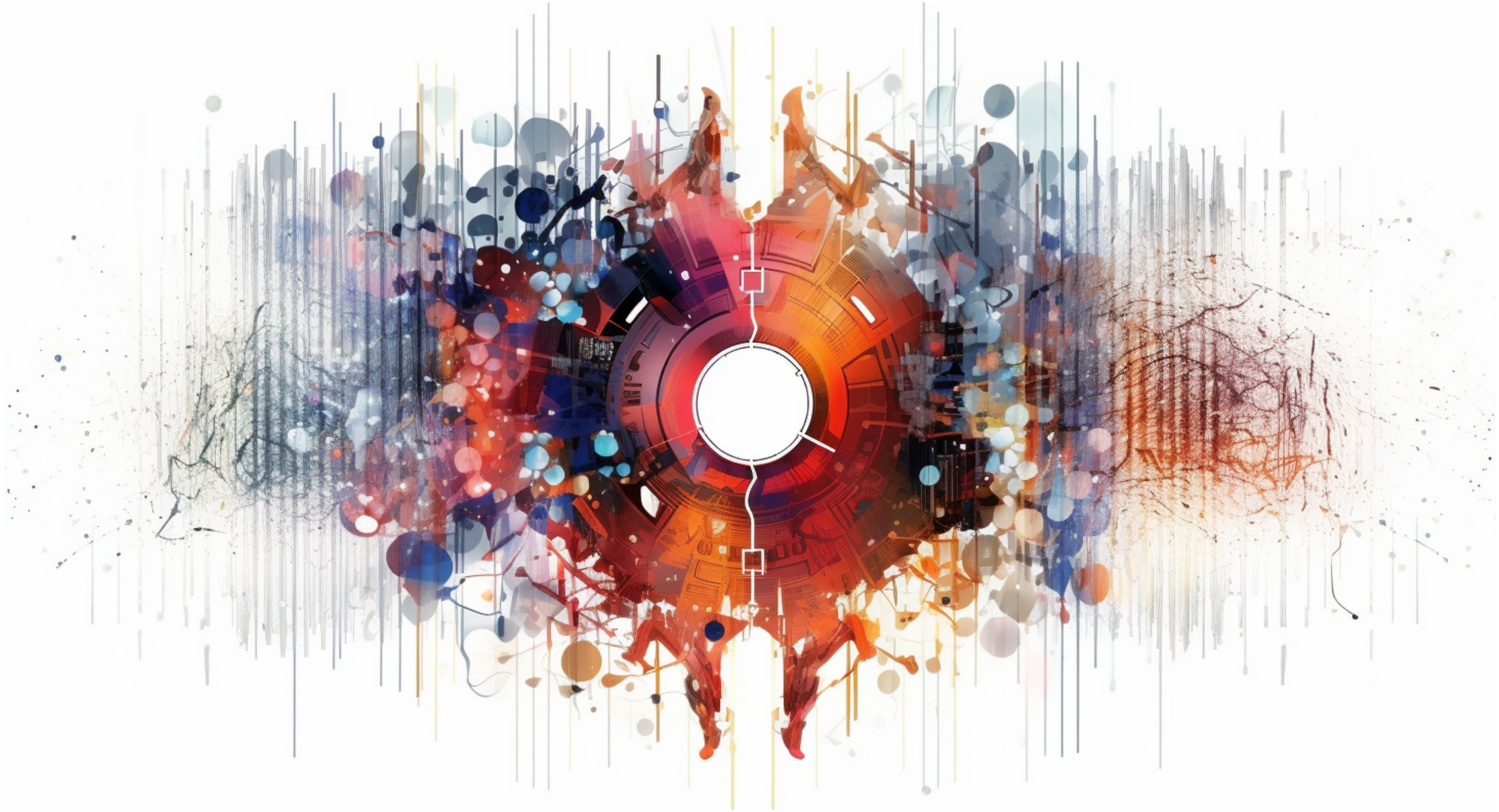
# Patent Analysis Challenge



Pablo Martín Ramiro

31/07/2023

# Agenda

- Problem definition
- Methodology for building a prototype
- Model evaluation
- Discussion
- Fine-tune a model for the patent domain

# Problem Definition

## Patent Analysis

Design a solution based on LLMs to extract information on measurements and their values from patents

## Three approaches

Ordered by complexity:

1. Use prompt-engineering on a general pre-trained LLM model
2. Fine-tune a general pre-trained LLM model on a custom dataset that includes examples of measurements and their values
3. Train an LLM model from scratch on a custom, high-quality dataset that includes data scraped from the Internet, prioritizing examples of measurements and their values

## Today, we will see…

- A simplified prototype based on few-shot prompt-engineering using a general pre-trained LLM model
- The methodology to extend this prototype and fine-tune a general pre-trained LLM model on a custom dataset that includes examples of measurements and their values

# Methodology for Building a Prototype

## Goal

Build a dataset of patents, extract the most relevant information and use prompt engineering to extract measurements and their values

## Challenges

- Data files from USPTO contain hundreds or thousands of patents
- Each data file contains patents granted in a week of a year, belonging to all subjects
- Only some patents contain relevant information for this task
- I am new to NLP and LLMs!

→ Extensive preprocessing is required build a dataset

## Approach

1. Download the data and process it to extract the individual patent files associated to specific subjects
2. Process each patent file to extract the most relevant fields (Title, Abstract, Description and Claims) and save them in JSON files
3. Process each JSON file to create the input text to be analyzed by the LLM model
4. Text segmentation + Prompt-engineering: experiment and improve performance on evaluation data
5. Post-processing of the LLM response
6. Assess performance on test data

# Model Evaluation

## Evaluation Setup

We have a single string that contains the most relevant information of a patent file, ready to be processed by a general pre-trained LLM

- Evaluation and test data: only 2 + 2 patents
- We optimize text segmentation, the prompt and the post-processing on evaluation data

## Prompt-engineering

- This is one of the keys for succeed:

> Instructions: Extract information about measurements and their values from the text. Identify all the objects that are measured. For each of these objects, extract the variables that are measured, the value of the measurements and the units of the measurements.
>
> Text: The table top has a length in the range of 6 cm to 8 cm, and a width of 20 cm to 22 cm. The table is particularly distinguished by having a red surface, and having a triangular form. The table legs have a diameter between 150 mm to 250 mm. The height is in the range of 0.7 m and 0.9 m.
>
> A: Measurement ID 1. Object: Table top. Variable: Length. Value: 6 cm to 8 cm. Units: cm. Measurement ID 2. Object: Table top. Variable: Width. Value: 20 cm to 22 cm. Units: cm. Measurement ID 3. Object: Table legs. Variable: Diameter. Value: 150 mm to 250 mm. Units: mm. Measurement ID 4. Object: Table legs. Variable: Height. Value: 0.7 m to 0.9 m. Units: m.
>
> Text: input_patent_text_goes_here
>
> A:

## Post-processing of the LLM response

- We need to extract the measurements from the LLM response (and discard anything else!) and save them in JSON format

# Demo time!

# Discussion

## Pre-processing

- It is essential and it worked fairly well in general
- Invest in things that last: it can be reused to fine-tune a model

## Text segmentation

- Fixed-length sequences: yields to missing context in the patent analysis by the LLM
- A better approach is needed: NLP to identify and filter individual paragraphs after further pre-processing?

## Prompt-engineering

- In this application, prompts do not generalize well to different patent subjects, text formats and measurements
- The sensitivity is limited to the units provided in the examples of the prompt
- LLMs make things up!

## Post-processing

- It is essential and it worked fairly well in general

This solution could bring value in specific subjects, but it does not have a reliable performance

# Fine-tune GPT for Patent Analysis

## Prepare the Training Data

1. Define the scope: Focus on a collection of measurements of interest. Examples: temperature, weight, length, or volume
2. Collect text data: Focus on relevant data. Examples: patent texts, scientific articles, or technical documents
3. Annotate the data: Identify the measurement entities in the text and annotate the start and end positions of these entities (manual)
    a) BIO encoding: Beginning (B), inside (I) and outside (O) of a measurement. “The temperature is 25 degrees Celsius.” “OBOBIIO”
    b) Handle the variation of formats: "25 degrees Celsius," and "25°C," should be considered as the same measurement entity
4. Tokenize and format the data: Includes tokenizing the text and the applying text segmentation
5. Correct class imbalance and create training, validation, and test sets
6. Prepare for fine-tuning: Convert tokens to input IDs, create attention masks, and convert labels into numerical format

## Fine-tune a Model

1. Train a model to predict a label for each token in the input sequence, indicating whether it is part of a measurement or not (B, I, O)
2. Evaluate performance on the validation set using metrics such as precision, recall, and F1-score
3. Post-processing: Extract measurements from the grouped tokens, add domain-specific rules, etc.