# Precog Recruitment Task
## NLP

Name: Abraham Paul Elenjical
Roll No.: 2022114007

**I did Task a, Task b and the Bonus Task.**

## Task a

### Part i

For the first part, (with constraints) I used the unsupervised training function from fasttext on the Brown Corpus to calculate the word embeddings, along with WordNet, resulting in an accuracy of approximately 61%.

### Part ii

Upon removal of constrains, I imported already existing word embeddings from the spacy library and used them along with WordNet, which improved the accuracy to around 65%.

## Task b

### Part i

For phrase similarity, I used the same word embeddings from the spacy library (en_use_lg), to compute each token's word embeddings in the phrase, then averaged them and computed cosine similarity between the two phrases. Using this method I was getting an accuracy of around 51%, which I expected as I read the paper which presented this PiC dataset, which stated that non-contextualized phrase embeddings generally result in an accuracy of around 50%.
I tried using similarity of POS tags to improve accuracy, but upon inspection of the dataset, I observed that this did not result an improvement.

### Part ii

For sentence similarity, I used the word embeddings from the spacy library, to get an accuracy of 48%. I tried running a kernel function to calculate similarity between the dependency trees of the sentences, but it did not work.

# Bonus Task

## Part i

For Part I, I did not fine-tune my own model, but rather used the already finetuned all-MiniLM-L6-v2 model which is an SBERT Transformer model from the SentenceTransformers HuggingFace python library. It's results were underwhelming as it gave me an accuracy of only around 50%.

## Part ii

I prompted the Gemini (Bard) API for results(on only around 5% of the dataset due to computational restrictions), and it returned results with an accuracy of around 45%, but also kept returning an error stating that my prompts were resulting in a safety violation.

I tried prompting the Llama 2 model(with 4-bit quantization) too, but it was returning completely wrong results.

## Part iii

Across the three settings, the static word embeddings gave me the highest accuracy. I think this can be attributed to the fact that I did not fine-tune my own transformer model for the task, but rather used an already existing HuggingFace model, as the general trend shows that Transformer based models significantly outperform methods such as static word embeddings in Semantic Similarity Tasks.
LLMs similarly did not give great results, as it often misunderstood my prompt and returned the wrong results.

# Scope for Improvement:

- Incorporating other evaluation metrics other than accuracy
- Implementing contextualised embeddings using a fine-tuned BERT model.
- Appropriately using prompt engineering to get better results out of the Commercial and Open-Source LLMs.