Classification of YouTube Videos

Department Of Computer Science and Engineering
(Supervisor: Dr Deepti Gupta)

TEAM

Name of Member	Roll Number	Role
Akash Kandpal	1513310027	Backend Part
Sivasish Koch	1513310214	Javascript Scripts
Saif Ali	1513321165	Classification checker(manually)
Rahul Bilra	1513310161	Scrapers

INTRODUCTION

- The total number of people who use YouTube - 1,300,000,000.
- 300 hours of video are uploaded to YouTube every minute.
- Almost 5bn videos are watched on YouTube every single day.
- YouTube gets over 30 mn visitors per day.

Would it not be better if we(mostly students) could get those videos tagged which could help us in avoiding un-educative videos?

LITERATURE SURVEY

- Software present currently are limited in scope and they don’t deal directly with classification of YouTube videos.
- Currently the systems mainly focuses on building recommendation systems for Youtube and likewise video sites.
- Youtube itself is working on classification of videos automatically.

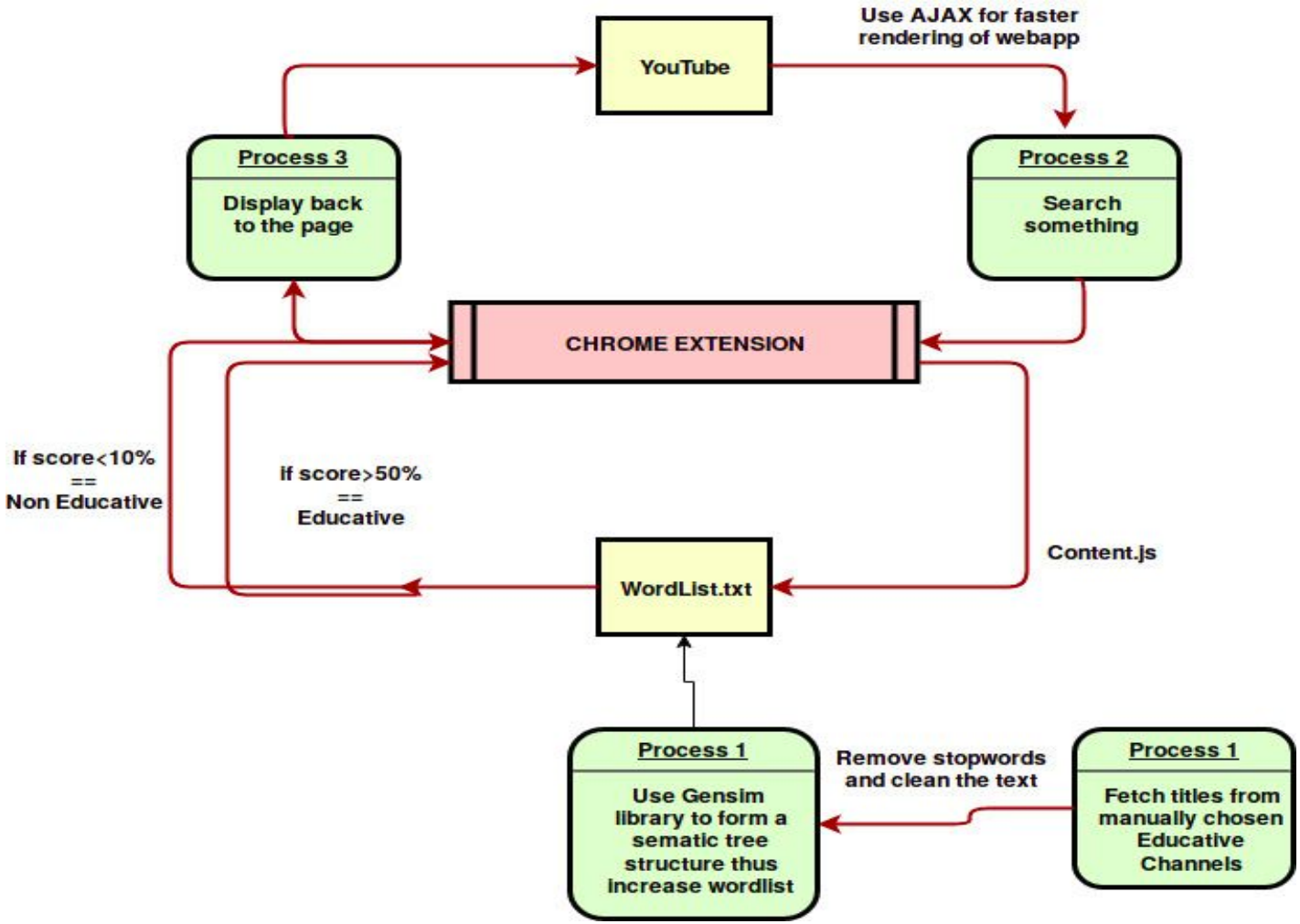
PROBLEM STATEMENT

Indian Youth users on YouTube gets distracted by much of the non-educative content on YouTube and watching those videos results in wastage of our time. This reduces our productivity and also makes us mentally tired so to stop this we require some strategy to tag those videos as non-educative ones.

PROPOSED METHODOLOGY

- We started with the generation of wordlist.txt file which contains the educative words from manually selected YouTube channels. As, there was a limit on the per day usage of YouTube API hits to 10k, so we created a scraper using Python and utilised it to get titles from these videos.
- Further we removed stop words and cleaned the text using NLTK library. Then we used gensim for increasing the coverage of those words by forming tree structure and this tree structure covered many synonyms of already present words.
- Further, the content.js contains the actual script which takes request onload() function and send back the request back to the caller.

FLOWCHART


YouTube
Use AJAX for faster rendering of webapp
Process 3
Display back to the page
Process 2
Search something
CHROME EXTENSION
If score<10% == Non Educative
If score>50% == Educative
Content.js
WordList.txt
Process 1
Use Gensim library to form a sematic tree structure thus increase wordlist
Remove stopwords and clean the text
Process 1
Fetch titles from manually chosen Educative Channels

PROPOSED METHODOLOGY

- Here, AJAX script is used for the faster rendering (without page reloading) of YouTube.
- Also, simple algorithm is used which matches the words in the title after removing the stopwords with the already generated wordlist.txt.
- At last it renders the calculated value back to the server and this displays whether a particular video is educative or not.
- This was kept static so as to prevent as it was not possible to automatically clean titles and save them again and again, earlier we were using a ML model for the same purpose but to mend the purpose of this project we made it static and this works good with 90%+ (checked manually) accuracy.

FEASIBILITY STUDY

The project utilises a simple chrome extension which means that no extra hardware required. Also, there is no scalability issue as it can be downloaded from the chrome store and can be installed easily.

The product is made for free use, not for monetary gains so it’s freely available on our github wherein it can be downloaded in zipped format and extracted , here is the installation video. It is not uploaded on Chrome Web Store.

MODULES

Module name	Progress (in %)
Clean the comments (removing stop-words) and manual cleaning (Backend)	100
Script for Fetch the titles and description and match with the list of words txt file. (Backend)	100
Generate the score for education videos using separate script. (Backend)	100
AJAX script for rendering server calls(Front-End)	100
Wordlist generation	100
Tagger for video on the basis of score(Back-end)	100
Send the score back to the browser.(Front-End)	100
Get titles and description of videos with more educative content and save their data in list of words (Backend)	100

HARDWARE REQUIREMENTS

Developer End Requirements

- OS: Any Linux variant
- Hard Drive: 500GB
- RAM: 2 GB

User End Requirements

- OS:Window 7
- Hard Drive: 100GB
- RAM: 256 MB

SOFTWARE REQUIREMENTS

- Gensim : https://radimrehurek.com/gensim/
- NLTK : https://www.nltk.org/
- Chrome Extension
- Vanilla JavaScript
- Plotly : For visualising graphs.
- XML Parsing
- YouTube API

LIVE DEMO

Installation Video

Demo Video

CONCLUSION

The project’s aim was to make it easier for tagging the videos for educative content and thus enhancing the user-experience of the Youtube users which is working with a reasonable accuracy.

So, this is a small effort from our side to improve the quality of videos and tag videos on Youtube by providing them scores, lot of advancements will be seen in this area in the near future.

REFERENCES

1. Bentivogli, L., Forner, P., Magnini, B., Pianta, E.: Revising WordNet Domains Hierarchy: Semantics, Coverage, and Balancing. In Proceedings of COLING Workshop on Multilingual Linguistic Resources, Geneva Switzerland (2004) 101-108

2. Yi, H.R., Deepu, R., Chia, L.T.: Semantic Video Indexing and Summarization Using Subtitles. Lecture Notes on Computer Science, Vol. 3331. Springer-Verlag Berlin Heidelberg New York (2004) 634-641

3. Declerck, T., Kuper, J., Saggion, H., Samiotou, A., Wittenburg, P., Contreras, J.: Contribution of NLP to the Content Indexing of Multimedia Documents. Lecture Notes on Computer Science, Vol. 3115. Springer -Verlag Berlin Heidelberg New York (2004) 610- 618

REFERENCES

4. Reidsma, D., Kuper, J., Declerck, T., Saggion, H., Cunningham, H.: Cross document annotation for multimedia retrieval. In EACL Workshop Language Technology and the Semantic Web (NLPXML) Budapest (2003)

5. Hangzai Luo, Jianping Fan, Jing Xiao, Xingquan Zhu, Semantic principal video shot classification via mixture Gaussian. In Proc. of IEEE International Conference on Multimedia & Expo. Vol.1, Baltimore, MD. (2003)

6. Suresh, V., Mohan, K.C., Swamy, K.R., Yegnanarayana, B.: Content-based Video Classification Using Support Vector Machines. In ICONIP-04, Calcutta, India (2004) 726- 731

THANK YOU