

An abstract composition of various 3D rectangular blocks in shades of red, orange, teal, and light blue, arranged in a stepped, architectural fashion on the left side of the image.

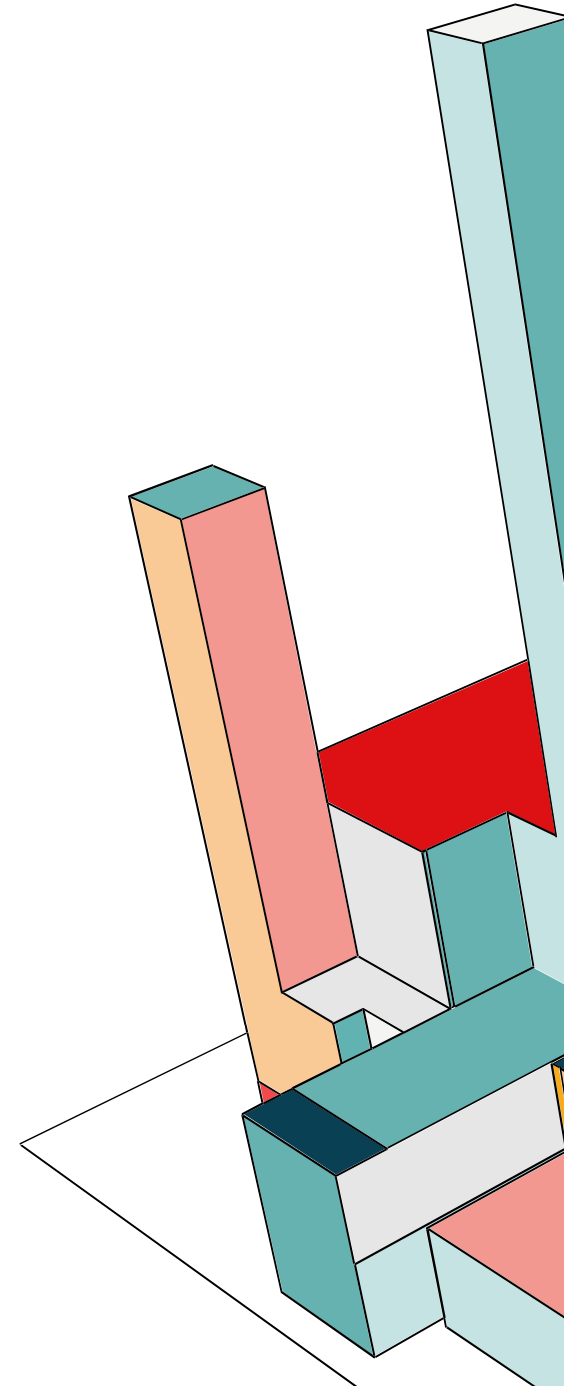
THOMAS MORE

BACHELOR PORTFOLIO

Charles Nana Kwkaye

TABLE OF CONTENT

1. Introduction
2. Assignment Description
3. The problem
4. Embeddings
5. Implementation options
6. Approach 1,2 and 3
7. Realizations
8. Project Deliverables
9. Update



BAD OR LACK OF DOCUMENTATION

- Slow or reduced efficiency
- Dependency on key personnel
- Increase in errors
- Delayed project timelines





WHAT IS VRT?

Vlaamse Radio- en Televisieomroeporganisatie, is the national public-service broadcaster for the Flemish Community of Belgium.



THE PROBLEM



Large DBT project



A lot of models,
packages and macros



Lack of
documentation



Difficult to understand
the queries when
navigating the project



DBT is an open-source command line tool specifically designed for data analysts and data engineers. It helps them develop, test, and deploy data transformations in a collaborative and reliable way



ASSIGNMENT DESCRIPTION

This project aims to explore the feasibility of using Large Language Models (LLMs) to document dbt queries and, if feasible, to implement a solution for the VRT data team's specific needs

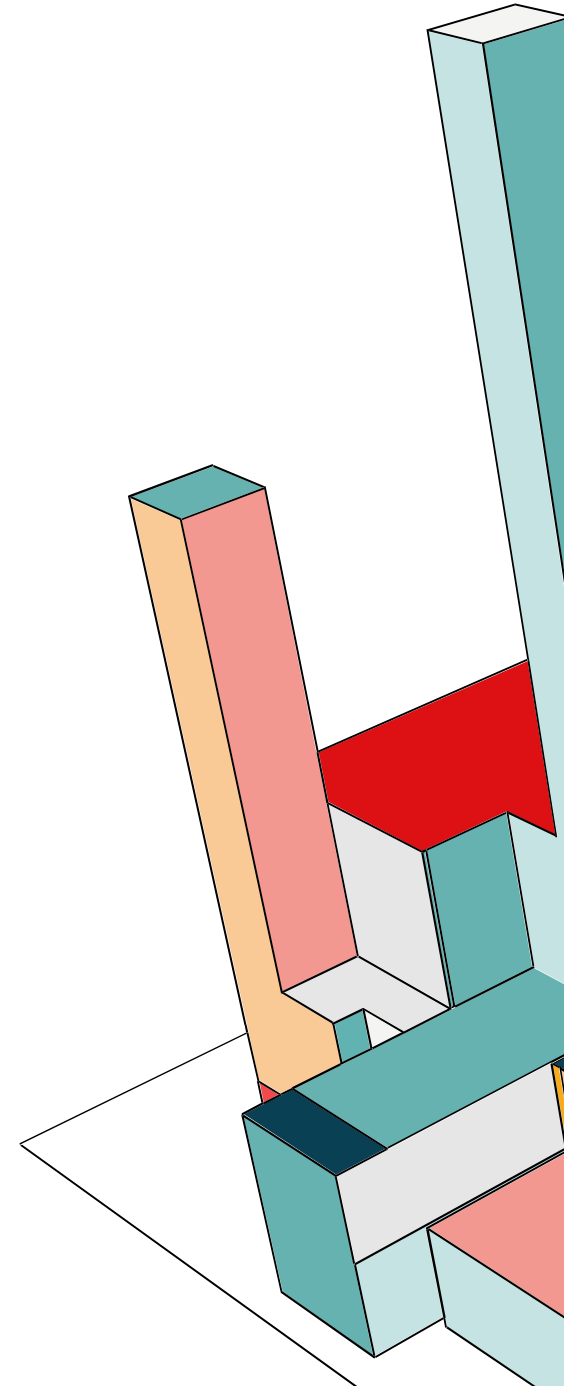


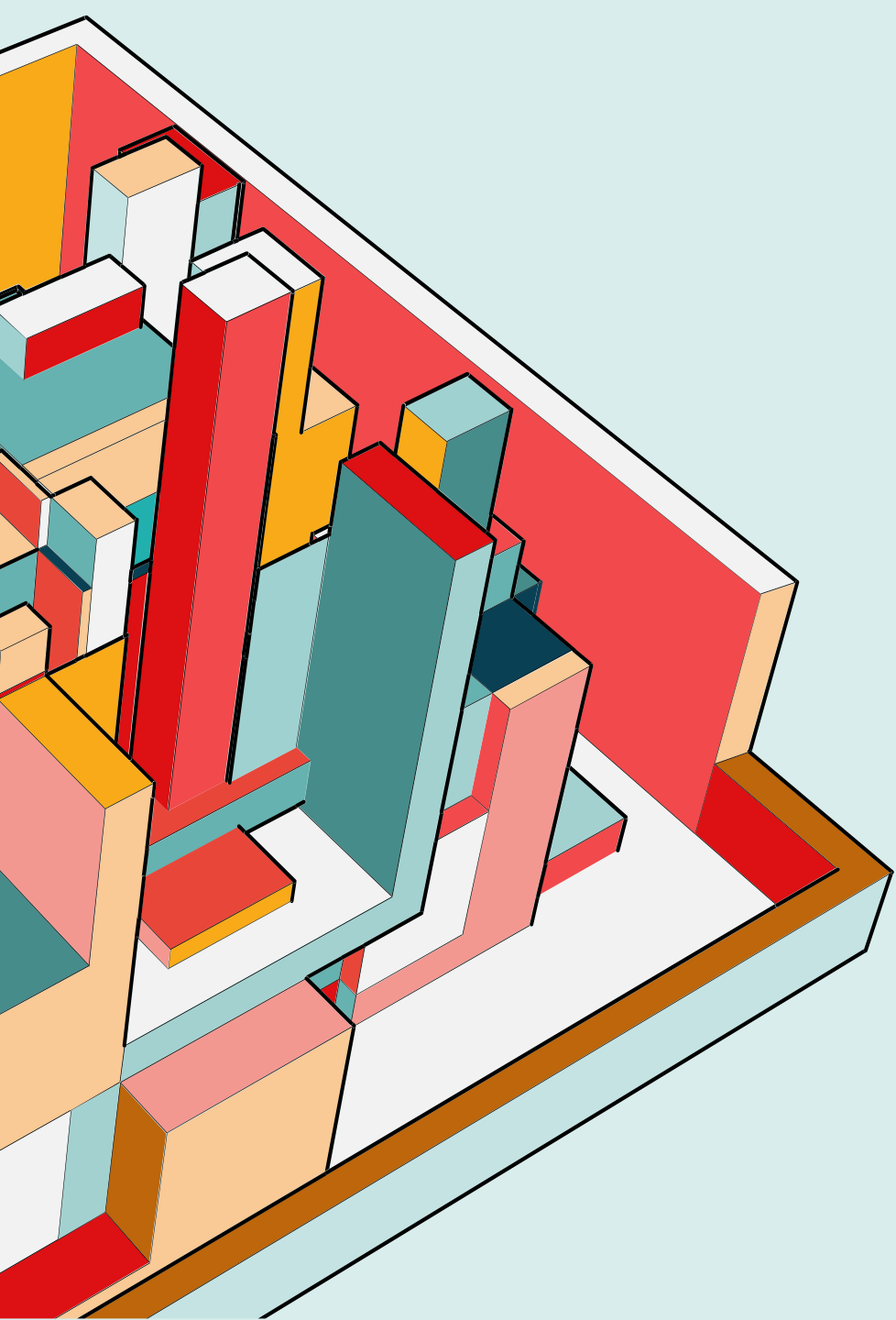
EXAMPLE OF DBT CODE

```
{{ config(
    materialized = 'incremental',
    incremental_strategy = 'merge',
    unique_key = ['timegranularity',
                  'kpi_date_id',
                  'first_touchpoint_platform',
                  'first_touchpointbrandgroup',
                  'marketing_channel_level',
                  'first_page_key',
                  'contactmoment_page_referrer_source'],
    table_type='iceberg',
    on_schema_change='append_new_columns',
    tags=["dafact"],
    partitioned_by = ['timegranularity', 'kpi_date_id']
) }}

{% set metrics_expression -%}
    {{ get_additive_marketing_metrics() }}
{% endset %}

{{ generate_marketingfact_dwm(metrics_expression = metrics_expression,
    prep_fact_table_ref =
    ref('prep_snowplow_contact_sessions_fact_enriched_grouped'),
    prep_fact_basetable_ref = ref('prep_snowplow_contact_sessions'),
    kpi_date = 'first_event_date',
    input_fields = ['timegranularity', 'kpi_date_id',
'first_touchpoint_platform',
'first_touchpointbrandgroup', 'marketing_channel_level', 'first_page_key',
'contactmoment_page_referrer_source'],
    include_first_field_null = true ) -}}
```





IMPLEMENTATION

CHOOSING A LLM FRAMEWORK

- Tools designed to simplify the development and deployment of applications powered by Large Language Models (LLMs).
- Have prebuilt components for prompt templates, indexing, retrieving, output parsing and chaining responses



VS



Langchain

General purpose framework for data processing, indexing, and interacting with LLMs

More flexible, allowing for customization of application behavior and chaining of complex LLM prompts

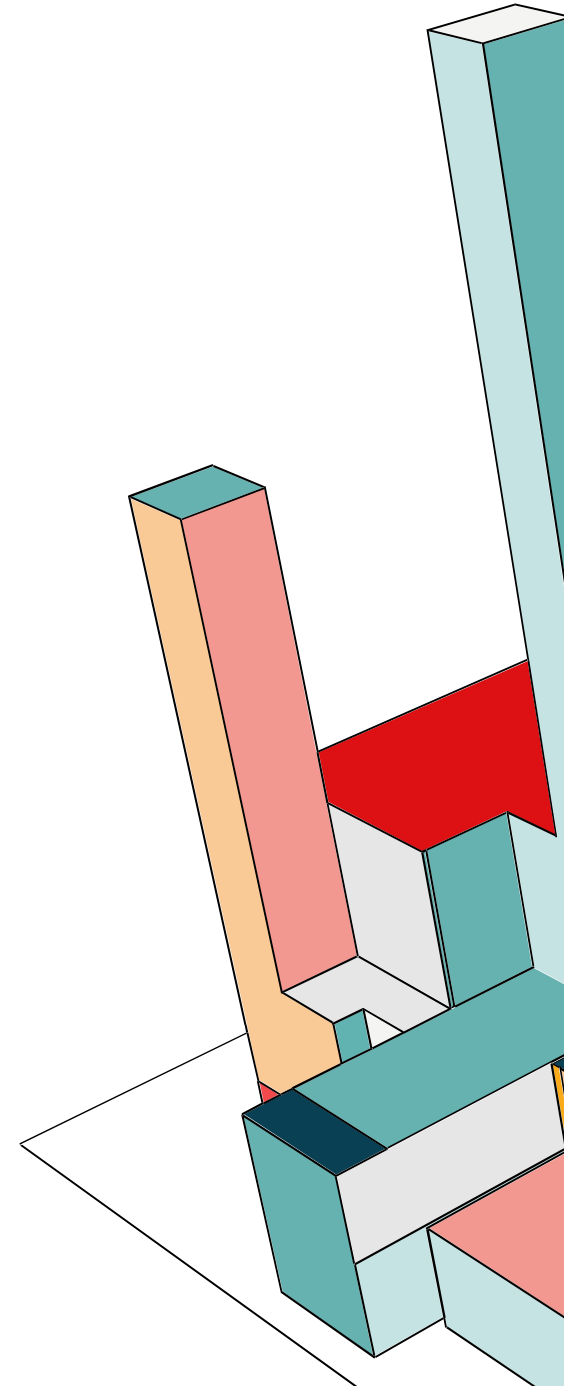
Has documentation of similar use cases

LlamaIndex

Specifically designed for building search and retrieval applications using LLMs

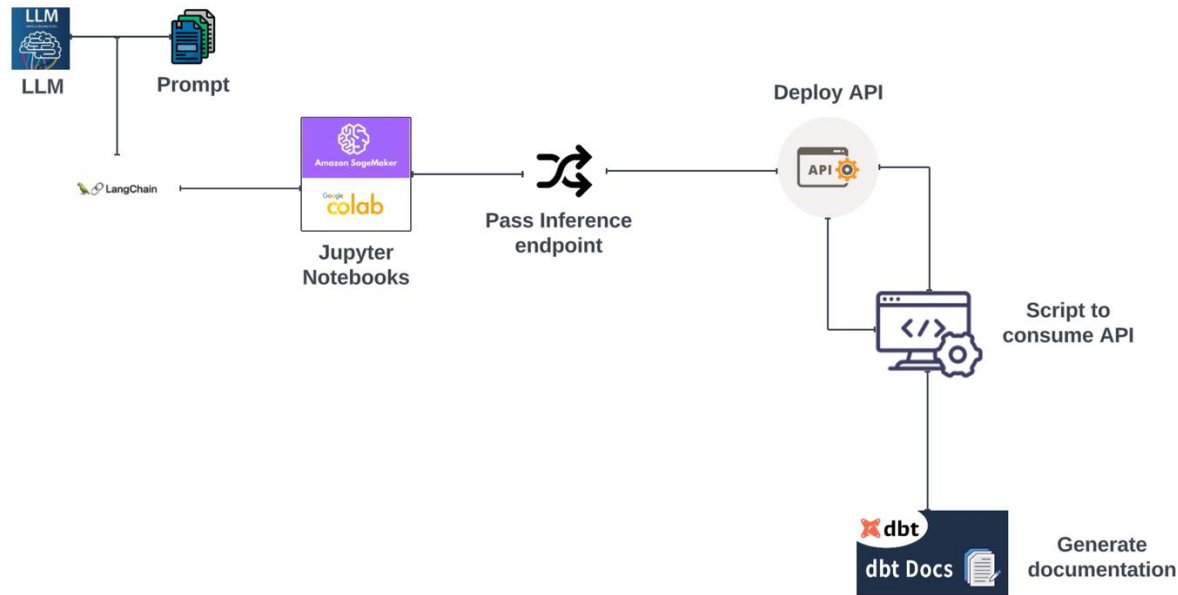
Less flexible, but provides a simpler interface for querying LLMs and retrieving documents.

Does not have documentation of similar use cases



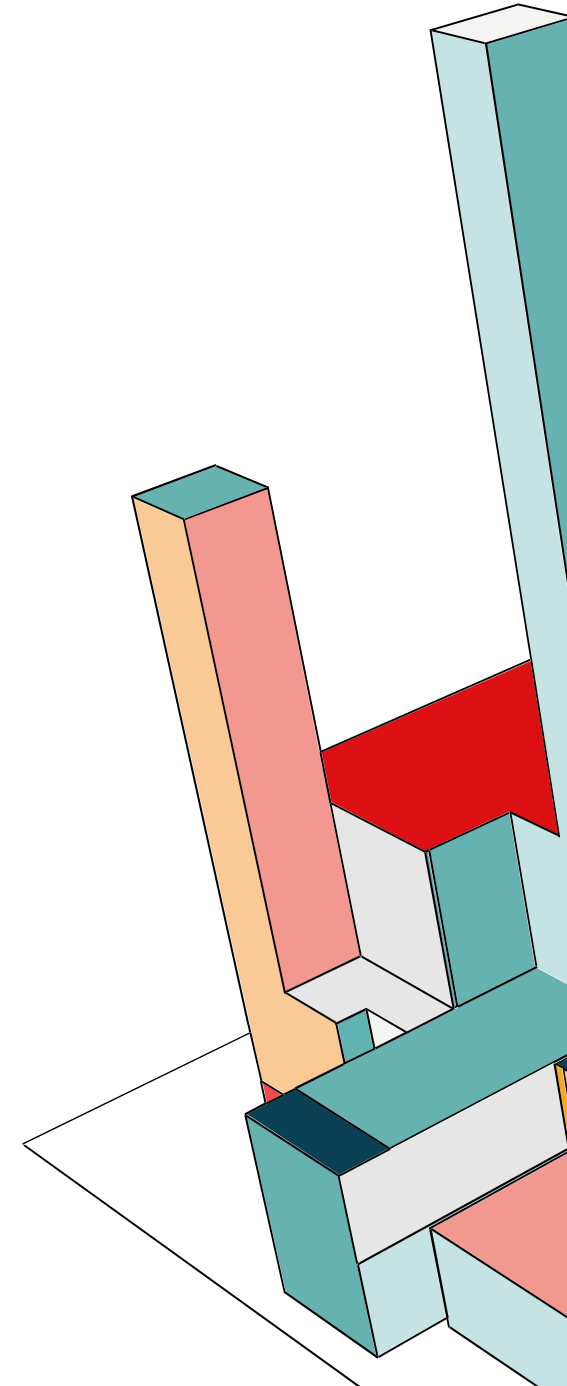
APPROACH 1

PROMPT BASED



Reason to choose

1. Simplicity: Straightforward and minimal infrastructure or coding
2. Flexibility: Easily modify prompts for different dbt query types and documentation style



REALIZATION FOR APPROACH 1



Incomplete text generation from model



Model relied heavily on quality of user prompts

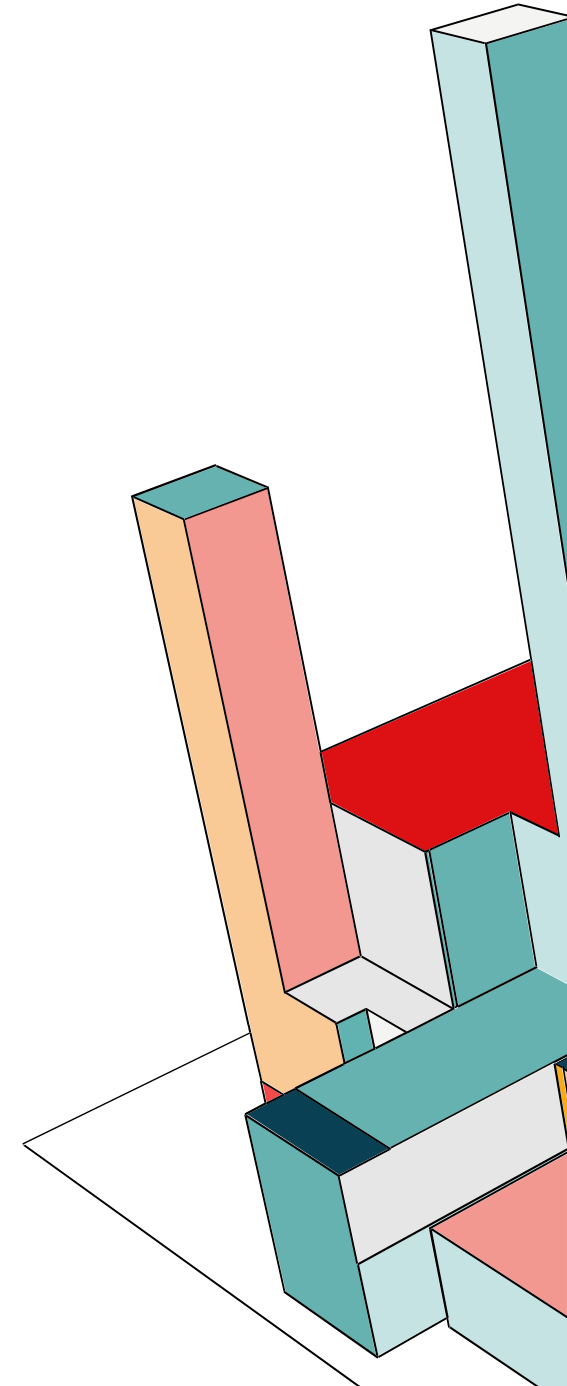
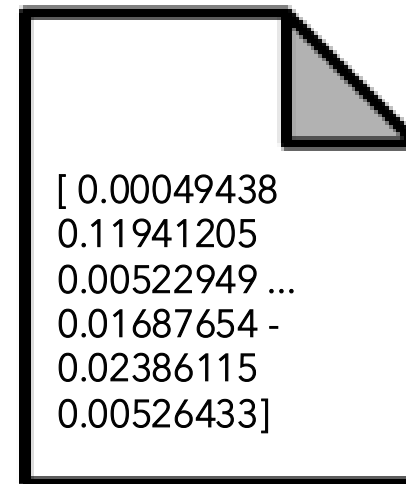
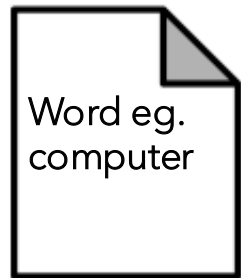


No memory

WHAT IS EMBEDDINGS

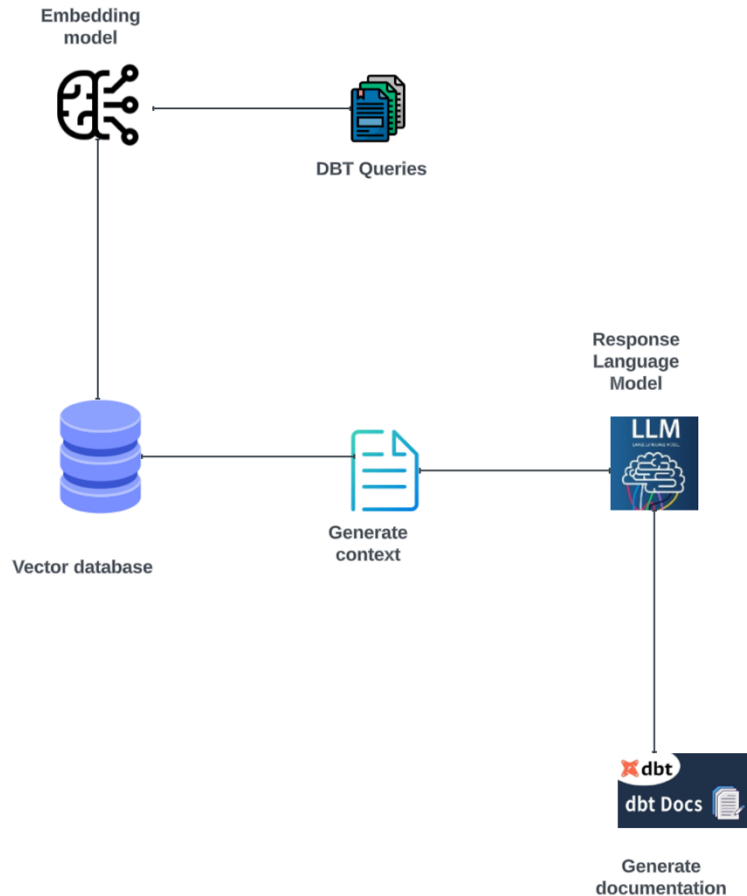
- Turning complex data (words, images, etc.) into simple lists of numbers.
- Like a "secret code" for computers to understand meaning.
- Similar things have similar codes, making connections easier.

Example:



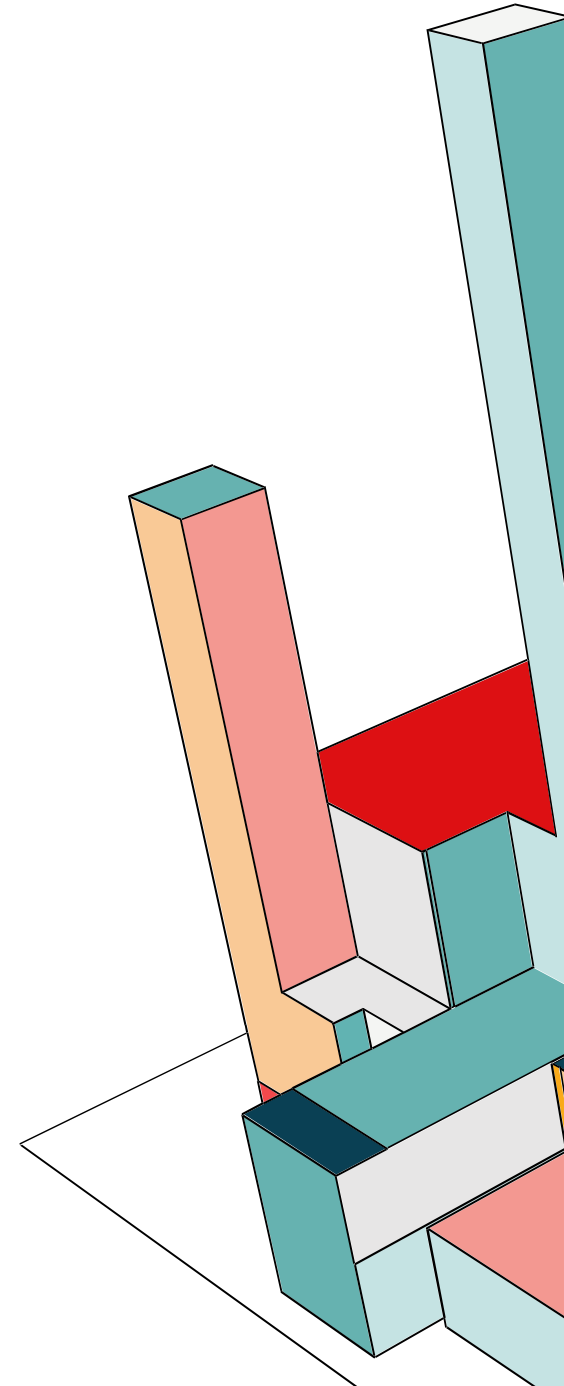
APPROACH 2

USING EMBEDDINGS

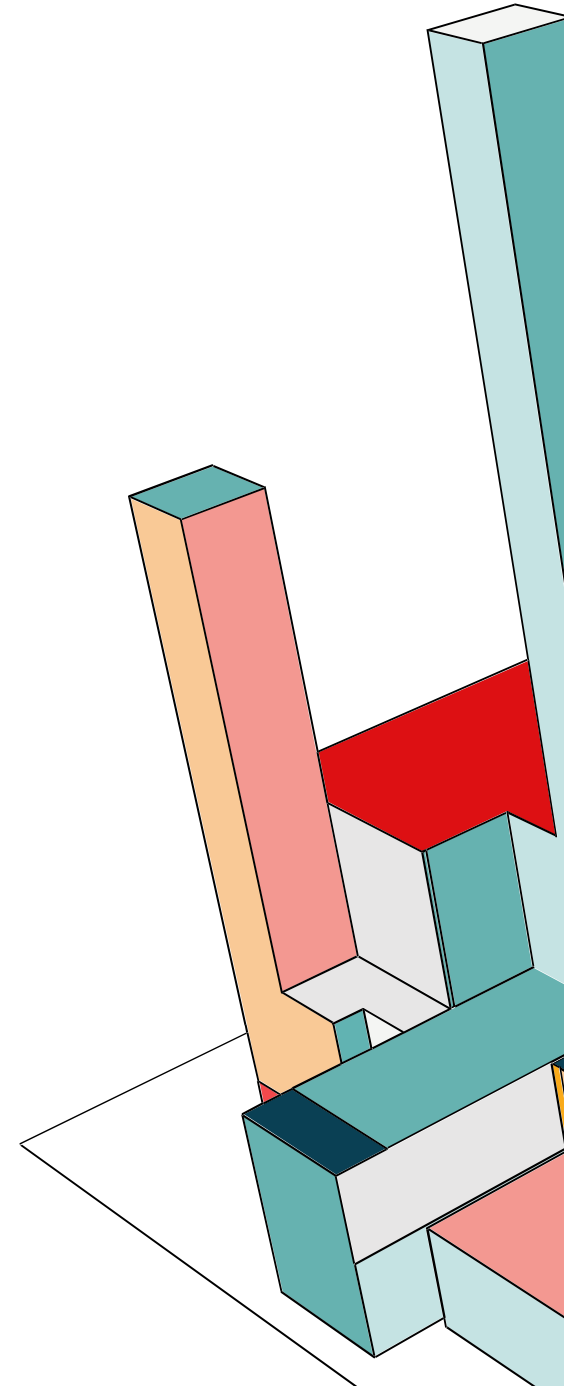
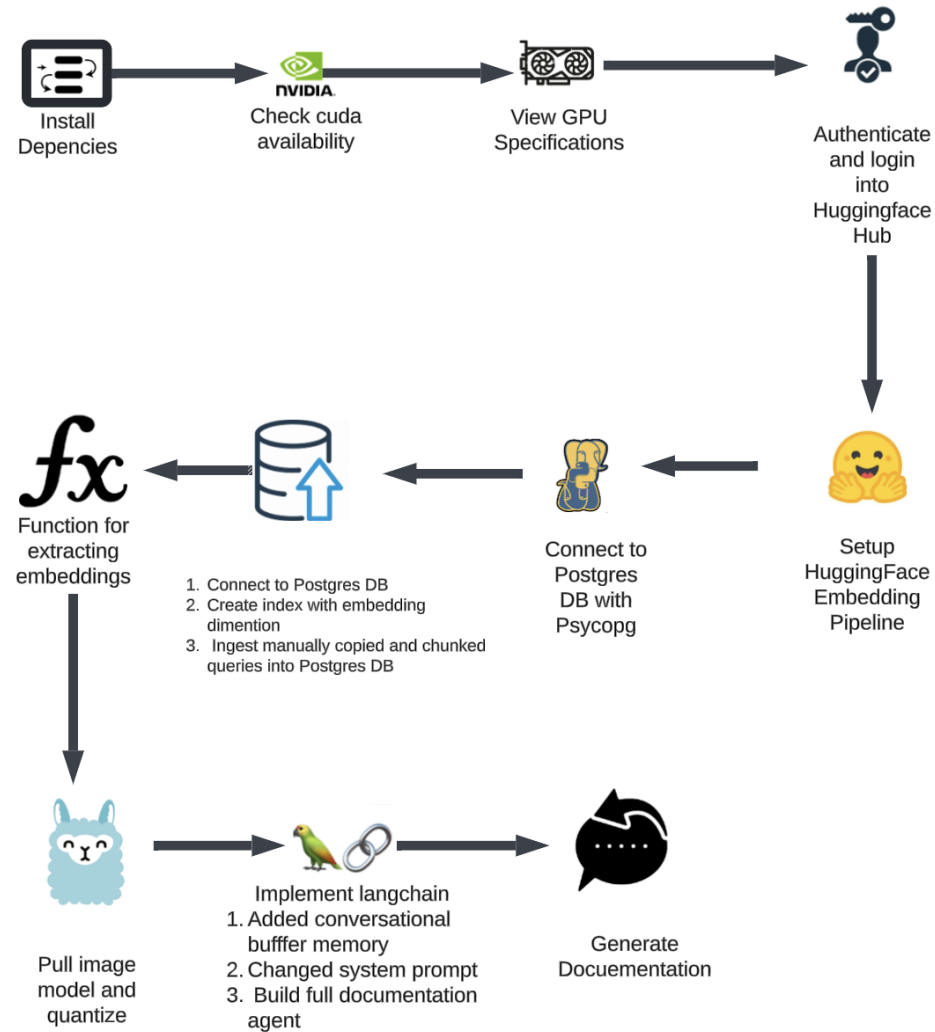


Reason for choosing

1. System prompt: Change default system prompt
2. Reusability: Redocument queries easily if documentation not up to par
3. Scalability: Can handle large dbt datasets more efficiently



APPROACH 2 NOTEBOOK WORKFLOW



AFTER SYSTEM PROMPT AND JSON STYLE RESPONSE

PROMPT

```
Human: [INST] Imagine this dbt query as a recipe. Can you walk me through the key steps it takes to
ingredients (data) into the final dish (output) in a simple and summarized documentation? Focus on
and complex parts of the transformation process, avoiding unnecessary details. [/INST]
User: {{ config(
    materialized = 'incremental',
    incremental_strategy = 'merge',
    unique_key = ['timegranularity',
        'kpi_date_id',
        'first_touchpoint_platform',
        'first_touchpointbrandgroup',
        'marketing_channel_level',
        'first_page_key',
        'contactmoment_page_referrer_source'],
    table_type='iceberg',
    on_schema_change='append_new_columns',
    tags=["difact"],
    partitioned_by = ['timegranularity', 'kpi_date_id']
) }}

{% set metrics_expression = %}
    {{ get_additive_marketing_metrics() }}
{% endset %}

{{ generate_marketingfact_dwm(metrics_expression = metrics_expression,
    prep_fact_table_ref = ref('prep_snowplow_contact_sessions_fact_enriched_grouped'),
    prep_fact_basetable_ref = ref('prep_snowplow_contact_sessions'),
    kpi_date = 'first_event_date',
    input_fields = ['timegranularity', 'kpi_date_id', 'first_touchpoint_platform',
        'first_touchpointbrandgroup', 'marketing_channel_level', 'first_page_key', 'co
referrer_source'],
    include_first_field_null = true ) -}}

{{ dbt_project().get_project_config().get('target').get('schema') }}.marketing_fact dw
```



RESPONSE

```
...
Please help me summarize the key steps in transforming the data from the raw ingredients to the final output.

assistant: Here's a summary of the key steps in transforming the data from the raw ingredients to the final output:

**Step 1:** The `dbt` project configuration sets the target schema and materialization strategy for the marketing fact table.

**Step 2:** The `get_additive_marketing_metrics()` macro is called to generate the additive marketing metrics expression.

**Step 3:** The `generate_marketingfact_dwm()` function is used to create the marketing fact dimension table. This function takes several inputs, including the metrics expression, the prepared fact table reference, the base table reference, and various date-related parameters.

**Step 4:** The function applies transformations to the data, including grouping, aggregating, and filtering based on the specified criteria.

**Step 5:** The resulting transformed data is written to the marketing fact dimension table in the target schema.

The key takeaway is that this dbt query transforms raw data from multiple sources into a refined and aggregated dataset, ready for analysis and reporting. The query uses advanced techniques such as macros, functions, and conditional logic to achieve its goals.
```

REALIZATION FOR APPROACH 2



LLM Documentation Generation Incomplete: Query Length Exceeded Token Limit



Documentation improvement relies heavily on quality of prompts



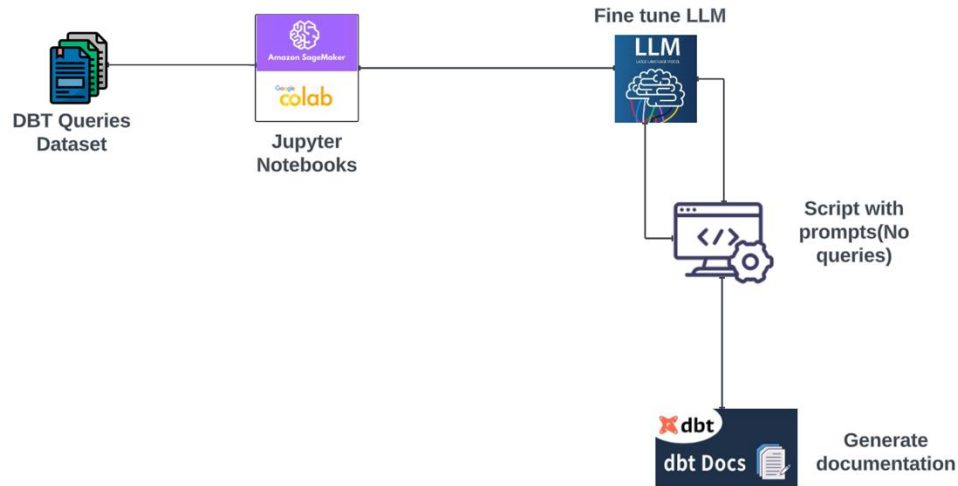
Has conversational buffer memory so it can be prompted in a series



Difficult to choose a well balance embedding model

APPROACH 3

FINE TUNING LARGE LANGUAGE MODEL



Reason to choose

1. Anticipated to be the highest accurate: Fine tuning enhances models understanding of the domain and improves documentation accuracy

17

ANTICIPATION FOR APPROACH 3



Data preparation for fine-tuning would be time-consuming and complex.



Would be difficult to maintain and update the fine-tuned model as dbt queries evolve. (Overfitting and loss of interpretability)

ANTICIPATION FOR APPROACH 3



Data preparation for fine-tuning would be time-consuming and complex.



Would be difficult to maintain and update the fine-tuned model as dbt queries evolve. (Overfitting and loss of interpretability)

DIFFERENCE BETWEEN APPROACHES

Prompt based approach

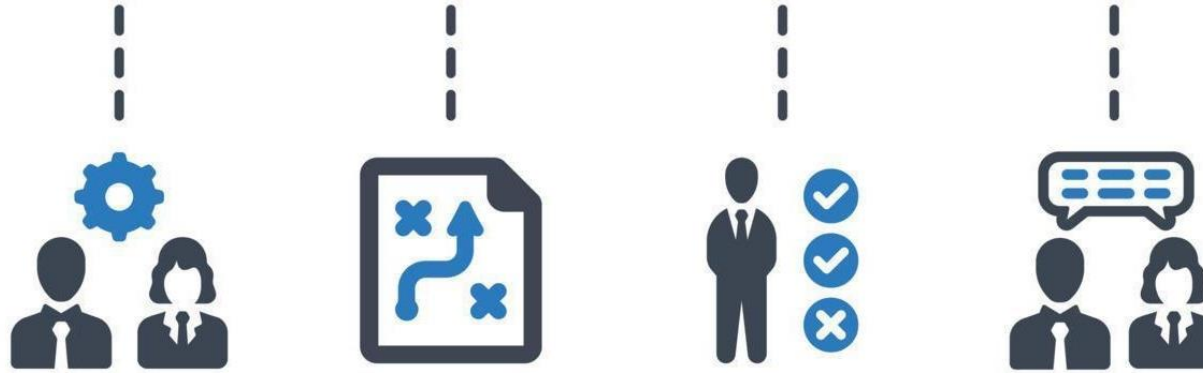
Embeddings

Fine tuning

Approach	Key-feature	Additional Requirements
1.Prompt-Based	Simple prompts to LLM	None
2.Embeddings	Embeds queries and metadata	Vector databse, Embedding model
3.Fine-tuning	Directly trains LLM on dbt queries	Large dataset with some documentation of dbt queries required for training

Soft skills

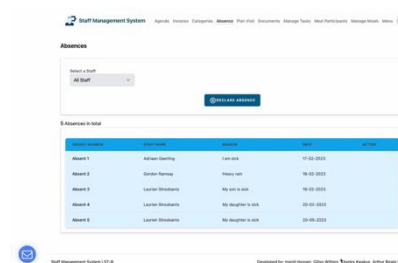
SOFT SKILLS



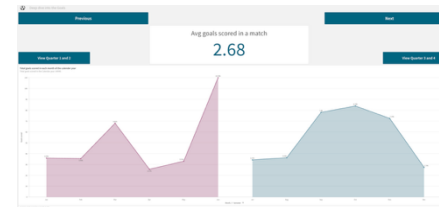
Teamwork Problem Solving Decision Communication



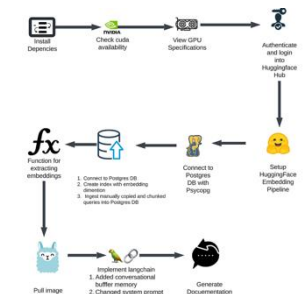
- Waterwatchers (6 people team)
- Shoestory (3 people team)
- Staff management system (5 people team)



- Staff management system (Document lead)



- Football Data Visualization (Ownership)
- Shoestory (Team Lead)



- VRT Internship project

Reflection, Growth and Goals



Charles Nana Kwakye **ROADMAP VISION**

Now 21 years

40 years

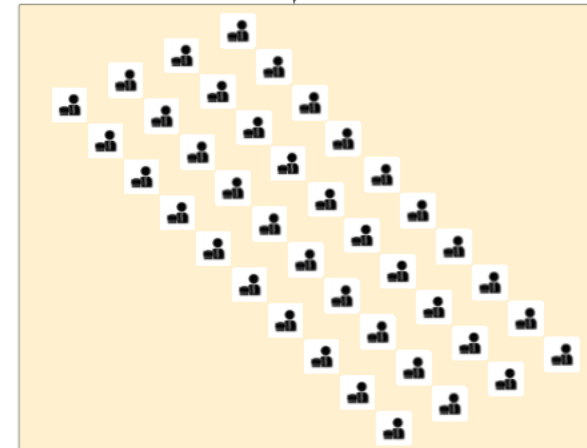
60 years



ME



COMPANY



FREELANCE EMPLOYEES

THANK YOU

Charles Nana Kwakye

