# LLMs Hallucinations

Yoav Kor, Yuval Assif, Yuval Reuveni, Ohad Carmel

Advisors: Dr. Yuval Pinter, Mr. Nitay Calderon

# Background

- These days many people rely on language models for various tasks.
- However, many language models provide incorrect answers with full confidence and without any disclaimers.
- Currently, there is no method that allow the user to estimate the reliability of the answer that the model provides.

# Background - Terminology

**Hallucinations** in LLMs occur when models generate responses that are factually incorrect or inaccurate.

**Hedging** is the term for when LLMs express doubt using phrases like “appears to be…” and “probably”

# Background

- In our project, we focused on a vision language model (Llava) which takes an image as an input and outputs a description of it.

Let's see some examples…


The image features a man standing on a roof wearing a high-visibility vest.
He is holding his hands up, possibly preparing to take a photograph.

The image features a man and a woman sitting on a stone wall overlooking the ocean. They both have sunglasses on, and the woman is smoking a cigarette.

The image features a young boy sitting on the bottom step of a staircase, with a toy in his hand.
The toy appears to be a robot-like figure, and the boy is holding it with affection

Even when a model uses hedging, it is not always directly correlated to the hallucination of the model.

The image features a young boy sitting on the bottom step of a staircase, with a toy in his hand. The toy **appears to be** a robot-like figure, and the boy is holding it with affection

# Project Efforts

- Experiment
- Detection Model

# The Experiment

- Goal: Test how hallucinations and hedging impact the model's trustworthiness in continuous interaction with humans.

# Trust in AI: Visual Descriptions and Human Judgment

Welcome to the Trust in Language Models Experiment. In this study, you will be presented with descriptions of images generated by a language model. For each description, you will see two statements about the image. Your task is to determine whether each statement is true or false based on the provided description and how much you trust the language model. Additionally, you will be asked to bet points on your level of trust in the model's description. There will be ten different descriptions for you to evaluate. Thank you for participating!

Display Description Word by Word

Display Hallucinations Marked

Display Low Confidence Marked

START

# Data Preparation

- Hundreds model-generated descriptions of images with manually-tagged hallucinations and hedging.
- Creating 4 probes for each image based on the generated description and manually tag it with True/False labels.

# The Experiment Implementation

- **Technologies Used:**
- **React**:
  - Built the dynamic and responsive front-end.
  - Utilized Material-UI for styling.
- **Node.js**:
  - Powered the back-end server.
  - Handled asynchronous operations.
- **Mongoose**:
  - Interacted with MongoDB.
  - Simplified data modeling and validation.

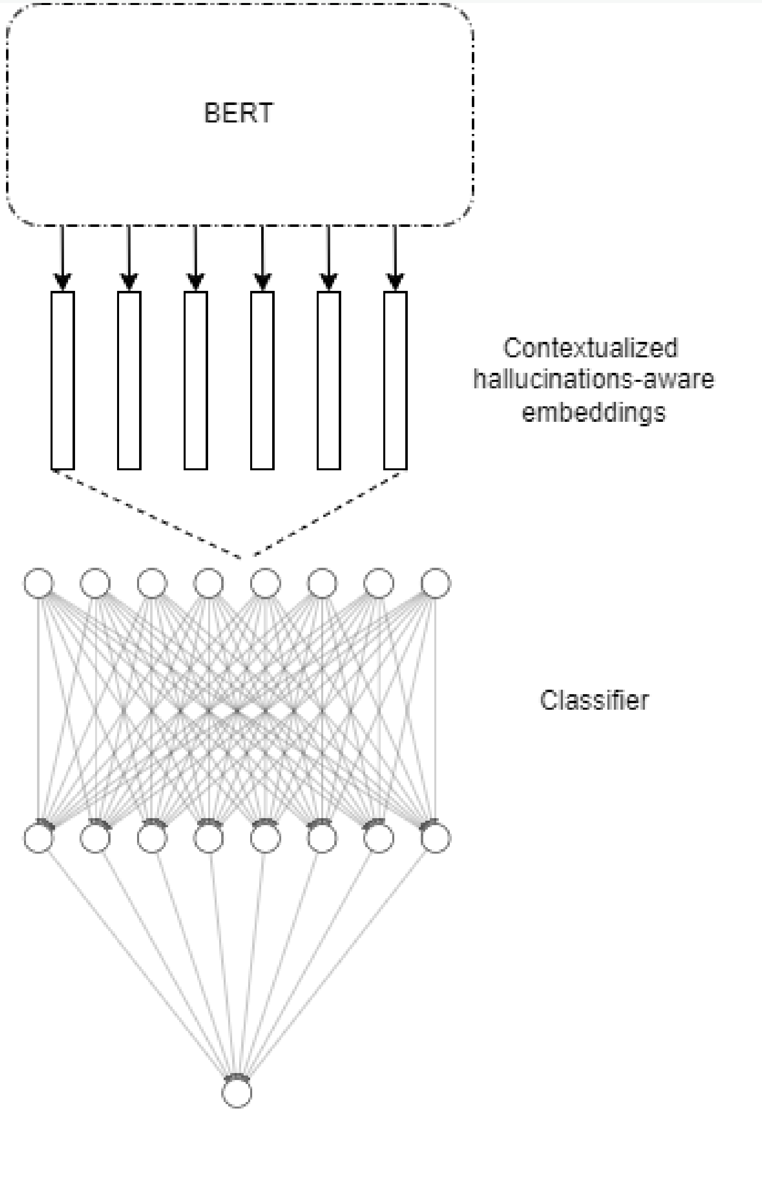
- **Key Features:**
- **Dynamic Flow**:
  - Real-time updates based on user input.
- **Data Management**:
  - Robust storage and retrieval with MongoDB.
- **User Interface**:
  - Intuitive design using React and Material-UI.

# The Detection Model

- Goal: Find a way to tell when the LLM hallucinates.
- We aim to manage to predict for each word (each token actually), if it is a hallucination or not.

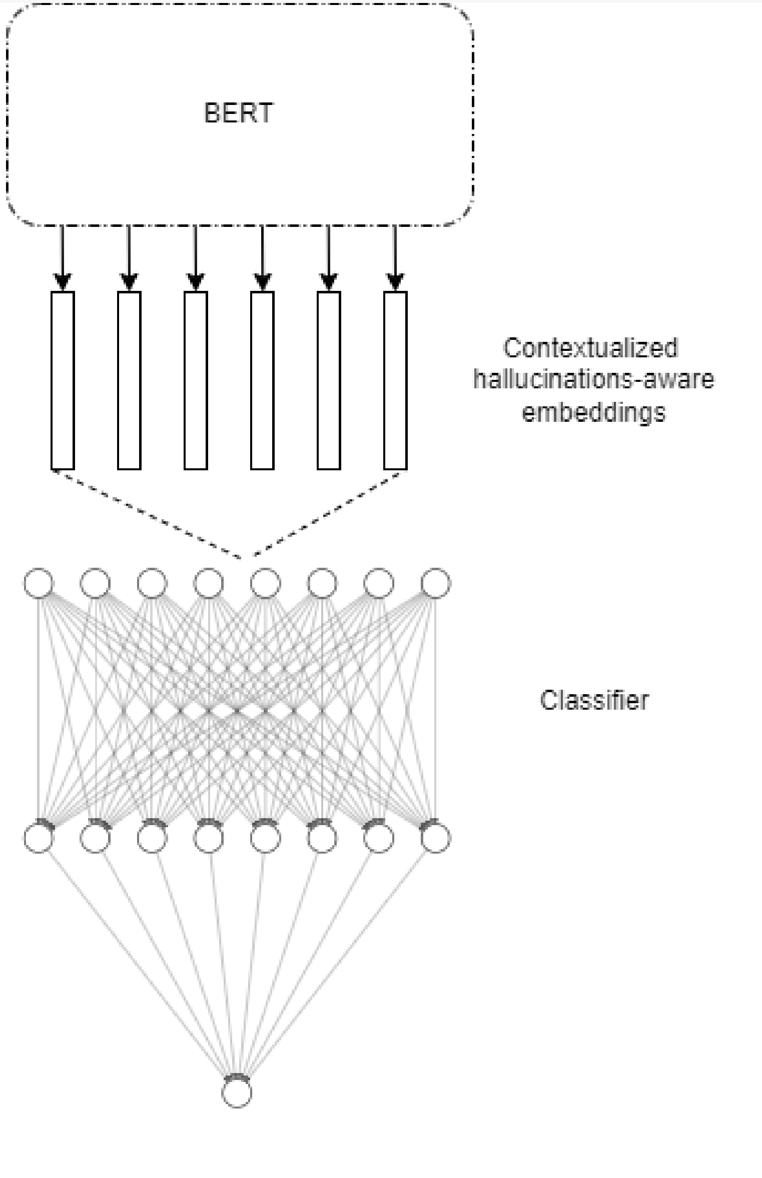
# Model Structure

- To solve this, we propose to fine-tune a pre trained *BERT* model with auxiliary classification feed forward neural network head.
- Classifier *FFN* consists of 3 layers and the last layers is a single neuron represents the token hallucination logit.



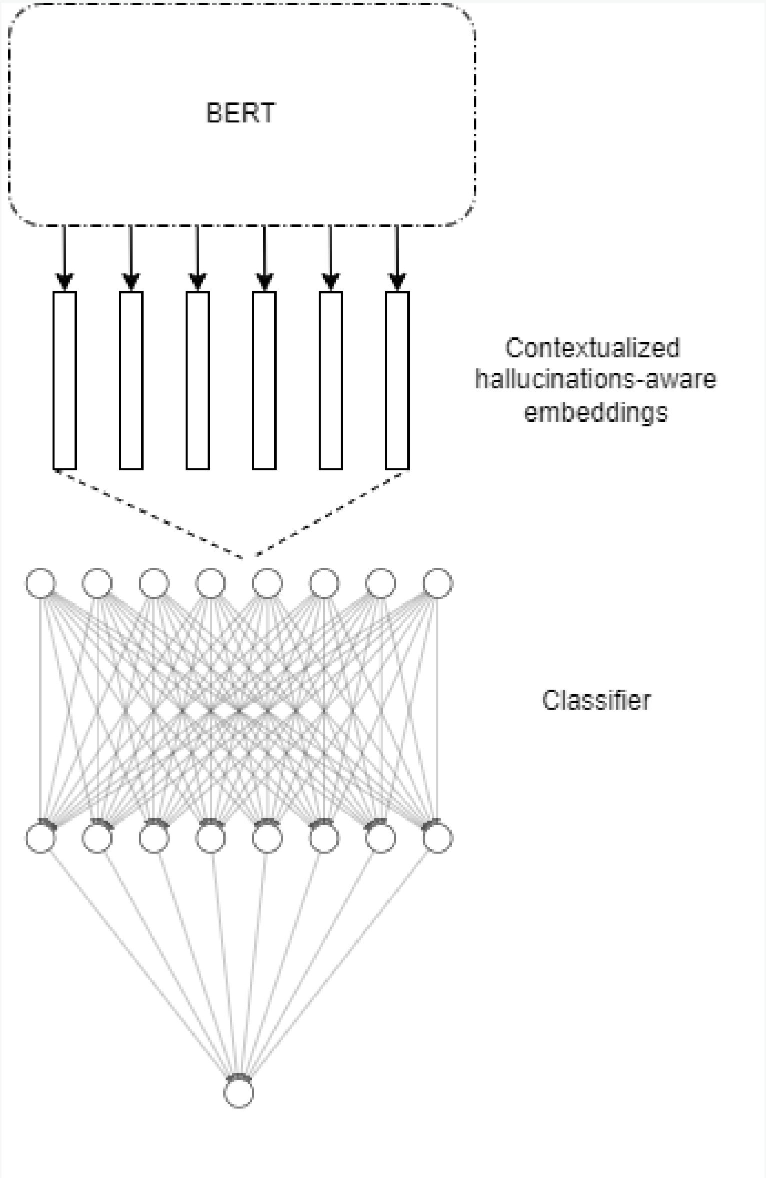
# BERT

- Input: Llava description of the image
- Output: Embeddings vector



# The Classifier

- Input:
  - Embeddings' vectors encoded by BERT
  - The logits (probabilities) that were given to the original word (token) by the LLM.
- Output: Probability that the word is a hallucination

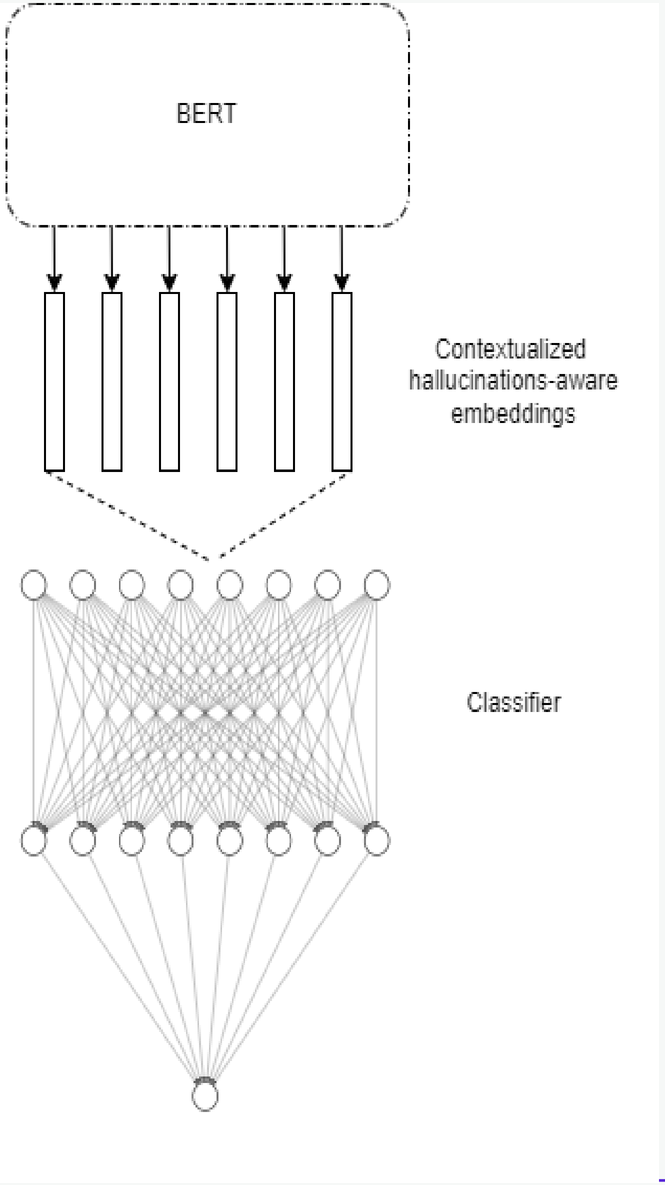


# Model Training

- We trained the model using BCE loss function for each token, i.e. $\frac{1}{N}\sum_{ij} BCE(t_{ij}, \widehat{t_{ij}})$

Where $t_{ij}$ denotes the token i of example j and N is the number of total examples.

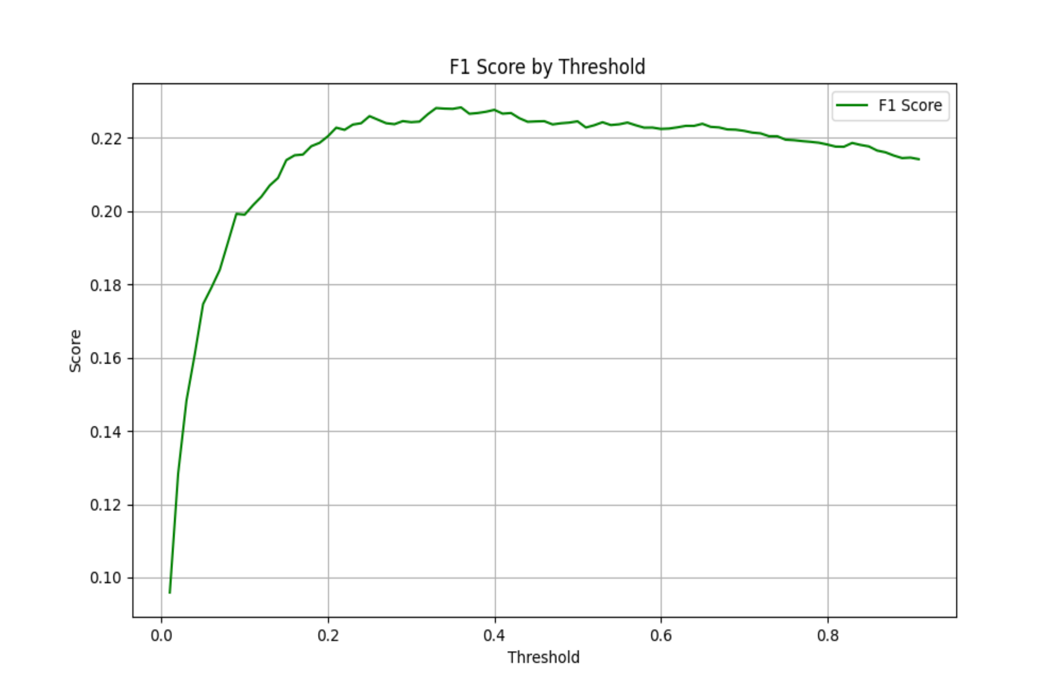
- Learning rate was set to $10^{-5}$ with a total of 3 epochs for training.
  Batch size was set to 1 and optimization was done using AdamW  optimizer.



# Results

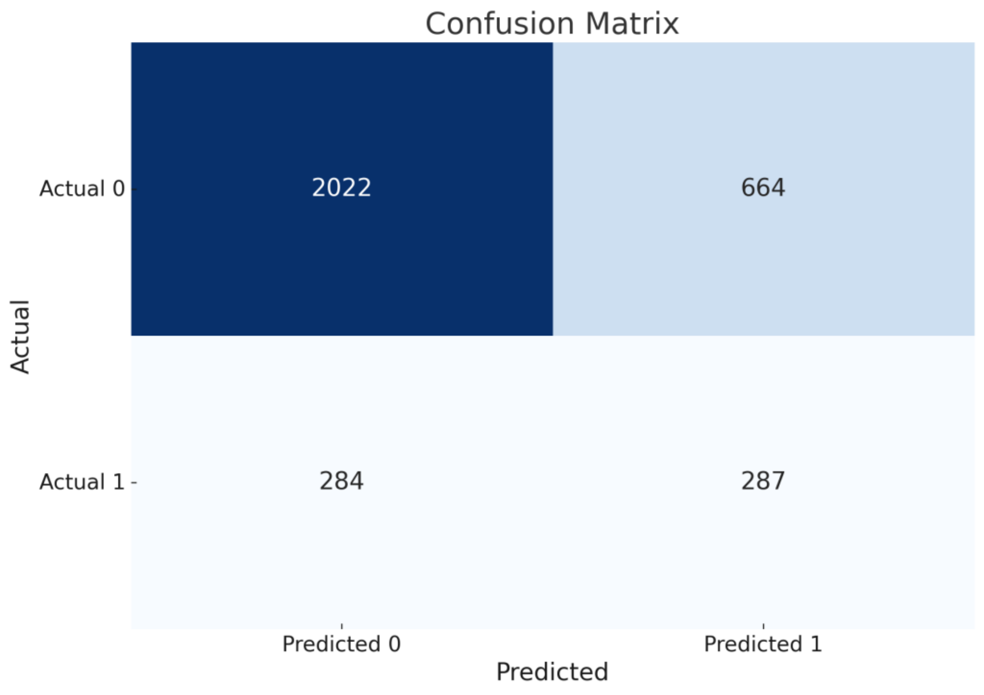
- Experiment – TBD
- For model evaluation we use the F1 metric which provides a desired balance between precision and recall

# Baseline Model




# Ours

F1 score: **~0.59**



Let’s see some examples…

False Negative
False Positive
True Positive
Better than us?

- In the image , two young boys are playing soccer on a field . one boy is wearing a red and white striped jersey , while the other one is wearing a black and purple striped jersey . they both seem to be running towards the soccer ball , which is placed towards the center of the field . there is also a group of three other people watching the game , standing near the edge of the field , possibly as spectators or waiting for their turn to play . overall , the scene represents an exciting moment during a soccer match .

False Negative
False Positive
True Positive
Better than us?

- the image is a large , detailed collage of various lego people . the lego figures are arranged in rows , creating a sense of unity and organization . the people are of different sizes and poses , showcasing the diversity of the lego characters . the collage is a vibrant and engaging display of lego people , capturing the essence of the popular toy.

# Plans for the Future

- Now the predictions are on token level. We plan to create a post-processing heuristic-based mechanism that determines whether a sequence of words (a span) is a hallucination or not.
- Improve the model score:
  - Although we provided a baseline model and an algorithm that beats the baseline in terms of F1 score, we believe that further research might produce even better results.
  - Some improvement ideas:
    - Use more data (using even multiple VL models)
    - Use more Features - using not just logits but also embeddings from inner layers of the model, using autoregressive model for encoding, adding POS at the token level etc.

# Thank you