

TECHNICAL SPECIFICATION

# TASK 15

## Text search service for media content



## **Technical specification "Text search service for media content"**

### **1. Relevance of the task**

The search function is relevant in the modern world due to the large amount of information available on the Internet. Users are looking for information about products, services, news and other topics, and the search function allows them to quickly and easily find the information they need. Additionally, the search feature is used on many websites and apps to enhance usability and improve the user experience.

"Yappy" is a social network of short vertical videos. Authors upload videos to the platform, enriching them with descriptions. The task of searching for content on request is relevant for the social network "Yappy". The quality and speed of content delivery for search queries is an integral function of the application. Using search, users can find content that interests them, discover previously unknown content categories, be inspired by new ideas, and watch current videos from different categories.

The current version of the application already has a search function. Now the search is carried out by searching for similar words in the description under the video that the user leaves. The description includes plain text and hashtags. A hashtag is text with a "#" symbol in front of it that briefly describes the main meaning of the video, for example: #food, #recipes, #delicious, #yum-yum. Very often, content authors leave incorrect descriptions and hashtags under videos. This affects search results. More features need to be included in the search, especially those that can be extracted directly from videos, for example using machine learning and deep learning models.

Your solution should improve the current search system and make the delivery of content better. This will increase the popularity of this feature among users of the social network, and in turn, increase the overall time spent by users in the application.

The target audience of the service is young people aged 14 to 35 years.

### **2. Description of the task**

Participants are invited to solve the following case - to create a "Smart Search" service for the Yappy social network. The social network "Yappy" is a platform of short vertical videos up to 1 minute long, uploaded by user-authors. Each video is enriched with a description compiled by the author himself.

The user enters some text in the search bar of the application. It is necessary to use the most relevant videos from all their diversity for its text. Currently, the search engine only takes into account the text from the description under the video. Note that for a search engine to work successfully, some index is required. An index refers to specially prepared signs for each entity - video, which is searched. You may develop any features and use any tools to store this information, except proprietary tools with closed licenses. Only open-source tools may be used.

In fact, the task is divided into two subtasks:

1. Processing a new video. Enriching it with additional features using special algorithms and machine learning models. Adding this information to the index repository.
2. Direct search. Providing the most relevant content to a text request from the user. It should be taken into account that the user can enter several letters, words, phrases, sentences, or any combination of characters.

The data provided to solve the problem consists of an mp4 video file and a description of the video in text format. The search system can take into account different entities, such as: a user description under the video, speech in the video itself, text in the video itself, a description of the video obtained by some system or neural network. The generation of such features can be carried out using any open-source or self-written solutions/tools.

The best search service can be integrated into the social network “Yappy”.

### **3. Possible custom path**

The user logs into the “Yappy” social network, goes to the search tab and enters his query into the search bar. It could be a set of letters, a word, a phrase or sentence, a set of incoherent symbols. The goal of the search engine is to give him the most relevant videos from the entire variety of videos uploaded to the platform. At the same time, the user can apply filters and sorting to the search results. For example, sorting the results by date added, popular first, etc.

For the user, the most important search parameters in the application are: the quality of content delivery and the speed of the system.

### **4. Solution requirements**

The solution must be a service that implements a search engine. The code for creating a service can be written in any programming language, but the most preferred options are Python or Go. When developing a search system, it is allowed to use open libraries, tools, algorithms, and machine learning models. The use of closed services that operate on a subscription basis or provide paid services, as well as those operating under a license that is not open to commercial activity, is prohibited.

Mandatory conditions for the final solution include:

1. Availability of a search system API with two main endpoints - public interface methods. One endpoint accepts a video with a description as input, or rather a link to the video and description. Extracts attributes using some set of algorithms. Puts all the necessary information into the index

storage. Another endpoint accepts a search query as a text string as input and produces a list of videos in the response.

2. System speed. No more than 5 minutes are allocated for video reception and indexing. It takes no more than 500 milliseconds (0.5 seconds) to issue a search query.
3. Provide the solution with a brief description of the logic of the system, the main algorithms and the tools used.
4. Deploy your service to make quick search queries to it. By service we mean a running backend with two active endpoints.

It would be a plus:

1. Come up with hints for auto-completion of the user's request. Example. The user enters the letters "car". As a hint, he is given a list from the auto-completion: "cartier", "carnival", "carnival", "carpe diem". As an illustration, you can consider examples of search results in Google/Yandex.
2. Describe the further development of the system. Even if something could not be done within the Hackathon time, you can provide the solution with ideas for improving/refining your system.
3. Describe the limitations of the current system in terms of the number of received requests per second, the speed of index calculation/recalculation, etc.
4. Create a typo correction tool for the user. For example, a user enters the word "bwaity", and the typo correction algorithm understands that the entered word is "beauty" and searches for the corrected word.

## **5. Presentation requirements**

The presentation of the final solution is sent in pptx/pdf format. Required slides include:

1. Team information;
2. Description of the advantages and disadvantages of the selected solution;
3. Description of the architecture and tools for the solution;
4. Description of the main algorithms and machine learning models involved in solving the problem;
5. Indicating the speed of the system for two main tasks: adding a new video to the index, direct search.

## **6. Requirements for accompanying documentation**

Accompanying documentation is sent in docx/pdf format. Mandatory items in the documentation are:

1. Description of the project structure;
2. List of tools/programming languages/libraries/open-source solutions used in the work;
3. The speed of the system in terms of two subtasks: inserting a new element into the database/index; responses to the user's text request.
4. Ideas and hypotheses about improving/refining the system that were not completed during the Hackathon.
5. Scheme of the logic of interaction of all modules within the project.
6. Indication of the uniqueness of the solution.

## 7. Data sources

The organizer will provide a data set of 400,000 videos with descriptions. Videos in the dataset will be presented as a link to a public repository.

Sample data:

```
{
  "video_link":
https://cdn-st.ritm.media/media/00/5d/fa5bcb3d40479d06d416d43a62ba/fhd.mp4,
  "description": "Do your cats also keep you up at night?
#top #marathoncontentyappy #iammarathonrunner #cat"
}
```

The videos will be available throughout the Hackathon; there is no need to download them.

## 8. Requirements for submitting solutions

The solution should be compiled into a single repository with code.

1. Link to an open repository with search service code
2. Link to solution presentation
3. Link to the deployed prototype of the solution
4. Link to accompanying documentation in format (docx/pdf)

## 9. Criteria for evaluation

Participants' solutions will be assessed based on the following main criteria:

1. The team's approach to solving the problem (completeness of the search system implementation, idea for solving the problem, set of selected algorithms and machine learning models) - 10 points;
2. Technical development of the solution:

- a) The effectiveness of the solution within the given task (the speed of the service for two main tasks) - 10 points;
  - b) Solution performance, solution supportability, code quality (10 points);
  - c) Innovative ideas to improve the user experience of interaction with the application's search engine (original solutions and proposals for improvement, solving additional problems) - 8 points;
- 3. Compliance of the solution with the task - 5 points.
  - 4. The effectiveness of the solution within the given task. The quality of the search results will be assessed (when evaluating the search results, several phrases/words will be entered into the search results. At the same time, the relevance of the search results from the top 10 videos for each phrase is determined. Each video from the search results is given quality points: 0 - not at all similar, 1 - not similar, 2 - similar, 3 - very similar. Thus, for each phrase you can get from 0 to 30 points. The total score will be given as the sum of points for all phrases:  $0 + 12 + 18 + 30 + 20 = 80$  points. Final score: 80 points. Further points will be converted into points according to the places taken by each team) - 10 points;
  - 5. The team's performance at the pitch session (persuasiveness, informativeness, concise and reasoned answers, compliance with regulations) - 10 points;