPT-3.5 which suggests that the model may have overfit the given training set and needs further corrections.

4.4 BART

BART is a transformer encoder-decoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. BART is pre-trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text. BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering). This model was trained on the MultiNLI (MNLI) dataset in the manner originally described in Yin et al. 2019. It produces the best training accuracy score of them all (**84.21**) but does not maintain the score on the test set.

4.5 T5 (Text-to-Text Transfer Transformer)

T5 is a powerful transformer-based language model developed by Google's AI research team. It is based on the Transformer architecture introduced by Vaswani et al. in the paper "Attention Is All You Need."

T5 is a neural network that is trained on a wide range of natural language processing tasks, such as language translation, summarization, question-answering, and text classification. Unlike previous transformer models that were trained for specific tasks, T5 is trained in a "text-to-text" manner, meaning it is trained on pairs of input/output sequences that are related to each other in some way. T5 also performs almost as well as BART with some deficiencies in the T/F task.

5. Results

The following results consider the Brier's Score as the measure for evaluation that is within the competition guidelines and we obtain results as follows: (Best scores are in Bold)

Model	T/F	MCQ	Numerical	Combined Test Score
Random Baseline	32.29	43.52	33.32	109.13
Calibrated RB	**24.99**	**39.13**	**22.62**	**86.74**
GPT-3.5	25.00	43.13	**22.62**	90.75
BART	25.00	**39.13**	22.65	86.78
T5	25.02	**39.13**	22.65	86.79

5.1 Github Link

Our GitHub repository

To run the code, open the "autocast_submission.ipynb" file and run the cells as per the model. The submission.zip file will be created for evaluation. All the data used was taken from the auxiliary data provided in the CS542 drive. You will need "autocast_questions.json" as well as "autocast_competition_test_set.json" to replicate our results, and the "autocast_test_set_w_answers.csv" for evaluation.

6. Conclusion

We have found that training for the numerical type questions seem to be the trickiest as the models we have utilized above are more suitable for text classification and labeling tasks. The Calibrated Random Baseline gave good results but these scores are not always reliable. GPT-3.5 has proven itself at the Numerical task but seems to fail at predicting true/false or MCQ type questions with a satisfactory score, as well as the additional overfitting issue. The BART transformer seems to perform the best out of them all. The T5 model performs well by creating embeddings but there is scope for improvement. We plan on extending these findings by fine tuning RoBERTa, GPT-4 and other models on the larger dataset and performing inference.

7. Future Scope

Creating an ensemble of the above models to predict the outputs may work better.
Attempting few-shot learning on the most similar questions in the training set and passing them through the model along with the question and context could help in classifying similar questions in similar ways.