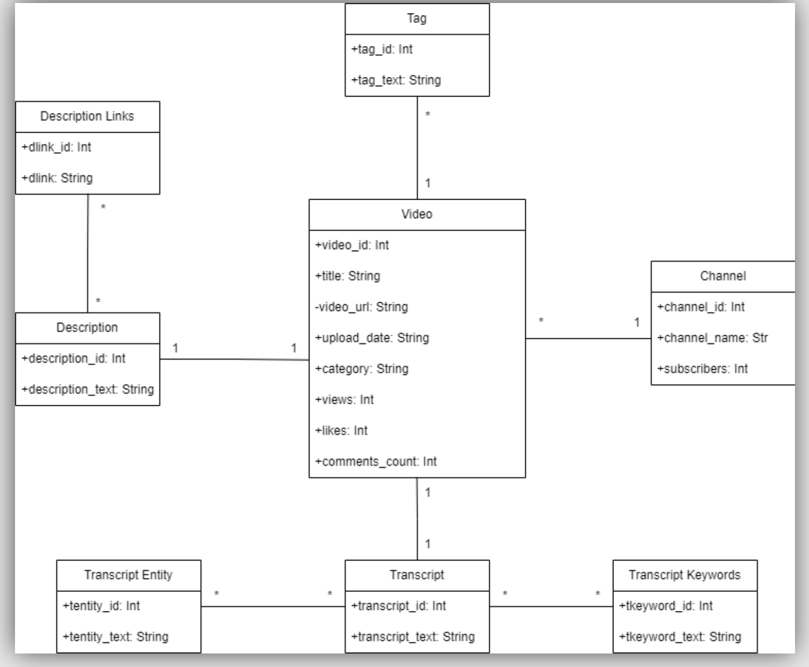
TechSpotter

- David Cordeiro – up202108820
- Diogo Viana – up202108803
- Gonçalo Martins – up202108707
- Sofia Minnemann – up202007342

Milestone 1 Overview

Metadata and Transcript of YouTube videos from popular English-Speaking Tech Youtubers, that have an available, manually written, English transcript, saved in JSON files.



Documents

The final dataset is structured as individual documents, each representing a YouTube video with their respective attributes.

```
{
    "id": “”,
    "title": "",
    "channel": "",
    "channel_id": "",
    "subscriber_count":,
    "upload_date": "",
    "video_url":””
    "category": "",
    "tags": [],
    "views": ,
    "likes": ,
    "comments_count": ,
    "description": "",
    "description_links": [],
    "transcript": "",
    "transcript_keywords": [],
    "transcript_entity_values": [],
    "transcript_entity_types": []
}
```

Milestone 2 Overview

Key Focus: Inicial version of the search engine retrieval system.

Solr Configurations:

- Setup 1:
  - Solr automated indexing process
  - Queries the fields: “transcript”, “transcript_keywords”, “transcript_entity_values”, “title” and “description”.

- Setup 2:
  - Custom Schema
  - Queries with the same fields in setup 1, boosting “transcript” by 2 and “title” fields by 3

Experiments

During the milestone, many experiments were conducted to improve the search engine. However, some of these experiments did not achieve the desired improvements:

- Tie-Breaker: adjusted flexibility
- Minimum Match: partial match conditions
- Phrase Slop: controlled word proximity
- Phrase Fields: boosted exact phrases in fields
- More Like This: analyzed key terms to find similar results

Enhancing Information retrieval system

In milestone 2 we implemented the initial version. In this version. The custom schema had

- Field types: text_en, string
- Filters(LowerCaseFilter, StopFilter , SynonymFilter and SnowballPorterFilter)
- Field boosts

Now:

Hypothesis: What if semantic search could be implemented alongside the lexical search?

Used semantic search to find more relevant documents.

Boosters tuned: Reduced "title" boost from 3 to 1.5 and "transcript" boost from 2 to 1.5

Query 1

AI Ethical Concerns

Information Need: *As an AI Developer, I want to explore the ethical concerns regarding Artificial Intelligence.*

Query

- q: "ethics artificial intelligence debate discussion concernsnews“
- fields: "id, score“
- qf: "transcript, transcript_entities, transcript_keywords, title, description“
- rows: 50

Query 2

Building a gaming PC with guides

Information Need: *As a PC building newbie, I want to find Step-by-Step guides to build a gaming PC.*

Query

- q: "how build a gaming pc tutorial“
- fields: "id, score“
- qf: "transcript, transcript_entities, transcript_keywords, title, description“
- rows: 38

Query 3

Durable laptops

Information Need: *As a user seeking longevity, I want to find durability tests in laptop reviews.*

Query

- q: "durability test laptops laptop“
- fields: "id, score“
- qf: "transcript, transcript_entities, transcript_keywords, title, description"
- rows: 18

Query 1

Metric	System 1	System 2	System 3
AvP	0.3112	0.7971	0.8015

Query 2

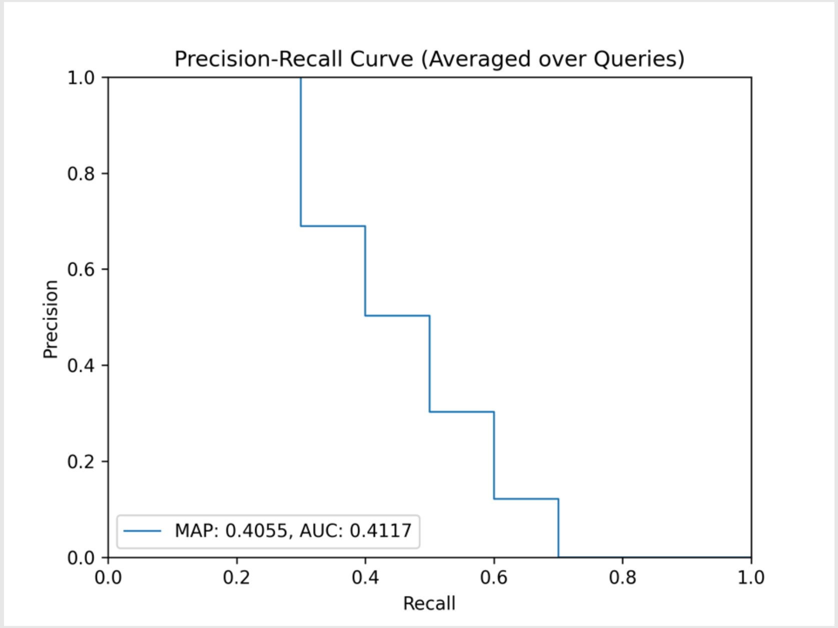
Metric	System 1	System 2	System 3
AvP	0.5682	0.6184	0.7834

Query 3

Metric	System 1	System 2	System 3
AvP	0.3373	0.3056	0.6597

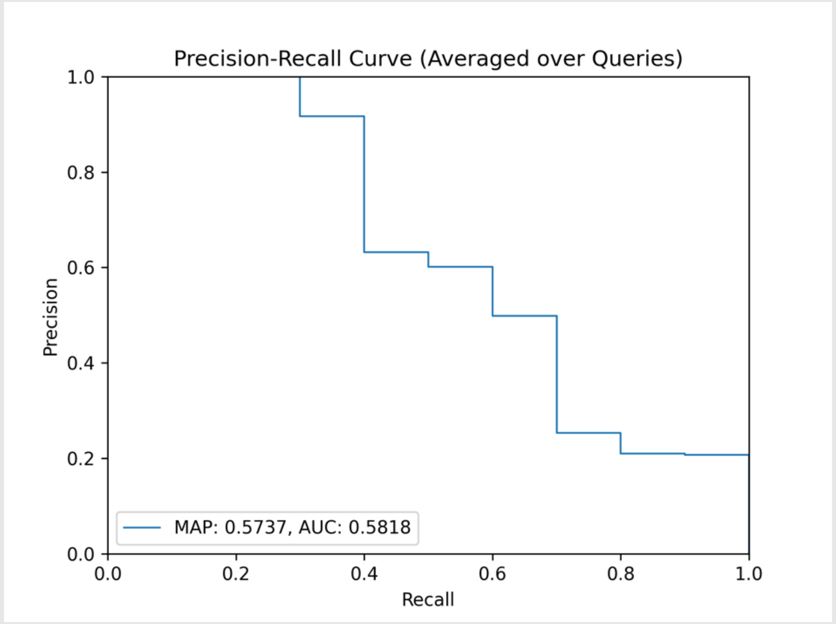
Performance Metrics

System 1



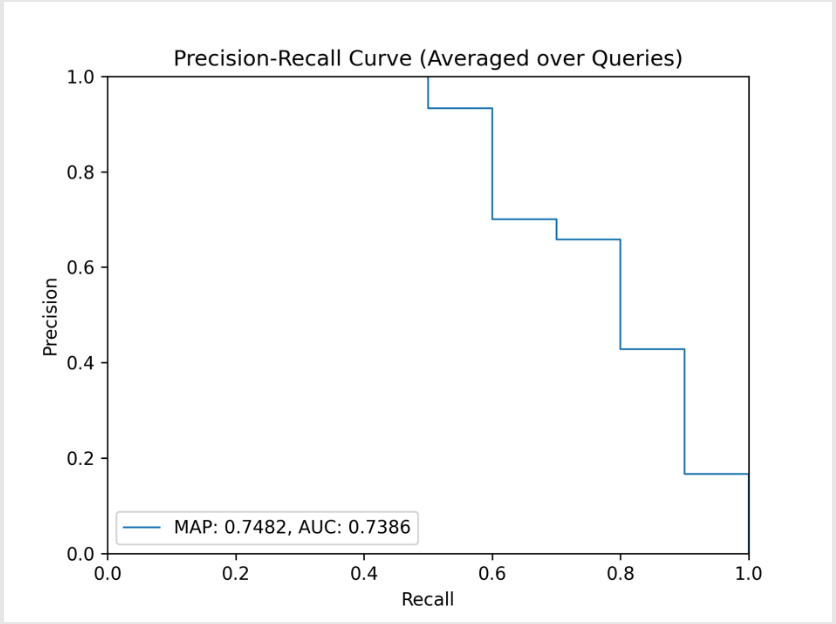
Performance Metrics

System 2

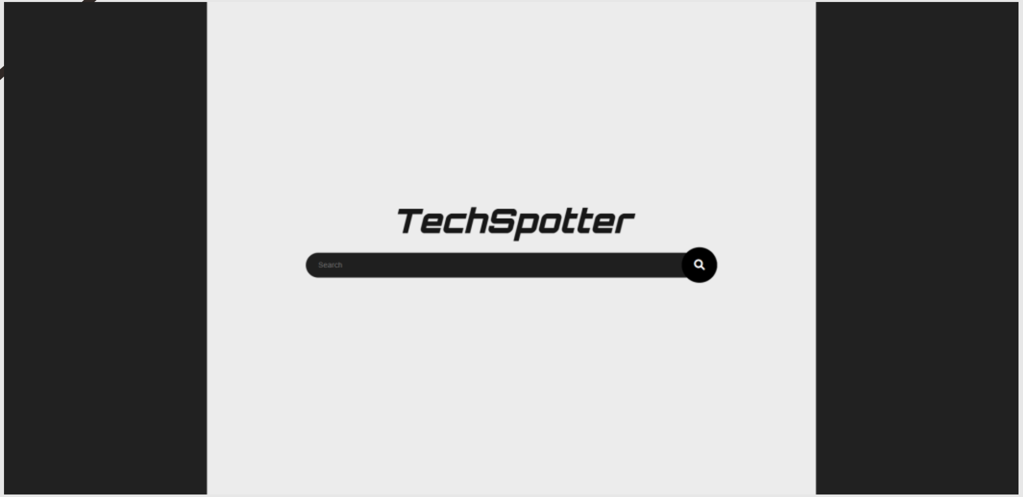


Performance Metrics

System 3



User Interface



We created a simple user interface to provide a seamless and intuitive way to use queries and interact with the search engine.
We used flask API integration for search queries
The interface tracks user clicks to dynamically update search results, re-ranking videos based on user behavior.

Final System Overview:

Key Features:

Hybrid Search (Lexial + Semantic)

- Combined lexical precision with semantic context understanding
- Normanlizes and Weighted scores for balanced results

User Interface Integration:

- Seamless UI with Flask API for search queriess

Dynamic Re-ranking Based on User Behavior:

- Clicl-through data tracked via User Interface
- Documents re-ranked based on user interactions (sored in JSON)

Questions?