Classification of Video Viewing Task Types and Recommendation of Videos

Tatsuro Ide[1][a] and Hiroshi Hosobe[2][b]
[1]*Graduate School of Computer and Information Sciences, Hosei University, Tokyo, Japan*
[2]*Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan*

Keywords: Adaptive Information Retrieval, Machine Learning, Video Recommendation.

Abstract: YouTube is one of the largest and most sophisticated recommendation systems and a useful source of information for users. In video search on YouTube, even the same user may have different purposes in mind depending on the user's state. However, videos are recommended based on the relevance of videos and the user's viewing history, regardless of the user's state. This paper proposes a classification of video viewing task types based on the user's behavioral characteristics. By classifying the user's purpose as a task type, it enables higher-order recommendation that fits the task type. Behavioral characteristics are momentary characteristics of the user that appear from actions such as screen scrolling. The system implicitly records the user's actions, classifies the task type based on these parameters, and recommends the related video list on a mobile application that imitates YouTube. We conducted experiments to evaluate classification of task types and recommendation of videos.

1 INTRODUCTION

YouTube is one of the most popular video sharing platforms and a useful source of information for users. In video search on YouTube, even the same user may have different purposes in mind depending on the user's current state. However, at present, it is presumed that videos are recommended based on the degree of relevance of videos and the past viewing history, regardless of the user's current state. For example, even when searching for content for a learning purpose, videos of subscribed channels or content that ignores the current purpose may be recommended. Also, even when exploring a wide range of videos, the user might be unable to make new discoveries by returning to the video group that the user habitually watches. Displaying related videos in this way makes it possible that the user may be trapped in a closed search space.

Previous research on personally adaptive information retrieval has been actively conducted on document retrieval systems (Athukorala et al., 2016). On the other hand, there is almost no research on information retrieval on video sharing systems such as YouTube. This may be due to the difficulty of extracting features from videos, the large number of video groups, or the difficulty of acquiring video and user data. A study on YouTube video search by Google (Covington et al., 2016) showed that they analyzed such a huge amount of personal information by various measures and applied it to recommendation of videos.

In this paper, we propose a classification of video viewing task types and a method for recommending videos appropriate for task types that are analyzed according to the user's behavioral characteristics of viewing videos. We base the classification and the method on a previous study on adaptive information retrieval for a document retrieval system (Athukorala et al., 2016). Our aim is to make higher-order recommendations of videos by classifying the user's dynamic purpose from the user's behaviors. Our method uses behavioral characteristics that are the instantaneous features of the user appearing from detailed actions such as scrolling the screen. We constructed a classifier that determines task types based on the parameters obtained from such behavioral characteristics. The classifier is a decision

[a] https://orcid.org/0000-0001-5787-0443
[b] https://orcid.org/0000-0002-7975-052X

Ide, T. and Hosobe, H.
Classification of Video Viewing Task Types and Recommendation of Videos.
DOI: 10.5220/0010824700003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 373-380
ISBN: 978-989-758-547-0; ISSN: 2184-433X

tree that was obtained by imposing video viewing tasks on users.

We implemented a classification and recommendation system as a mobile application that imitated YouTube. The system records the user's actions, classifies task types, and recommends related videos. It implements video search by using the YouTube Data application programming interface (API), and realizes recommendation by filtering the search for related videos and adding the separately searched videos to the related video list.

To evaluate the task type classification and the video recommendation, we conducted an experiment by using this system. To reproduce the participants' usual use of YouTube in the proposed method, we also conducted a pre-questionnaire about their interests in subscribed channels and video categories. The result of the experiment indicated that the classification accuracy of the task types was 60%, which was higher than 1/3. However, regarding the video recommendation, we found limitations due to incorrect task types and unsuitable recommended videos.

2 RELATED WORK

2.1 Classification of Task Types According to Behavioral Characteristics

Search activities can be divided into two major categories: lookup and exploration (Marchionini et al., 2006). In lookup search, in order for the user to reach the correct area of the information space, the user first accurately expresses the information that the user has, quickly refers to the related result, and finally arrives at the most suitable item. On the other hand, in exploratory search, user behavior is dynamic. The user starts the search with an unclear search purpose in mind and initially issues an inaccurate search query. In addition, the user reads the search results and repeatedly reformulates queries according to the newly found keywords.

Athukorala et al. (2015) showed that these lookup and exploratory searches could be categorized by easily measurable behavioral characteristics. Query length, scroll depth, reading time, task completion time, and cumulative numbers of clicks were shown as effective behavioral characteristics for classification. Based on this research, they further constructed a classifier that recorded implicit search behaviors on a paper search engine such as Google Scholar and classified exploration tasks and research tasks according to their parameters (Athukorala et al., 2016). This classifier was implemented in the article search system, and the determined tasks were used for recommendation. In this system, query length, reading time, and cumulative numbers of clicks were used as behavioral characteristics.

2.2 YouTube Video Recommendation

Google has adopted various measures for YouTube video recommendation. Among them, in the study on the application of deep learning to a recommendation system (Covington et al., 2016) and the research on efficient input of context information (Beutel et al., 2018), implicit features were introduced and used to construct recommendation systems. Both studies used implicit features such as viewing histories and user genders, but they did not use detailed and instantaneous user actions such as scroll depth.

The former study showed that the conventional matrix factorization-based recommendation algorithm was replaced with deep learning to improve accuracy. They proposed that solving difficult problems with large amounts of fresh content could be roughly divided into two stages: (1) narrow down candidates from millions of videos; (2) rank the videos according to their scores. In stage (1), viewing histories, search results, user genders, viewing areas, training sample ages, etc. were inputted as feature quantities. The age of the training sample is the time elapsed since the video was uploaded, and it was observed that fresh content tended to be viewed more frequently regardless of taste. In stage (2), the embedded vectors of videos and the numbers of recommendations were used for scoring. The numbers of recommendations were used for learning to lower the scores of unselected videos even if they were displayed multiple times.

The latter study attempted to solve the problem that the model size became large when the embedded vectors were connected to the user's context information. The used information was the elapsed time before and after viewing, the device to be viewed, and page information. Page information was a feature of each page such as the top page and the video playback page, and there was a tendency for new content to be viewed on the top page.

3 DECISION TREE GENERATION

In this paper, C4.5 (Quinlan et al., 1993), which is a classification learning algorithm among the machine learning algorithms of Weka (Witten et al., 2006), is used to generate a decision tree. C4.5 is an algorithm based on the divide-and-conquer method. Weka takes a dataset in a special format called ARFF format as input and outputs the result by the selected classification learning algorithm and evaluation method. Each data in the dataset is a set of the input variable that is the branching condition of the decision tree and the possible output of the leaf node that has no children. We use the data obtained by imposing a task on users as an input dataset, and describe the query length, reading time, scroll depth as input variables, and the task type as possible output. The process of decision tree generation is shown below.

3.1 Quantification of Conditions based on Entropy

Based on information theory, C4.5 uses entropy to quantify the discriminating power of the leaves of the decision tree. In the set C of the dataset, the possible outputs belong to the set D, and the probability at which $x \in D$ occurs is expressed as $p_x(C)$. The entropy $M(C)$ for the set C of the dataset is as follows:

$$M(C) = -\sum_{x \in D} p_x(C) \log p_x(C)$$

When the number of classes that divide the base of the logarithm (possible output x) is set, the maximum of $M(C)$ becomes 1. When it is close to 1, the dataset is in a messy state.

3.2 Selection of Conditions

The information gain obtained by dividing C into k pieces with the input variable as a condition is $G(C)$:

$$G(C) = M(C) - \sum_{i=1}^{k} \frac{|C_k|}{|C|} \times M(C_k)$$

The information gain can be interpreted as the degree to which the disorder is reduced depending on the conditions. The quality of division can be defined by this information gain. The dataset is divided under each condition, and the one with the large information gain is set in the leaf node. This is done recursively in each subtree of the child to generate the decision tree.

4 PROPOSED METHOD

In this paper, the user's dynamic purpose which can be read from the user's behavioral characteristics is classified as a task type and applied to recommendation. The proposed method mainly consists of three components: a user interface, a classifier, and a recommender:

1. The user interface records the user's actions when viewing videos, which is realized as a YouTube client application. This is almost the same as that of YouTube Mobile, but it implicitly records the actions.
2. The classifier obtains the parameters that the user interface extracted from the actions. Then it determines the task type.
3. The recommender filters the video search by the task type and also adds the separately searched video to the list.

4.1 Definition of Task Types and Behavioral Characteristics

The task type is defined based on the paper (Athukorala et al., 2016). They defined two task types, “lookup” and “exploration”, for article search. In this paper, we define the following three task types for video search by newly adding “repeat”.

- Lookup: A task where the video to be searched for is decided in advance; the user searches for a specific video as a target (White et al., 2006).
- Exploration: A task where the video to be searched for is not decided; the user searches a wide range of content based on their interests.
- Repeat: A task where the video to be search for is habitually checked by the user.

Behavioral characteristics are behaviors that represent instantaneous user characteristics. In this paper, the following three behavioral characteristics were recorded and used as parameters.

- Query length: The number of words entered in the query in the first search session; count by separating them with spaces (Jansen et al., 2001).
- Scroll depth: Depth of scrolling up and down the view of the video list.
- Reading time: Time to start watching the first video.

4.2 Classifier Parameter Determination

In this paper, Weka is used to select the parameters of behavioral characteristics. The data obtained by imposing a task on the participants on this application is used as a dataset. Table **1** shows the assigned tasks and each recorded parameter. The tasks assigned to the participants are 4 lookup tasks, 8 exploration tasks, and 8 repeat tasks for a total of 20 tasks. In the lookup task, participants searched for the video that they watched two hours before. In the exploration task, participants searched videos in the category of interest that they answered in advance. In the repeat task, we used a group of videos that participants habitually checked.

Table 1: Participant task completion data used in the dataset.

Participants	Query length	Reading time	Scroll depth	Task
1	2	35	0	Lookup
	2	113	30	Exploration
	2	102	71	Exploration
	1	185	30	Repeat
	1	58	2	Repeat
2	3	97	51	Lookup
	2	187	74	Exploration
	3	97	38	Exploration
	1	181	24	Repeat
	2	46	51	Repeat
3	2	25	0	Lookup
	1	256	162	Exploration
	1	116	71	Exploration
	0	389	85	Repeat
	0	427	238	Repeat
4	2	35	24	Lookup
	1	112	133	Exploration
	2	262	67	Exploration
	0	96	48	Repeat
	0	22	37	Repeat

Using this dataset as input data, J48, which generates a decision tree based on C4.5 (Quinlan et al., 1993), was selected as the machine learning algorithm. The reason for adopting this algorithm was that it was easy to understand the cause of classification failure from the excellent visibility of the decision tree. Figure 1 shows the decision tree generated using cross-validation for the training data. Leaf nodes with children represent query length, reading time, and scroll depth respectively. Leaf nodes that have no children represent the three tasks, i.e., lookup, expansion, and repeat. The branch comparison operation branches according to the parameters of the parent node of each data.

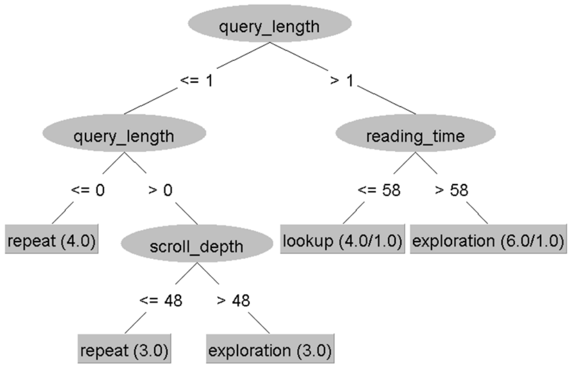

Figure 1: Task type decision tree based on user data.

The query length was selected as the first condition in the generation of this decision tree because its information gain was larger than those of the scroll depth and the reading time. The information gains of the datasets in Table 1 are calculated as follows. Since the data is divided into three categories, lookup, exploration, and repeat, and also since the numbers of data in their classes were 5, 10, and 10 respectively, the entropy is the following:

$$M(C) = \frac{5}{25}\log_3\frac{25}{5} + \frac{10}{25}\log_3\frac{25}{10} + \frac{10}{25}\log_3\frac{25}{10} \cong 0.960$$

The information gain obtained by dividing the query length on the condition that it is larger than 0 is the following:

$$G(C) = 0.960 - \left\{\frac{10}{20}\left(\frac{3}{10}\log_3\frac{10}{3} + \frac{7}{10}\log_3\frac{10}{7}\right) + \frac{10}{20}\left(\frac{4}{10}\log_3\frac{10}{4} + \frac{5}{10}\log_3\frac{10}{5} + \frac{1}{10}\log_3\frac{10}{1}\right)\right\} \cong 0.253$$

4.3 Recommender

The search parameters are changed according to the task type determined by the classifier. If the task type is classified as exploration, videos of the category of interest are added to the related videos. If the task type is classified as lookup, new videos and live streaming of subscribed channels are displayed in descending order of the relevance of the videos. If the task type is classified as repeat, the new video of the subscribed channel of interest is added to the related video display regardless of the relevance of the video.

5 IMPLEMENTATION

We implemented our system as a mobile application by using Android Studio and Google Pixel 4a. We obtained YouTube video data by using the YouTube Data API and played them back by using the Android Player API. This application records the user's behavioral characteristics while the user is viewing videos on YouTube, classifies task types, and applies them to recommendations. The query length, reading time, and scroll depth are recorded as behavioral characteristics.

The user interface imitates YouTube Mobile. Figure 2 shows screenshots of the top page and the video playback page. The valid bottom tabs are Home, Search, and Subscribed Channels, and a pre-questionnaire searches the video list for each participants. The user presses the magnifying glass icon at the top of the screen to start the search action. If the user taps a video from each tab or the video list of the search results, the video will be played. On the video playback page, a list of related videos is displayed below the video player. The user plays the first video by searching for a video or selecting a video from the video list on each tab. After that, the user can search for a video by selecting a video from the related video display or swiping to return to the previous screen. The user presses the account icon to the right of the magnifying glass icon at the top of the screen to go to the login page. On this page, it provides OAuth authentication to use YouTube user data associated with the Google account.

Figure 2: Implemented application.

6 EXPERIMENT

We conducted an experiment to evaluate the dynamic classification of task types and recommendation of videos according to behavioral characteristics. Our system uses a classifier that records data for three user interactions (query length, scroll depth, and reading time). In addition, the classifier treats these input data as parameters and predicts whether the user's task is lookup, exploration, or repeat. In this section, the system that enables this classifier is called the full system; the system that disables this classifier is called the baseline system and is used for comparison. There were 5 participants, the average age was 22 years old, and they habitually used YouTube. A pre-questionnaire was conducted to simulate the video list on each tab of YouTube.

6.1 Design

Each participant was asked to complete a total of 5 tasks, each of which consisted of 1 lookup task, 2 exploration tasks, and 2 repeat tasks, for the full system and the baseline system. Also, the task order was balanced to avoid the order effect.

The lookup task asked participants to search for the video that they watched 2 hours before the experiment. By this method, we ensured that the participants would remember information about the video that they watched, but forget detailed information such as the video title and channel name (Kane et al., 2000). For the exploration task, we asked them to answer the categories of videos in which they were interested but about which they did not know well (Schraefel et al., 2005), and then to select the appropriate kind of videos and investigate them freely. This attempted to reproduce learning, which is a typical exploratory search (White et al., 2006). Table 2 shows the correspondence between the categories selected by the participants in the exploration task. In the repeat task, the videos that the participants habitually watched were answered in advance and used as usual. After the two exploration tasks and the two repeat tasks were completed, we asked a question about the related video display. This allowed the full system and the baseline system to change the orders of related videos.

Table 2: Video categories selected by the participants in the pre-questionnaire.

Participant	Video category
1	Music
2	Science & Technology
3	News & Politics
4	Sports
5	Music

6.2 Procedure

The procedure of the experiment was divided into three stages: a pre-questionnaire, watching a video for the lookup task, and the main experiment. In the pre-questionnaire, we prepared questions to reproduce the individual YouTube pages. We asked them to select a frequently viewed category from the YouTube categories for the content displayed on the home tab, and to answer a list of frequently viewed subscribed channels for the subscribed channel tab. In addition, for the categories to be searched by the exploration task, we asked them to answer multiple categories in which they were interested but about which they did not know well, and to select the one that could obtain valid results from the API.

To reproduce the situation of the lookup task, we asked the participants to watch a specific video individually 2 hours before the experiment. To prevent the content from being memorized in detail, we did not mention the lookup task to be done later. This video was about 5 minutes long, and when the participants finished watching it, we asked them to return to their respective tasks, and told them that they would perform the experiment 2 hours later.

After 2 hours and before this experiment, we explained the tasks and functions of the application. Each participant was asked to perform a total of 5 tasks. The lookup task was limited to the maximum of 15 minutes, and the participant could finish the task when the video was found. The participants spent 20 minutes each on the exploration and the repeat task. Each of these experiments took about 90 minutes.

6.3 Accuracy of the Classifier

Table 3 shows the task types judged by the full system. Since each participant had 5 tasks and 5 people worked on it, the accuracy was calculated for a total of 25 tasks. Among the 25 tasks, 15 were classified correctly, and the overall accuracy was 60%. Among these, the lookup task were 5 tasks, and 3 tasks, i.e., 60% of the tasks, were classified correctly. The accuracy of the exploration task was 50% because 5 out of 10 tasks were classified correctly. The accuracy of the repeat task was 70% because 7 of the 10 tasks were classified correctly.

6.4 Evaluation of the Recommender

As a result of the post-questionnaire, 20% of the participants answered that the full system (i.e., with recommendation) was suitable, 50% of the participants answered that the baseline system (i.e., without recommendation) was suitable, and 30% of the participants answered that they did not notice any difference between the two systems. There were positive evaluations such as the related video display that caught the eye in the exploration task (participant 3). However, many of the participants answered that they did not notice the difference in the related video display throughout the task. Some people said that they noticed that the related video display had a video display that was completely different from the intended one (participants 1, 2, and 5). This is because the related video display included a video that had nothing to do with the purpose because the judged task was different from the original task. Other participants answered that the same video was displayed repeatedly (participants 1 and 3). This is because the video was added to the related video display as a recommendation.

Table 3: Participant behavioral characteristics and the task types judged by the full system.

Participant	Query length	Reading time	Scroll depth	Judged task type	Correctness
1	2	34	1	Lookup	Correct
	2	62	63	Exploration	Correct
	3	70	37	Exploration	Correct
	1	67	94	Exploration	Incorrect
	2	417	92	Exploration	Incorrect
2	2	50	4	Lookup	Correct
	1	112	60	Exploration	Correct
	2	64	13	Exploration	Correct
	0	23	31	Repeat	Correct
	0	62	7	Repeat	Correct
3	1	25	16	Repeat	Incorrect
	1	49	4	Repeat	Incorrect
	1	73	108	Exploration	Correct
	1	22	8	Repeat	Correct
	2	38	54	Lookup	Incorrect
4	1	14	5	Repeat	Incorrect
	1	86	30	Repeat	Incorrect
	1	147	35	Repeat	Incorrect
	0	19	4	Repeat	Correct
	1	142	19	Repeat	Correct
5	2	32	3	Lookup	Correct
	1	115	6	Repeat	Incorrect
	1	119	48	Repeat	Incorrect
	1	146	12	Repeat	Correct
	1	120	1	Repeat	Correct

7 DISSCUSSION

Our experiments showed that the task types were classified with a certain degree of accuracy. On the other hand, we found two problems from the participants' evaluations. The first problem consisted of two cases: (1) a recommendation was obtained from an incorrect task type; (2) an inappropriate recommendation was obtained from a correct task type. An example of case (1) is that a subscribed channel video was added to the related video display because it was classified as a repeat task during the lookup task. In case (2), even if the task type was correctly classified, the video added by recommendation did not satisfy the user's intention. This problem was related to the recommendation evaluation method. Although qualitative evaluations could be obtained through questionnaires and interviews, quantitative evaluations of recommended videos could not be performed; this is because we could not implement a system that would lead to quantitative evaluation by, e.g., analyzing videos in the viewing history.

To evaluate the recommendation result, a quantitative evaluation of recommended videos should be performed. At present, we simply add videos that have been categorically searched, and add videos from subscribed channels. It is necessary to consider what kind of recommendation is preferable in consideration of the results obtained by the quantitative evaluation.

Since the task type is determined in the first search session and the recommendation is continued based on the task type after that, inappropriate recommendation is continuously made in the case of an incorrect classification. It is necessary to consider a system that redetermines the task after watching the video several times. Then, even if the user's purpose in mind changes, it will be possible to continue to adapt to it dynamically by periodically redetermining the task type.

In recent years, research on neural networks for recommendation systems has progressed. YouTube also incorporates the context of user information such as video viewing histories and search histories into a neural network and uses it for recommendation (Covington et al., 2016). In this paper, instead of such a large amount of data, we focused on the user's behavior, which may be based on the user's purpose. At this time, an important problem is how the user's purpose and the user's behavior are linked in the video search. The high readability of the decision tree makes it easier for us to understand this problem, which will be difficult when a neural network is used instead.

Bhabad et al. (2017) realized the recommendation of video related information by ASR and OCR. The method recommended web links, image links, and YouTube links based on the text data from images and sounds cut out from the video. Based on the task type of this paper, Bhabad's method worked effectively when the purpose was clearly defined such as lookup tasks. On the other hand, it was not suitable when the users wanted to search a wide range of videos such as exploration tasks and repeat tasks. Silva et al. (2017) showed that comments on videos could devide into technical, or instructional videos, and non-technical videos. Although this is similar to our research background, the point of view is defferent. Since Silva's method focuses on the video itself, there are problems with videos increasing every day and videos with only few comments. By contrast, since our method focuses on the user's behavior, it is possible to avoid problems caused by the video itself.

C4.5, which was used in the decision tree generation algorithm, has the problem that the decision tree cannot be updated sequentially. Especially when the dataset and the user's behavior are extremely different as in the experiment in this study, an inappropriate decision is made. Supervised learning such as C4.5 requires input and correct answer data and given tasks, and such data cannot be analyzed sequentially by a decision tree generation algorithm. It will be necessary to consider measures that can be updated sequentially, such as implementing the decision tree generation algorithm itself in the application.

We adopted query length, scroll depth, and reading time as behavioral characteristics because they were general behaviors in search systems. In addition to these, YouTube has other characteristic operations such as video preview and maximization and minimization of the video player. It is necessary to verify whether these actions and other actions that are effective in document retrieval are also effective in video search.

Although we attempted to reproduce the function of YouTube, some part of it imposed difficulty. For example, the list of soaring videos on the exploration tab and query search could be implemented with the current YouTube Data API. However, the list of recommended videos for a user displayed on the home tab could not be implemented because it was excluded from the current API. Also, even if the list of subscribed channels is obtained, the list of videos in order of posting date and time cannot be obtained. Therefore, we conducted a pre-questionnaire to simulate YouTube without using these unavailable functions.

8 CONCLUSIONS AND FUTURE WORK

We proposed a classification of video viewing task types and a method for recommending videos appropriate for the task types. To dynamically classify the task type of a user, the parameters of the behavioral characteristics were recorded and analyzed by a decision tree. We implemented an application that implicitly recorded query length, scroll depth, and reading time, determined the task type by using a decision tree, and reflected it in the related video display. We conducted an experiment to evaluate the classification accuracy and the recommendation of videos. Improvement of the evaluation method such as implicitly evaluating the search result list with bookmarks as in the paper (Athukorala et al., 2016) is a future task.

REFERENCES

K. Athukorala, A. Medlar, A. Oulasvirta, G. Jacucci and D. Glowacka. (2016). Beyond Relevance: Adapting Exploration/Exploitation in Information Retrieval. In *Proc. ACM IUI*, pp. 359-369.

P. Covington, J. Adams and E. Sargin. (2016). Deep Neural Networks for YouTube Recommendations. In *Proc. RecSys*, pp. 191-198.

G. Marchionini. (2006). Exploratory search: From finding to understanding. In *Com. ACM*, vol. 49, no. 4, pp. 41-46.

K. Athukorala, D. Głowacka, A. Oulasvirta, J. Vreeken and G. Jacucci. (2015). Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks. *JASIST*, vol. 67, no. 11, pp. 2635-2651.

A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto and E. H. Chi. (2018). Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proc. WSDM*, pp. 46-54.

J. Quinlan. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

I. H. Witten, E. Frank. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

R. W. White, B. Kules and S. M. Drucker. (2006). Supporting exploratory search. *Com. ACM*, vol. 49, no. 4, pp. 36-39.

B. J. Jansen and U. Pooch. (2001). A review of web searching studies and a framework for future research. *JASIST*, vol. 52, no. 3, pp. 235-246.

M. J. Kane and R. W. Engle. (2000). Working-memory capacity, proactive interference, and divided attention: Limits on long-term memory retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 26, no. 2, pp. 336-358.

M. C. Schraefel, D. A. Smith, A. Owens, A. Russell, C. Harris and M. L. Wilson. (2005). The evolving mSpace platform: Leveraging the Semantic Web on the trail of the Memex. In *Proc. Hypertext*, pp. 174-183.

D. Bhabad, S. Therese, M. Gedam. (2017). Multimedia based Information Retrieval Approach based on ASR and OCR and Video Recommendation System. In *Proc. CTCEEC*, pp. 1168-1172.

H. Silva, I. Azevedo. (2017). Instructional Videos and Others on YouTube. In *Proc. CSEDU*, pp. 418-425.