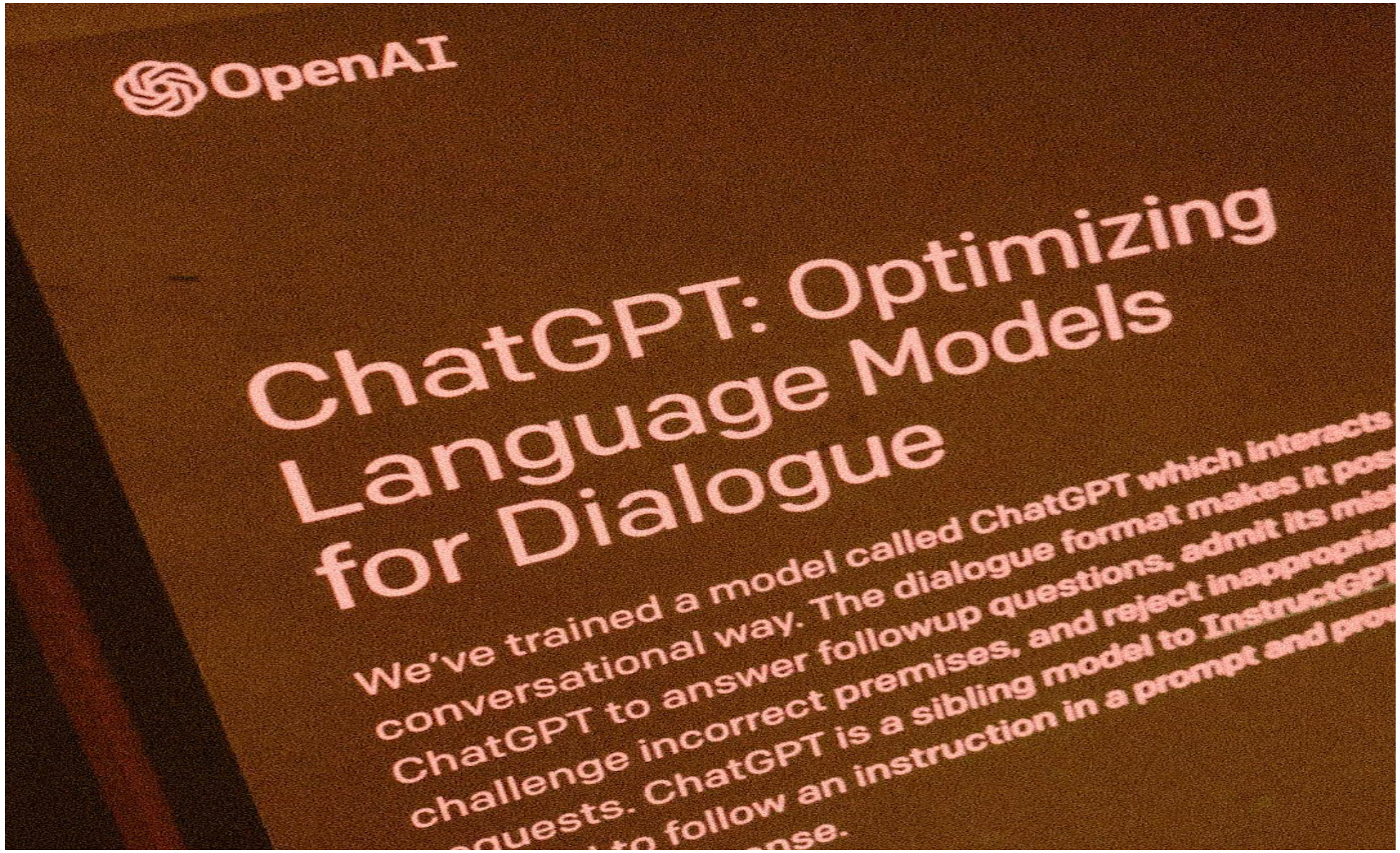
Product Descriptions with AI In Production

By: Patrick Ayers

PREVIOUSLY ON COMP 4449 CAPSTONE


OpenAI
ChatGPT: Optimizing Language Models for Dialogue
We've trained a model called ChatGPT which interacts
conversational way. The dialogue format makes it pos
ChatGPT to answer followup questions, admit its mis
challenge incorrect premises, and reject inappropria
quests. ChatGPT is a sibling model to InstructGPT
to follow an instruction in a prompt and pro
nse.

PART TWO: PRODUCT DESCRIPTIONS WITH AI IN PRODUCTION

Background Information

Who is Open AI?

OpenAI is a research and deployment company focusing on AI. They have made many great AI models in the last year.

- ChatGPT => Optimize language models for dialogue
- Dall-E2 => image classification and generation
- openAI codex => Natural Language to code

What model from Open AI did I use?

- ChatGPT API was not available at the time of this analysis, so I needed to use a GPT-3 model called text-davinci-003.

What was completed and learned in part one

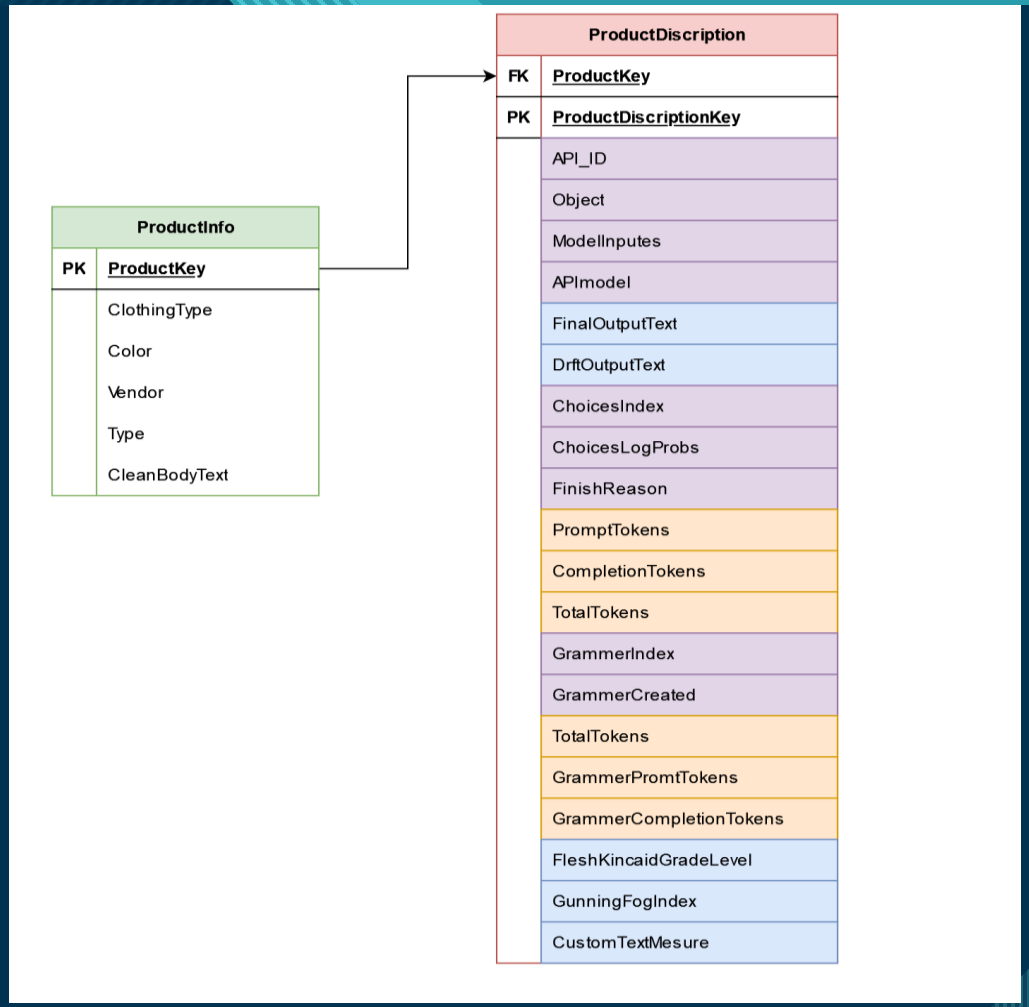
1. Generate product description using OpenIA
2. Compare the models with each other using sequencem matching and sentiment analysis
3. The cost of using chatGPT was less then $10 for 4.5Krequest

Outline of Presentation

- Database adjustments
  - Table additions
  - Information gained
- Features implementation
  - Grammar fixing
  - Text analysis for readability
  - Choosing the write product description
- Final process walkthrough
- Next step in the process

Database Addition

- We have added a new table called Product Descriptions
- Purple Cells are just metadata related to text generation and parameters
- Orange cells are related to the number of tokens that were used to make the API call. This is all related to cost
- Blue cells are the most critical.
  - Final output, this is after first tent generation and fixing grammar
  - Draft output, this is the text that is returned after the first API call
  - Flesh Kincaid Grade level Gunning fog index are used to measure the overall complexity of the text
  - Customer text measure is a value that is used to judge the quality of the text that is generated.



Features Implementation

Correction of Grammar

- Fixing the Grammar is being completed with OpenAI editor tool
  - Using the same model that was used for the draft of the product description
  - Instruction of fixing the grammar mistakes
  - N is the number of attempts
- It will return a diction of metadata
- Choices => text will be the product description after the grammar has been fixed
- Usage is a cost information

```
tempOutPut = openai.Edit.create(
    model='text-davinci-edit-001',
    input="Product Description Draft",
    instruction='Fix the grammar mistakes',
    n=1)
```

```
# Output
{
"choices":[{
    "index":Number,
    "text":Product Description eddited
    }],
"created": number,
"object": String,
"usage":{
    "completion_tokens":Number,
    "prompt_tokens":Number,
    "total_tokens":Number
}}
```

Writing Complexity Metrics

Flesh Kincaid Grade Level

- $(206.835) - 1.014\left(\frac{Total\ words}{Total\ Sentences}\right) - 84.6(\frac{Total\ syllables}{Total\ Words})$
- Scores are bounded between 100 and 0
- The lower the number, the more complex it will be to understand.
- College level will have a score of 50 – 30
- $8^{th}/9^{th}$ grade will have a score of 70 – 60

Gunning Fog Index

- $0.4[\left(\frac{words}{sentences}\right) + 100\left(\frac{Complex\ words}{Words}\right)]$
- Scores are between 0 and 17
- The higher the score, the more complex the writing will be
- A score of 13 is suited for a college freshman
- A score of 9 is suited for a high school freshman

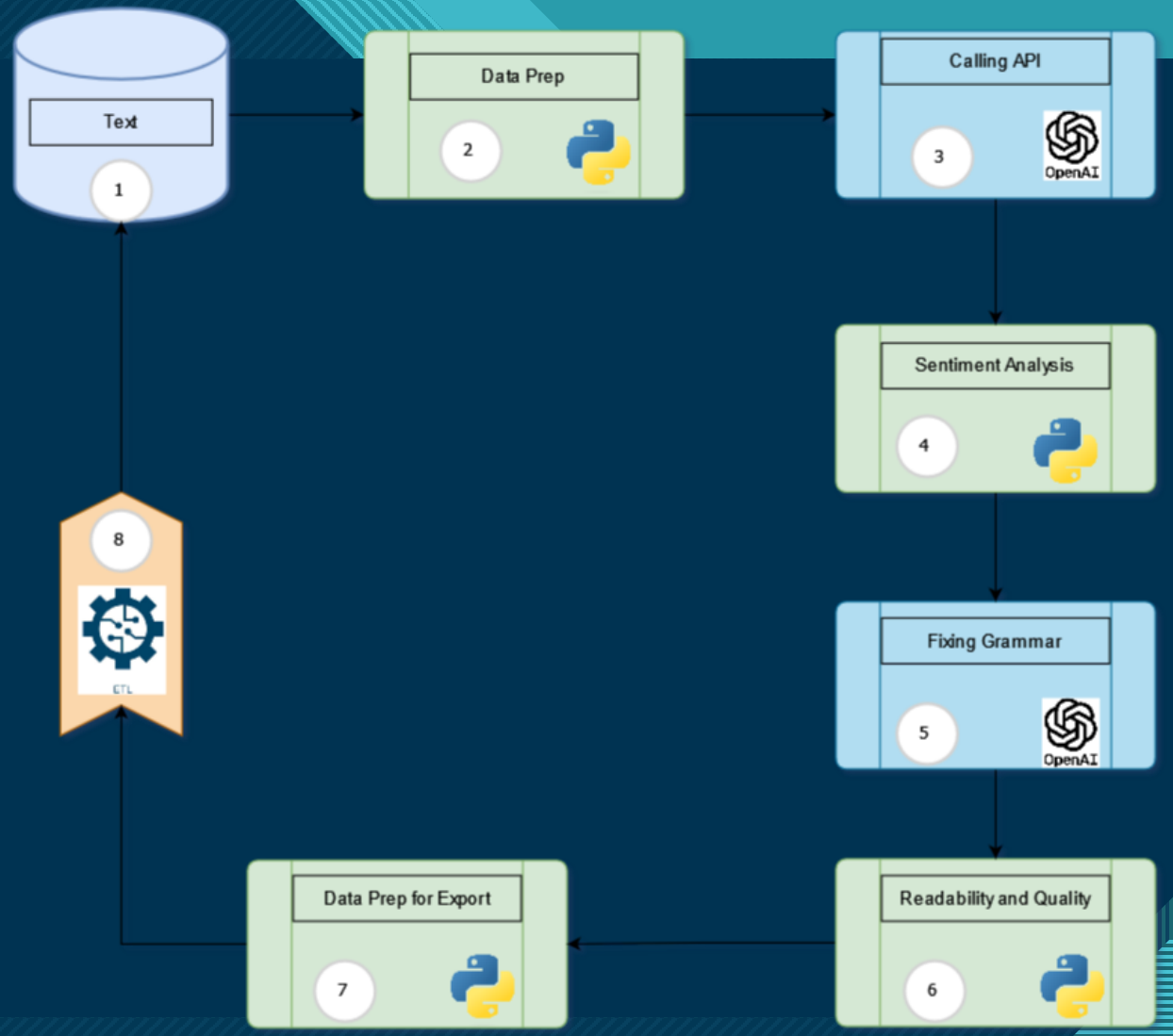
- These metrics have been added to ensure that the output of the product description is still readable for the average customer.

Product Description Quality Check

- This is used in conjunction with the other two text analysis
- This is not measuring the complexity of the text it measures the overall quality.
- A value closer to zero implies a lower quality and unbalanced text.
- $$\left[\frac{(Unique\ Word\ Count)}{(Total\ Word\ Count)}\right] * \frac{UDA\ Use\ Count}{(Number\ of\ UDA)} * \left[\frac{(ADJ\ Count)}{(Total\ Word\ Count)}\right]$$

Final Process

1. Look for items that need a product description
2. Data prepping for calling API
3. Running both sequence and sentiment analysis
4. Running Sentiment analysis and string matching function
5. Calling grammar fixing functions
6. Running readability and quality analysis
7. Prepping data for exporting
   - Converting Json to Dataframe
8. ETL to load data into a stage table and then append it to the product table



Next Step

- Reaching out to an e-commerce company to allow us to run this process on company data.
- This would allow us to try the process on live data and would also give access to other information that can give insight into how successful the model is
  - Customer engagement
  - SOE analysis
  - A/B testing

Thank You

Questions