BASF Measurements Parser

Problem Statement

The problem at hand is to develop an application that effectively addresses the challenge of extracting measurements information from research patents. The objective is to create a solution that can scan and parse the text of patent documents, enabling the accurate and efficient extraction of relevant measurement data.

Solution Overview

For the problem at hand, we use LLMs with Prompt Engineering to solve the problem. I divide the problem into subproblems:

Data Problem:

Given an XML file contains many patents, we follow these steps:

1- Split the patents.
2- We filter patents using <section> element to only include Chemistry patents.
3- We extract the test from the filtered patents, we use the text in <**description**>, <**abstract**>, <**claims**>, then put all text a txt file for later usage.
4- We further clean the text by removing unwanted characters.

Prompt Engineering Problem

We use Few-shots promotion, where we provide prompt instruction as well as some demonstrations. We follow prompt building principles; they are mentioned in the GitHub repo.

Evaluation problem

For the Prompt Evaluation, we follow two paradigms:

Context Evaluation

This Evaluation answers the question, Does LLM understand the prompt context? Does it understand the provided demonstration?

To answer this question, we use the examples in the prompts as inputs, and check whether GPT outputs the expected output or no. TO a prompt to be considered, the prompt need to score 100% accuracy for this evaluation.

Test set Evaluation

After Context Evaluation, how can we make sure that the prompt is not overfitting over the examples? We carefully choose a test set (with inputs and ground truth) from the provided patents manually, then use it as evaluation set for all prmpts, same test set is used for all prompts so we can compare.

Communicating with LLM

We use **Langchain** to split the text into documents and send it by chunks to the LLM. We also add the Retriever option, where we use the **Embedding model** to embed text then use a prompt to retrieve relevant documents before sending all documents to LLM.

Output format

The LLM output is not 100% guaranteed, so we must make a post processing function to process the output. We add the output format instructions to the prompt, we expect it to be JSON, so we just parse the output using **json.loads**, if the output format is good, it will be parsed, else we get error, if we get error, we further process the output to fix the json string, if not fixed, **we ignore it.**

Outputs

All outputs are in structured format (Json or Excel).