IR: Personal Recommendation based on YouTube comments

Tim Apers, Morgan Logghe & Sien Nuyens

October 2022

Admin
Tim Apers: s0183700
Morgan Logghe: s0170922
Sien Nuyens: s0181397

Scope
For this project, we would like to make personal recommendations based on YouTube comments. More specifically, given a piece of text (comment, mail, book snippet,...), we would like to create a model that can output video recommendations. These videos will fall under one of 9 Lifestyle topics, namely Fashion, Fitness, Food, Hobby, Pets, Beauty, Technology, Tourism, Vehicles.

To do this, we will first have to scrape a large amount of comments from videos within the specified categories. The YouTube Data API allows us to execute 10,000 requests per day where each request can contain a maximum of 100 comments. This means we can fetch a total of 1,000,000 comments a day which will be more than sufficient for this assignment (we can generate more over time if needed). Furthermore, we can fetch corresponding videos for each of the 9 categories as they contain a topicId that is attached to those categories. Lastly, we will limit ourselves to English comments that are long enough to contain valuable information.

Next up, we will have to transform this textual information into a vector representation that we can later use for our model. This is where Information Retrieval comes in. We will look at approaches we have seen in class, for example TF-IDF, but also try out other methods outside of the course scope, like word2vec, in order to select the most fitting embedding for our data. Lastly, the data will be normalised, if not already, to better train our model.

Once we have a sufficient vector representation of our comments we plan to train a deep neural network on the data. This network will have a fixed number of inputs that will depend on our input data and will have 9 outputs, namely the 9 categories. Intuitively, the output of a comment will be a vector of size 9

where the corresponding category is set the 1 and the rest to 0. With a softmax function to determine probabilities at the end and fine-tuning the number of layers, neurons and hyperparameters we expect this model to achieve interesting results.

Finally, we still need to recommend videos from the best matching topic(s). For this task, we will score the input text against all comments under each of the videos in the selected topic(s). This will result in a ranking of the most interesting videos for that person and will be our recommendation.

Resources
YouTube Data API
Python:
- PyTorch
- Requests
- pandas
- scikit-learn

Communication Platform:
- Discord
- GitHub

Literature
YouTube Data API documentation
Research papers on text embedding, neural networks and the course slides.
Manning, C. D., Raghavan, P. & Schütze, H. (2009). Introduction to Information Retrieval. Cambridge University Press.

Evaluation
The evaluation will be split in two parts that both use ranked evaluation methods. It is important to note that we will only evaluate using YouTube comments, the rest of the input texts are ignored. First of all, we would split the data in 3 subsets: a training, validation and test set. As explained in multiple courses, we will use the validation set to track the loss of our neural network and evaluate under- or overfitting. Then, we can use the test set to determine Precision, Recall, setup a PR-curve, etc. of our overall model. These values are computed by checking if the predicted topic is the same as the topic from the original video that comment was posted on and if the original video is listed in the top recommendations.