Journal of Cases on Information Technology
Volume 24 • Issue 3 • July-September 2022

PerSummRe:
Gaze-Based Personalized Summary Recommendation Tool for Wikipedia

Neeru Dubey, Indian Institute of Technology, Ropar, India
Amit Arjun Verma, Indian Institute of Technology, Ropar, India
Simran Setia, Indian Institute of Technology, Ropar, India
S. R. S. Iyengar, Indian Institute of Technology, Ropar, India

ABSTRACT

The size of Wikipedia grows exponentially every year due to which users face the problem of information overload. The authors propose a remedy to this problem by developing a recommendation system for Wikipedia articles. The proposed technique automatically generates a personalized synopsis of the article that a user aims to read next. They develop a tool, called PerSummRe, which learns the reading preferences of a user through a vision-based analysis of his/her past reads. They use an ensemble non-invasive eye gaze tracking technique to analyze user reading patterns. This tool performs user profiling and generates a recommended personalized summary of yet unread Wikipedia articles for a user. Experimental results showcase the efficiency of the recommendation technique.

KEYWORDS

Computer Vision, Eye Gaze Tracking, Image Processing, Machine Learning, Natural Language Processing, Personalized Summary, Recommendation System, Summary Recommendation, Wikipedia

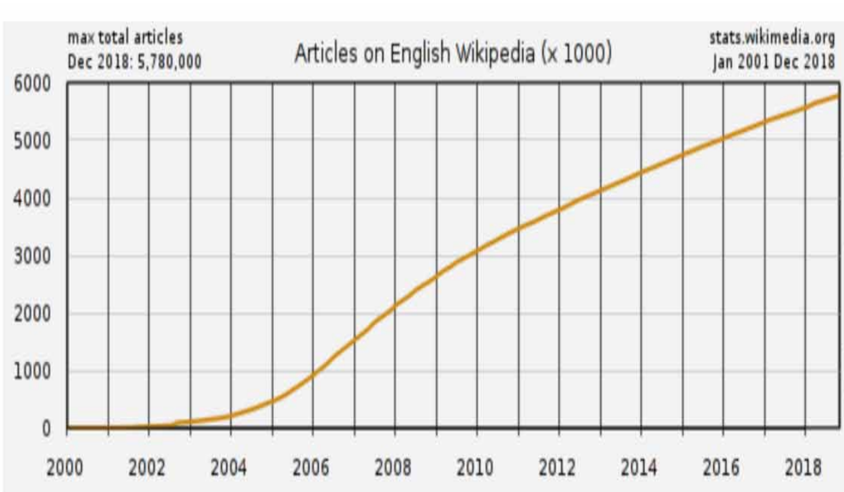
INTRODUCTION

The current human population is blessed with the ease of information availability on digital platforms. The same blessing turns into a bane of information overwhelm when a user does not know what to search for. A study performed by Lyman et al. (2000) showed that the world's annual output of web content is roughly 1.5 million terabytes. Similar observations can be made on Wikipedia. Wikipedia is the world's largest crowd-sourced encyclopedia. Since its inception, there has been an exponential increase in the number of articles present on Wikipedia (Oecd & OECD, 2010). Users can currently access Wikipedia articles through one of the two methods: First, by browsing on either a subject index or a title index sorted alphabetically, and second, by following hyperlinks embedded within article pages, called WikiLinks (Lüer & Cummins, 2009). These access methods are static since they are subjected to manual editing.

English Wikipedia alone contains more than 6 million articles, over 3.6 billion words. It has as many words as the 120-volume English-language *Encyclopædia Britannica (online),* and more words than the enormous 119-volume Spanish-language *Enciclopedia universal ilustrada europeo-americana* . Figure 1 shows the rapid increase in the size of English Wikipedia. Unfortunately, not

DOI: 10.4018/JCIT.20220701.oa7

Figure 1. Plot demonstrating expeditious increase in the number of articles in English Wikipedia (Source: PerSummRe_images\wiki_size300)

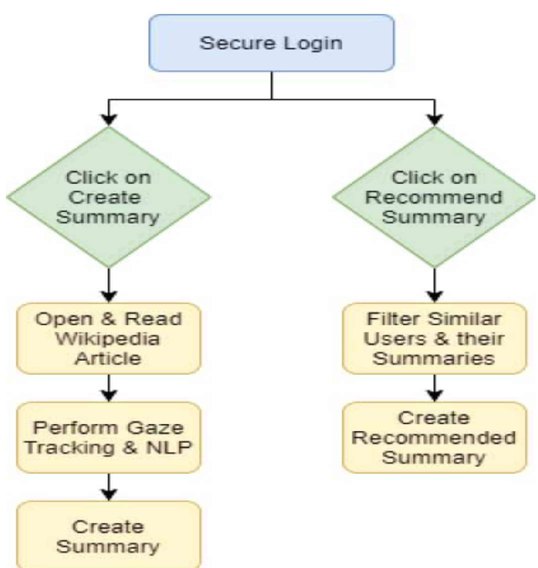


all Wikipedia articles are of interest to the user. Presenting the user with a synopsis of the interesting articles can save the user by identifying which articles are most relevant to them.

This can either be a generic summary, which gives an overall sense of an article's content, or a personalized summary, which presents the filtered content as per the user's interest. Several researchers have investigated various approaches to create personalized document summaries (Park, 2008; Y. Liu et al., 2008). These approaches require the sharing of personal information or some complex algorithm to create summaries.

Eye gaze tracking provides a new dimension to capture a user's ROI (Region Of Interest). But to create summaries using eye gaze, users are required to read the article during run-time. Personalized

Figure 2. Overall workflow of PreSummRe operations (Source: PerSummRe_images\workflow300)



recommendations are a vital method for information retrieval and content discovery in today's information rich environment. Recommendation, combined with summarization, can allow users to face a huge amount of information to navigate that information efficiently and satisfyingly. In this paper, the authors present a novel approach to recommend personalized summaries for Wikipedia users. The proposed method depends on the collaborative filtering-based recommendation. It extracts information from the past summaries created by the user to identify similar users in-terms of the reading pattern. Then it uses the summaries of these users on the mentioned article to create a recommended summary. Note that this is different from merely creating a new summary using the reader's eye gaze. Here, user behavior has been analyzed based on his/her past reads and then recommend the sentences from the new article, which are of the user's interest.

The authors develop a standalone cross-platform application, called PreSummRe, to recommend personalized summaries for users based on their reading patterns. This tool has an in-built option to create new gaze-based summaries using a publicly available tool, WikiGaze. The authors also present users with a list of recommended articles for future reads. The source code of the proposed recommendation system is available on GitHub (link to be provided post-acceptance). Figure 2 shows the overall workflow of the proposed tool for summary creation and summary recommendation.

RESEARCH CHALLENGES AND PROPOSED SOLUTIONS

In this section, some of the commonly faced challenges by the recommendation systems have been mentioned. The authors also briefly mention the proposed remedies to these challenges. The detailed remedies can be found in the methodology section:

- **Utilizing Gaze for Recommendation:** Eye gaze tracking is a non-trivial task. Several eye-gaze trackers are available today (Farnsworth, n.d.), but they are expensive and cause unpleasant user experience. To enable eye gaze tracking at users' sites, there exists a requirement of an economical and ubiquitous solution. Therefore, the authors use a simple web-camera along with pre-trained eye-tracking deep models and natural language processing to generate personalized summaries.
- **Cold Start Problem:** Cold start problem refers to the lack of data in a new system or when a new user joins the system. This problem mainly occurs in collaborative filtering algorithms. Here, the authors deal with various cases of cold start individually. The authors use pre-trained deep learning-based models to provide recommendations when the system does not have sufficient similar users to create recommended summaries.
- **Sparsity Problem:** The main reason behind data sparsity is that most users do not wish to share their data. The authors deal with this problem by enabling the recommendation option only when users agree to share their summaries. The algorithm also filters out users in a primary step to prepare a dense dataset for further processing.
- **Scalability Problem:** This problem emerges in the recommendation systems that cannot handle the explosive growth of data with time. The authors ensure that our system sustains increasing user demand by pre-processing the data, clustering the articles, and using user-based collaborative filtering. The details have been discussed in relevant sections.

BACKGROUND

The recommendation system is defined as the system that recommends an appropriate product or service after learning its tendency and desire. Today, many researchers are conducting a rigorous study on the recommendation system. The most crucial factor in the recommendation system is filtering and analyzing user preference correctly, thereby recommending the best product that the user wants to be based on an accurate estimation approach (Cho et al., 2002; Sarwar et al., 2000). This leads to research on the personalized recommendation method. Two of the well-known representative methods are content-based filtering (Mooney & Roy, 2000) and collaborative filtering (Goldberg et al., 1992).

Wikipedia has attracted the attention of researchers since its inception. Many attempts have been made to simplify the process of searching for information on this vast encyclopedia. Frankowski et al. (Frankowski et al., 2007) proposed a community-maintained, generic Wikipedia recommendation system. It served as a base to build new recommendation systems rapidly. One such work was SuggestBot (Cosley et al., 2007). It enabled contributors to easily find work in Wikipedia by recommending related articles based on the similarity of text, WikiLinks, and common editors. Apart from these, various features of Wikipedia have been used for recommendation purposes. (Szymański, 2010)) proposed a semantic schema to provide meaningful suggestions for editing the pages and improving search capabilities. Wikipedia's structure is also well explored to find document semantics, which in-turn help searching related articles. Musto et al., (2016) developed a content-based recommendation (CBRS) framework which uses textual features extracted from Wikipedia to learn user profiles based on such Word Embeddings. Piccardi et al, (2018) introduced a category-based approach for recommending sections of Wikipedia articles to editors. This can help editors to find appropriate sections to edit in an article. In the same line of research, Moskalenko et al. (2020) develop a scalable system on top of Graph Convolutional Networks and Doc2Vec, learning how to represent Wikipedia articles and deliver personalized recommendations for editors.

Apart from Wikipedia specific research, the general document recommendation system has been well explored in the past few years. Let us have a look at these techniques to understand more about some famous document recommendation systems. iScore (Pon et al., 2007) used several features of a document using a naive Bayesian classifier or a linear correlator. They were able to outperform the most popular filtering techniques. The personalized recommendation has especially attracted many researchers. In (Ahn et al., 2004), a personalized recommendation system was proposed based on the process of dimensionality reduction. They perform web mining to extract users related items. In (Liang et al., 2008), an Internet recommendation system was proposed that allowed customized content to be suggested based on the user's browsing profile. Habibi et al. (2015) proposed a keyword extraction and clustering for document recommendation in conversations. Xu et al. (2020) conducted user behavior study towards recommendation systems and analyzed that the recommendations help users access their documents significantly faster. Similar to our work, Radev et al. (2001) also presented a document summarization and document recommendation system (called WebInEssence) deployed on the Web. But as opposed to WebInEssence, our approach utilizes the eye gaze of the readers to provide personalized recommendations.

The current work also involves the creation of personalized multi-document summaries from the summaries of similar readers of the current user. They are decided based on the current user's past reads. Therefore, it is necessary to discuss some famous multi-document summarization techniques to cover this work's complete background. In (Liu Y. a., 2008), a personalized MDS technique was proposed in which the customized PageRank ranking process was performed based on personalized prior probability for each sentence in the corpus. (Liu, 2018) proposed to generate multi-document summaries of Wikipedia articles by summarizing each paragraph of each source article. Some studies also include a graph-based approach for MDS (Uckan, 2020).

The complexity of all the techniques mentioned above depends on the number and size of the input documents, and they do not take into account the reader's current cognitive state. As opposed to the literature mentioned above, this paper utilizes the collective effort model for Wikipedia based recommendation system to generate personalized summaries using eye-gaze information of users.

PERSUMMRE TOOL OVERVIEW

The PerSummRe is a cross-platform, standalone tool for Personalized Summary Recommendation for Wikipedia articles. This tool enables users to create gaze-based personalized summaries and then utilize these summaries to train a summary recommendation system. Each user is asked to make a secure profile. Upon login, they are given options to either create their summary or get a personalized recommendation. To enable the option of recommendation, users must create at least five summaries and share all the summaries. This restriction is imposed to gather enough data to learn about his/her reading pattern. Figure 3 demonstrates the two most significant screens of the tool. Let us discuss various features in detail:

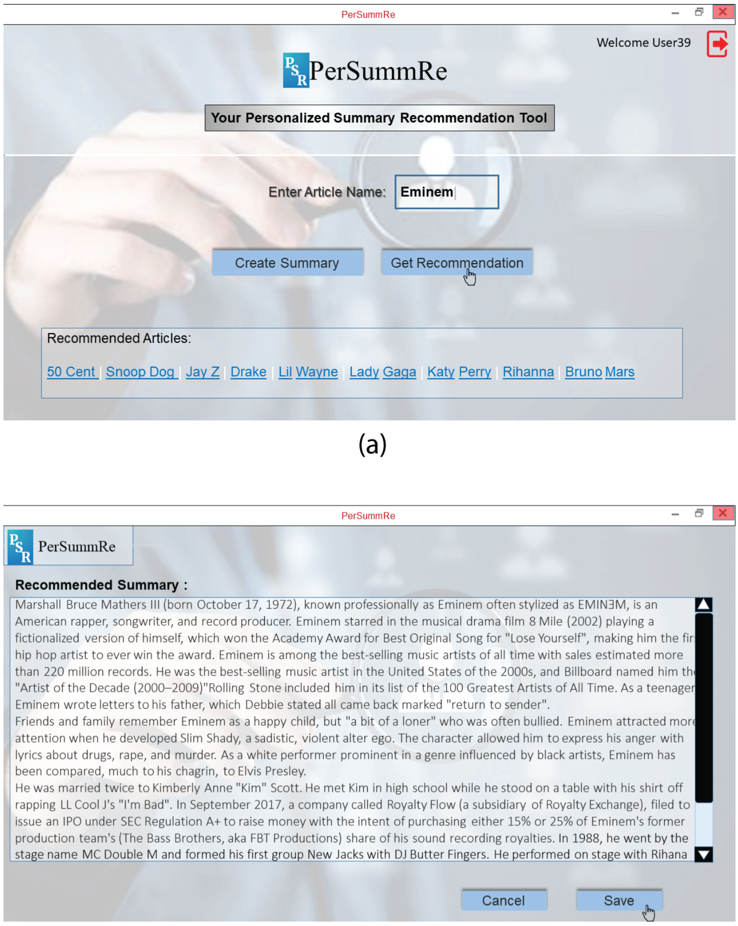
- **Secure Login:** Security threats have risen with the advancement of cloud technologies like a malicious insider attack, data loss, and privacy breach. Since all the users share their gaze-based summaries in the system, it is necessary to protect their login from any kind of malicious attack. Here the authors have designed a login system built with PHP PDO and Bootstrap. This simple, lightweight security system can protect the system from attacks like SQL injection, cross-site scripting, session hijacking, and brute force attacks.
- **Summary Creation:** Past studies (Ajanki et al., 2009) have revealed that our eye gaze pattern is closely related to our cognitive process. The current work utilizes the user's eye-gaze pattern to extract relevant portions of an article to create a summary. The summaries are being generated using a publicly available cross-platform application named WikiGaze. This tool was developed by us to provide a new dimension to articles. This application is embedded in the PreSummRe tool. Once the user clicks the option to create a new summary, he/she is redirected to the desired Wikipedia article. While the user reads the article, the tool performs eye gaze tracking to extract ROI and provide a summary to the user at the end of the reading session.
- **Summary Recommendation:** The authors learn a user's reading pattern in terms of eye gaze points density on articles (as shown in Figure 4) and past summaries that he/she has shared. With this information, the system analyzes the desired article's summaries shared by similar users (based on the similarity in the reading pattern) and creates a consolidated summary for the current user.

PROPOSED SUMMARY RECOMMENDATION METHOD

It has been observed that each Wikipedia reader follows an information search path on an article based on their professional and personal background (Singer et al., 2017). In Figure 4, the authors show the gaze points distribution of 2 different users on the same Wikipedia article frame. Building on this observation, a recommendation system has been proposed, which generates recommended summaries by capturing a reader's reading preferences. It is important to note here that the proposed technique is not creating gaze-based summaries because users are required to perform reading operations in-order to generate summaries. Instead, the authors propose to learn reading patterns of users from the past summaries (of different articles) and recommend summarized article content. To generate gaze-based summaries, the WikiGaze tool has been used. This tool captures users' reading patterns and generates personalized summaries by selecting sentences based on gaze points' density. The details regarding the working of the WikiGaze tool can be found on their website (https://wikigaze.github.io/). To generate recommendations, PerSummRe makes use of the similarity of reading patterns between different users. For instance, when geographers visit any countries' Wikipedia article, they tend to be more interested in the article's geography section.

As mentioned earlier, to enable the recommendation option, each user must create at least five summaries (using integrated WikiGaze in the PerSummRe tool) and share all the summaries. A summary database has been maintained, along with corresponding user details. This database represents the reading pattern of the users. Let us call the user who requests the recommended summary of a Wikipedia article as "target user" (T). When the target user requests for a recommended summary, PerSummRe starts searching the similar readers to the target reader based on their past reading patterns. The target user is required to enter the title of the desired article and the compression ratio

Figure 3. Application interface screens. (a) Main screen with options to create a personalized summary by reading an article or to get a recommended summary. (b) Summary screen showing the recommended summary of an article. (Source: (a)PerSummRe_images\screen1300, (b) PerSummRe_images\screen2300).



(a)

(b)

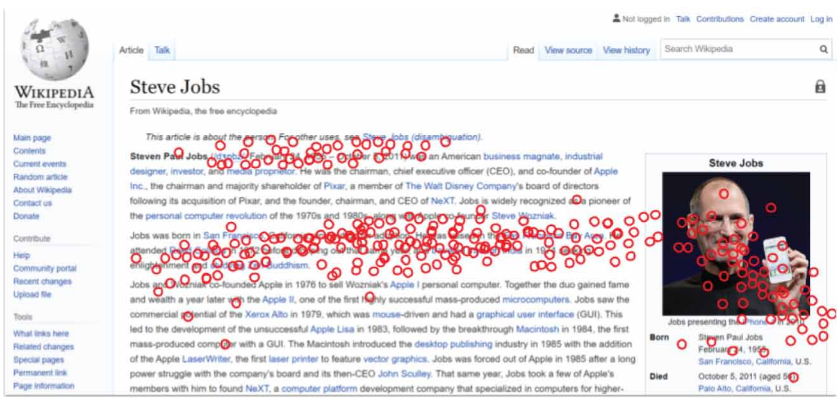
(R) to be applied during summary creation. In the pre-processing step, the system filters the users who have shared a summary of the desired article (A) and also read 60% common articles to T. This will result in a set of similar users of T, say U. The pre-processing step will reduce the complexity of the recommendation algorithm and also deals with the sparsity problem. Post primary filtering step, the recommendation system can have three possible cases based on the number of similar users (U) selected and the number of past summaries shared by the target user (S_T). Let S denote the set of summaries shared by all the users in the system. In the definition of the proposed cases, modes represent the number of items in respective sets and α and β are empirical learned constants that represent the respective lower bounds. Table 1 shows the definition of the frequently used symbols in the paper.

Case 1 - $|U|$ is greater than α and $|S_T|$ is greater than β: This scenario occurs in a mature setup where there are a sufficient number of users in the system as well as a sufficient number of shared summaries by the target user. Among the users in set U, the system finds top-k most similar users by using a pre-trained collaborative filter based deep neural network (S. Zhang et al., 2019). This

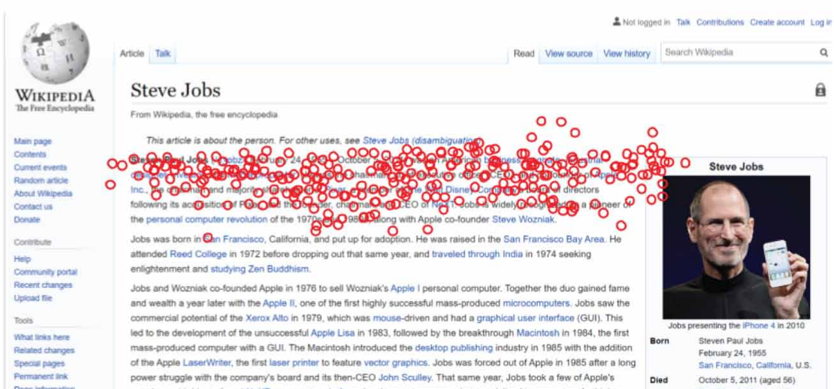
Table 1. Explanation of symbols used in the paper

Symbol	Meaning
R	Summary compression ratio defined by user
A	Desired article for which summary is requested
T	Target user who requests for recommended summary
U	Similar users to the target user
S	Set of summaries shared by all the users in the system
S_T	Set of summaries shared by the target user

Figure 4. The subfigures (a) and (b) represent gaze points (red circles) mapping of two different readers on the same Wikipedia article frame. Here, one can observe the difference between the reading preferences of different readers. (Source: (a)PerSummRe_images\gaze1300, (b) PerSummRe_images\gaze2300).



(a)



(b)

model finds the similarity between the users based on their past summaries. The scoring function used by the model for user U_i w.r.t. the user T; is defined as follows:

$$\hat{a}_{ij} = f\left(W_1^T \cdot x, W_2^T \cdot y \middle| W_1, W_2, \Theta\right) \tag{1}$$

In equation (1), x is vector representation of a summary shared by the user U_i and y is vector representation of a summary shared by $T\left(y \in S_T\right)$. W_1, W_2 are the weight matrices for the x and y vector matrices, respectively. Θ is the learning parameter of the network. This approach results in selection of top-k similar users. The value of k is set as 5 and this results in a set of most similar users based on their reading patterns. As specified in the pre-processing step, these users have also read and shared a summary on article A. Let M represent the set of selected summaries by the model. These summaries are used to create the recommended summary for T. All the sentences belonging to the selected summaries are ranked using modified TF-IDF (equation 1). TF-IDF is a popular metric to determine word/sentence relevance (Li & Zhang, 2015). Each sentence is viewed as a set of words $\left(s_i = \left(w_1, w_2, w_3, \ldots, w_n\right)\right)$. While ranking the sentences in selected summaries, one also has to consider the reading preference of the target user:

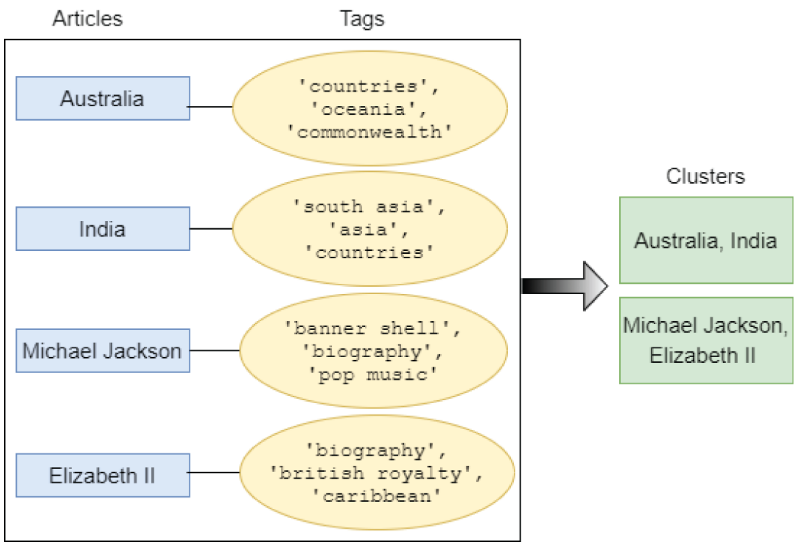
$$Score\left(s_i, S_T, M\right) = \Sigma_{w \in s_i} \frac{1}{IDF\left(w, \left(S_T \cup M\right)\right)} * \frac{TF\left(w, M\right)\left(k+1\right)}{c\left(w, M\right) + k\left(1 - TF\left(w, S_T\right)\right)} \tag{2}$$

In equation (2), $IDF\left(w, \left(S_T \cup M\right)\right)$ is the *IDF* value for w in union of sets S_T and M. $TF\left(w, M\right)$ and $TF\left(w, S_T\right)$ are *TF* values for w in M and S_T sets respectively. $c\left(w, M\right)$ is the count of the word w in summary set M. Parameter k is to be empirically set. The above equation gives more preference to sentences which contain words which have high *TF* for S_T and M, and low *IDF* for their union. Top ranked sentences are extracted while respecting the compression ratio mentioned by the target user. The uniquely extracted sentences are rearranged in their order of occurrence in the original article.

Case 2 - $\left|U\right|$ is less than α and $\left|S_T\right|$ is greater than β: This case occurs due to a huge variety of readers in the system. Here, the system has a sufficient number of past summaries shared by target (S_T) users, but do not have sufficient similar readers (U) whose summaries can be used to create a recommended summary. In this scenario, the tool works on the assumption that if a person reads about history in an article on countries, it is highly probable that he will be interested in the history portion of the new article on a country. Therefore, the system makes clusters of all the summaries present in S_T based on the tags associated with the corresponding Wikipedia articles. Each Wikipedia article contains a set of tags based on the categories it belongs to. Figure 5 demonstrates an instance of the clustering of articles based on common tags. It is interesting to note that one article can be present in multiple clusters due to semantic similarity.

Post clustering, the system identifies the cluster of the article for which the summary is desired. It gathers articles from that cluster and ranks sentences based on TF ranking. Equation (3) is used to rank sentences. The ranked sentences are fed as input to the new article:

Figure 5. Demonstration of clustering of articles based on the common tags. Tags are assigned to each Wikipedia article based on the categories it belongs to. (Source: PerSummRe_images\clustering300).



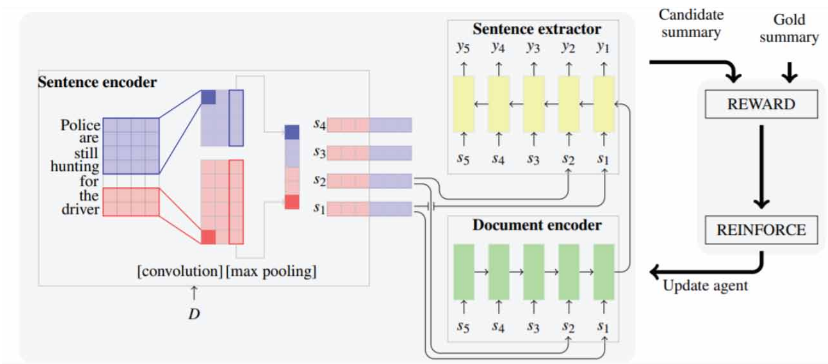
$$Score\left(s_i, S_{T_c}, A\right) = \Sigma_{w \in s_i} \frac{TF\left(w, S_j\right)}{IDF\left(w, S_{T_c}\right)} \tag{3}$$

The above equation scores the sentence s_i of summary S_J of an article which is in the same cluster as the article A. S_{T_c} denote the subset of summaries in S_T which share the same cluster as the A. The above equation prefers sentences which contain words having high TF and low IDF, i.e., which have high frequency in all the summaries. The rest of the terms carry similar meaning, as discussed above. This helps us to find out the reading pattern of T using his/her own past summaries. The system performs word frequency search and determine the rank of sentences based on their saliency for T. Post ranking, the sentences are extracted and rearranged in the same manner discussed in case 1.

Case 3 - $|U|$ is less than α and $|S_T|$ is less than β: This is the case of Cold Start, where the target user is new to the system, and thus he/she does not have enough summaries shared and thus not enough similar users. In this case, the system applies a pre-trained deep learning-based summarization model to extract sentences from the article A. The authors have adopted the model provided by Narayan et al., 2018.

This model uses temporal narrow convolution by applying a kernel filter K of width h to a window of h words in sentence s to produce a new feature. This filter is applied to each possible window of words in s to produce a feature map $f \in R^{k-h+1}$ where k is the sentence length. Then it applies max-pooling over time over the feature map f and take the maximum value as the feature corresponding to this particular filter K. It uses multiple kernels of various sizes and each kernel multiple times to construct the representation of a sentence. In Figure 6, kernels of size 2 (red) and 4 (blue) are applied

Figure 6. Extractive summarization model with reinforcement learning: a hierarchical encoder-decoder model ranks sentences for their extract-worthiness and a candidate summary is assembled from the top ranked sentences; the REWARD generator compares the candidate against the gold summary to give a reward which is used in the REINFORCE algorithm (Williams, 1992) to update the model (Source: PerSummRe_images\model300)



three times each. Max-pooling over time yields two feature lists f^{K_2} and $f^{K_4} \in R^3$. The final sentence embeddings have six dimensions.

The system uses the pre-trained model and fine-tune it on all the summaries shared on the desired article *A* by all the users in the system. A learning rate of .0001 and 10 epochs have been used while fine-tuning the model for current summary dataset. This model ranks the sentences in the article for extraction. Post ranking, the sentences are extracted and rearranged in the same manner discussed in case 1.

EXPERIMENTS AND RESULTS

To evaluate whether the eye gaze approach can improve recommendation systems' performance, the authors develop a standalone application named PreSummRe. The experiments are conducted in the setup of a computer lab. PreSummRe is an Operating System independent application. It was implemented in the Ubuntu environment and tested for Ubuntu, Microsoft Windows, and Mac environments. It is developed in Python, C, PHP, and SQL Server.

A total of 78 undergraduate and graduate students have been invited to participate in the dataset collection process. The participants belong to various majors. Prospective participant-observers were required to have normal visual acuity. They ranged in age from 19 to 31 (M = 23.75, SD = 4.71). They have been asked to read Wikipedia articles belonging to five categories: biography, countries, president, music, and food. As mentioned earlier, to enable receiving recommended summaries, users need to use the tool to create at least five summaries by reading Wikipedia articles. These summaries help in learning users' reading patterns.

Along with this, some of the users are also asked to create manual summaries of any Wikipedia article. At the end of every reading session, participants have been asked to create a manual summary of the article for which the tool has generated a recommended summary. This gives us the ground-truth value for sentence extraction during performance analysis. By the end of the data collection process, the authors can collect a total of 648 summaries created by users using the tool, 170 recommended summaries, and 170 manual summaries. These summaries represent a unique dataset of user preference in Wikipedia.

Table 2. WikiGaze summary quality evaluation metrics

Metric	Average Value
P	40.703
R	36.610
f-measure	38.548
FRE	67

Evaluation of Summary Creation Tool

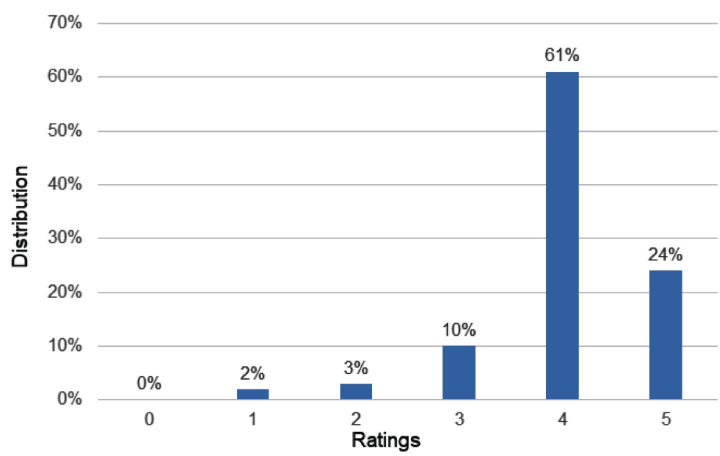
The authors showcase the efficiency of WikiGaze by conducting ROUGE-1(F. Liu & Liu, 2008) analysis and by calculating readability value of each summary. The official ROUGE script (http://www.berouge.com/) (version 1.5.5) has been used to evaluate the summarization output. The ROUGE is calculated between the automatic summaries generated by the WikiGaze tool and the human made summaries.

To evaluate the readability of the summaries created by WikiGaze tool, one of the most famous estimators called Flesch-Kincaid Grade Level (FRE) (Kincaid et al., 1975) has been used. The FRE formula is a combination of sentence length, word length, and word and sentence density (equation 4). It works with an underlying assumption that longer sentences and words indicate greater reading difficulty. A higher FRE index of a text document indicates a higher difficulty level:

$$FRE = 0.39\left(\frac{words}{sentence}\right) + 11.8\left(\frac{syllables}{word}\right) - 15.59 \quad (4)$$

The results of the evaluation are shown in Table 2. One can observe that the summaries score high values in the ROUGE metric, which indicates that the extraction process proposed by the WikiGaze tool is very efficient. There is a considerable overlap between the automatic and manual summaries. It can also be observed that the average readability score (FRE) is 67, which indicates that the summaries have moderate difficulty level in reading the text.

Figure 7. Plot demonstrating the distribution of user ratings for the summary recommendations



Evaluation of Summary Recommendation

To measure user satisfaction level with the recommended summaries, users are asked to provide ratings to the summaries. The ratings vary in the range of 1 to 5. The user rating distribution plot can be seen in Figure 7. It is evident from the plot that users like the quality of the recommended summaries and thus gave high ratings. The mean value for user rating is 3.8 (out of 5).

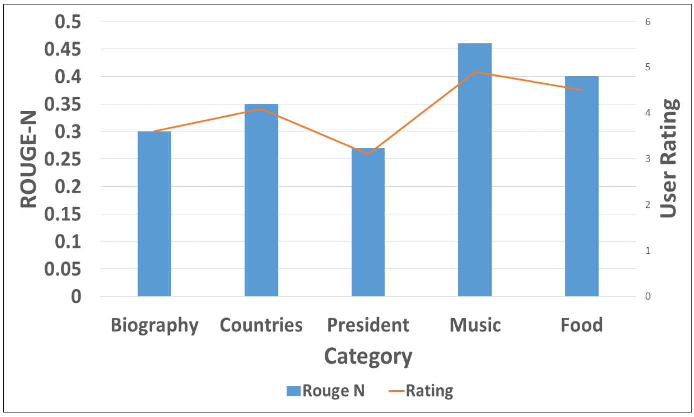
- **User satisfaction evaluation:** To measure user satisfaction level with the recommended summaries, users are asked to provide ratings to the summaries. Users give this rating based on the ease of using our tool and the tool's summaries' quality. The ratings vary in the range of 1 to 5. The user rating distribution plot can be seen in Figure 7. It is evident from the plot that users like the recommended summaries' quality and thus gave high ratings. The mean value for user rating is 3.8, with a standard deviation of 0.72. The authors observe that 85% of the participants gave four or more ratings for the recommended summaries. The high user satisfaction ratings show the efficiency of the proposed reading pattern analysis approach.
- **Comparison with manual summaries:** To evaluate the proposed summary recommendation technique's performance, the authors have used the ROUGE-N (N=1,2) metric. It measures summarization quality by counting overlapping units such as the N-gram, word sequences, and word pairs between the candidate summary and the reference summary. The ROUGE-N calculates the recall score (R), the precision score (P), and the text document's f-measure score. Table 3 shows the metrics' normalized values for the recommended summaries versus manual summaries, and the tool created summaries for the same article. The experimental results' overall values, normalized across various article categories, are shown in Figure 8(a). The results showcase the positive correlation between ROUGE-1 (f-measure) and user rating.

The recommendation system's performance is supposed to improve with the increase in the system's knowledge base. The authors evaluate the impact of the summary repository's size created and shared by each user on the performance of the respective recommendation. The ROUGE-1 value has been calculated for the recommendation for different summary repository sizes belonging to various users. Figure 8(b) shows the direct proportionality between the summary repository size and the average ROUGE-1 of the recommended summaries. It motivates users to share more summaries. It shows that the performance of the recommendation method increases with the increase in summary database.

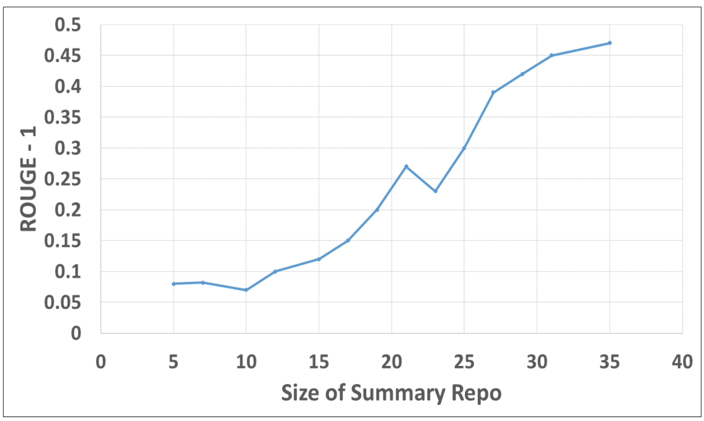
- **Comparison with state-of-the-art deep models:** The process of recommending summaries can be considered a special case for generating summaries based on users' previous reads. For comparison with the extractive models, the authors include LEAD-k, a strong baseline for single document summarization tasks and takes the first k sentences in the document as summary (Abigail See, 2017). LexRank (Radev., 2004) is a graph-based method, where nodes are text units, and a similarity measure defines edges. SumBasic (Nenkova, 2005) is a frequency-based sentence selection method that uses a component to re-weigh the word probabilities to minimize redundancy. Other extractive baselines are the near state-of-the-art models C SKIP from (Rossiello, 2017) and SemSenSum from (Faltings., 2019). The former exploits word embeddings' capability to leverage semantics, whereas the latter aggregates two types of sentence embeddings using a sentence semantic relation graph, followed by a graph convolution. The last model PriorSum (Z. Cao, 2015), combines neural models and dozens of hand-crafted features.

The authors fine-tune the models for Wikipedia data dump 5. This dump contains only the current revision of all the English Wikipedia articles as of September 22, 2019. It does not contain talk or user pages. During fine-tune, a learning rate of 0.0001 with ten epochs has been used. All

Figure 8. (a)Variation of ROUGE-1 and User rating for various categories of articles. (b) Variation of ROUGE-1 with the size of summary repository.



(a)



(b)

Table 3. Evaluation of recommended summary verses automatically generated summary by tool and manual summary

Metric	Recommended vs. Automatic summary	Recommended vs. Manual summary
nR	0.326	0.28
nP	0.42	0.35
nf-measure	0.367	0.311

models are fine-tuned on an Nvidia GeForce GTX 1080 Ti GPU (60GB RAM, 12 GB dedicated graphic card, and 200 GB Hard drive space). The authors run all models with their best-reported parameters. Post fine-tuning the pre-trained models, the performance comparison of these models has been performed. They have been compared with the proposed approach to the collected dataset of automatic and manual summaries (along with source articles for each summary). The authors

Table 4. Comparison with other extractive summarization deep models

Model	R-L	R-1	R-2
LEAD-5	23.18	11.81	3.22
LexRank	29.14	23.22	4.92
SemSenSum	30.46	25.56	3.79
C SKIP	30.66	32.90	4.25
SumBasic	32.42	35.11	5.56
PriorSum	29.80	35.92	6.01
Ours	30.92	37.61	6.26

consider three vital reference-based evaluation metrics: ROUGE 1, 2 & L. Authors use the recalls of ROUGE-1 (unigram), ROUGE-2 (bigram), and ROUGE-L (LCS). The official ROUGE script6 (version 1.5.5) has been used to evaluate the summarization output. In Table 4, one can see the comparison result. Our approach performs better than these models for ROUGE 1 & 2. SumBasic performs better for the ROUGE-L metric on our dataset. The reason for the better performance of our approach for ROUGE 1 & 2 is that it directly extracts sentences based on user preference. While SumBasic performs better in ROUGE-L because it considers the frequency of the words in the source documents, it results in the selection of repeatedly mentioned sentences, which again have a higher probability of being read by the users.

LIMITATION AND FUTURE WORK

Currently, our system is capable of creating recommended summaries for single articles. In the future, the authors plan to enable the feature of creating meta-summaries, which will include a summary of multiple articles mentioned by the user. The authors also plan to extend the idea for other crowd-sourced portals, such as Stack Exchange, Reddit, and Quora. They believe the proposed method can help understand users' engagement on these portals by unraveling (Abigail See, 2017) their eye gaze information. It will help editors to identify areas in an article/post where readers pay more attention.

In current work, a standalone application has been developed to generate summary recommendations. Still, the application can be replaced with a browser plug-in to enhance ease of usage in the future.

The authors are in talks with the Wikimedia foundation to include our application in Wikipedia as a separate tab in the Wikipedia interface. This will help us collect a huge dataset of user behavior on Wikipedia due to Wikipedia's huge reader base.

CONCLUSION

The number and size of Wikipedia articles are increasing at an exponential pace. It makes it difficult for users to find the required data in the massive ocean of information. In this paper, the authors presented a novel approach to create a summary recommendation system for Wikipedia. The summaries are recommended based on the user's reading pattern. So that user need not search for hidden information over and over again. The proposed recommendation system works in collaboration with an eye gaze-based summarization system. The authors discussed how one could utilize a user's eye gaze to capture his/her ROI in an article. The system filters the similar readers present in the system and appropriately integrates their summaries to generate a recommendation for the user who has demanded it.

ACKNOWLEDGMENT

This work was funded by the assistance received from CSRI, Department of Science and Technology India via grant no. SR/CSRI/344/2016.

REFERENCES

Ahn, D. H., Kim, J. K., Choi, I. Y., & Cho, Y. H. (2004). A personalised recommendation procedure based on dimensionality reduction and web mining. *International Journal of Internet and Enterprise Management*, *2*(3), 280–298. doi:10.1504/IJIEM.2004.005372

Ai, Q., Zhang, Y., Bi, K., Chen, X., & Croft, W. B. (2017, August). Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 645-654). ACM.

Ajanki, A., Hardoon, D. R., Kaski, S., Puolamäki, K., & Shawe-Taylor, J. (2009). Can eyes reveal interest? Implicit queries from gaze patterns. *User Modeling and User-Adapted Interaction*, *19*(4), 307–339. doi:10.1007/s11257-009-9066-4

Cao, F. W. (2015). Learning summary prior representation for extractive summarization. *ACL*.

Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, *23*(3), 329–342. doi:10.1016/S0957-4174(02)00052-0

Cosley, D., Frankowski, D., Terveen, L., & Riedl, J. (2007, January). SuggestBot: using intelligent task routing to help people find work in wikipedia. In *Proceedings of the 12th international conference on Intelligent user interfaces* (pp. 32-41). doi:10.1145/1216295.1216309

Erkan, G. a. (2004). Lexpagerank: Prestige in multi-document text summarization. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 365-371.

Faltings, D. A. (2019). *Learning to Create Sentence Semantic Relation Graphs for Multi-Document Summarization.* arXiv preprint arXiv:1909.12231.

Farnsworth, B. (n.d.). *Top 12 Eye Tracking Hardware Companies (Ranked).* Retrieved September 21, 2020, from https://imotions.com/blog/top-eye-tracking-hardware-companies/

Frankowski, D., Lam, S. K., Sen, S., Maxwell Harper, F., Yilek, S., Cassano, M., & Riedl, J. (2007). Recommenders everywhere. *Proceedings of the 2007 international symposium on Wikis - WikiSym '07*. doi:10.1145/1296951.1296957

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61–70. doi:10.1145/138859.138867

Kincaid, J. P., Fishburne, J., P., R. R., L., C. R., & S., B. (1975). *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. 10.21236/ADA006655

Lee, J.-W., Lee, S.-G., & Kim, H.-J. (2011). A probabilistic approach to semantic collaborative filtering using world knowledge. Journal of Information Science, 37(1), 49–66. doi:10.1177/0165551510392318

Li, L., & Zhang, R. (2015). *Recommended model of information flow based on TF-IDF*. 10.14257/astl.2015.82.02

Liang, T.-P., Yang, Y.-F., Chen, D.-N., & Ku, Y.-C. (2008). A semantic-expansion approach to personalized knowledge recommendation. Decision Support Systems, 45(3), 401–412. doi:10.1016/j.dss.2007.05.004

Liu, F., & Liu, Y. (2008). Correlation between ROUGE and human evaluation of extractive meeting summaries. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Short Papers - HLT '08.* doi:10.3115/1557690.1557747

Liu, P. J. (2018). *Generating Wikipedia by summarizing long sequences.* arXiv preprint arXiv:1801.10198.

Liu, Y. a. (2008). Personalized PageRank based multi-document summarization. In *IEEE International Workshop on Semantic Computing and Systems* (pp. 169-173). IEEE. doi:10.1109/WSCS.2008.32

Liu, Y., Wang, X., Zhang, J., & Xu, H. (2008). Personalized PageRank Based Multi-document Summarization. *IEEE International Workshop on Semantic Computing and Systems.* doi:10.1109/WSCS.2008.32

Lüer, C., & Cummins, J. (2009). Collaborative Web Search with WikiLinks. *2009 Sixth International Conference on Information Technology: New Generations.* doi:10.1109/ITNG.2009.293

Lyman, P., Varian, H. R., Dunn, J., Strygin, A., & Swearingen, K. (2000). *How much information?* Berkeley: School of Information Management and Systems.

Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. *Proceedings of the fifth ACM conference on Digital libraries - DL '00.* doi:10.1145/336597.336662

Moskalenko, O. P.-T. (2020). *Scalable Recommendation of Wikipedia Articles to Editors Using Representation Learning.* arXiv preprint arXiv:2009.11771.

Musto, C. S. (2016). Learning word embeddings from wikipedia for content-based recommender systems. *European Conference on Information Retrieval*, 729-734. doi:10.1007/978-3-319-30671-1_60

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Ranking Sentences for Extractive Summarization with Reinforcement Learning. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1. doi:10.18653/v1/N18-1158

Nenkova, A. &. (2005). *The impact of frequency on summarization.* Microsoft Research, Tech. Rep. MSR-TR-2005, 101.

OECD. (2010). *Rapid growth of Wikipedia.* <ALIGNMENT.qj></ALIGNMENT>10.1787/trends_edu-2010-graph50-en

Abigail See, P. J. Abigail See, P. J. (2017). *Get to the point: Summarization with pointer-generator networks.* arXiv preprint arXiv:1704.04368.

Park, S. (2008). Personalized Document Summarization Using Non-negative Semantic Feature and Non-negative Semantic Variable. *Lecture Notes in Computer Science*, *5326*, 298–305. doi:10.1007/978-3-540-88906-9_38

Piccardi, T. C. (2018). Structuring Wikipedia articles with section recommendations. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

Pon, R. K., Cardenas, A. F., Buttler, D. J., & Critchlow, T. J. (2007). iScore: Measuring the Interestingness of Articles in a Limited User Environment. *2007 IEEE Symposium on Computational Intelligence and Data Mining.* doi:10.1109/CIDM.2007.368896

Radev, D. R., Fan, W., & Zhang, Z. (2001, June). Webinessence: a personalized web-based multi-document summarization and recommendation system. *NAACL Workshop on Automatic Summarization.*

Radev., G. E. (2004). *LexRank: Graph-based Lexical Centrality as Salience in Text Summarization.* ArXiv abs/1109.2128

Rossiello, G. B. (2017). Centroid-based text summarization through compositionality of word embeddings. *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, 12-21. doi:10.18653/v1/W17-1003

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. *Proceedings of the 2nd ACM conference on Electronic commerce - EC '00.* doi:10.1145/352871.352887

Singer, P., Lemmerich, F., West, R., Zia, L., Wulczyn, E., Strohmaier, M., & Leskovec, J. (2017). Why We Read Wikipedia. *Proceedings of the 26th International Conference on World Wide Web.* doi:10.1145/3038912.3052716

Szymański, J. (2010). Mining Relations between Wikipedia Categories. In Networked Digital Technologies (pp. 248–255). doi:10.1007/978-3-642-14306-9_25

Uckan, T. a. (2020). Extractive multi-document text summarization based on graph independent sets. Egyptian Informatics Journal. doi:10.1016/j.eij.2019.12.002

Van Gysel, C., de Rijke, M., & Kanoulas, E. (2016). Learning Latent Vector Spaces for Product Search. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* doi:10.1145/2983323.2983702

Wikipedia. (n.d.). *Size comparisons*. Retrieved September 21, 2020, from https://en.wikipedia.org/w/index.php?title=Wikipedia:Size_comparisons&oldid=979060489

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System. ACM Computing Surveys, 52(1), 1–38. doi:10.1145/3285029

Zhang, Y., Chen, X., Ai, Q., Yang, L., & Bruce Croft, W. (2018). Towards Conversational Search and Recommendation. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* doi:10.1145/3269206.3271776

Neeru Dubey is currently a PhD student at the Indian Institute of Technology Ropar, India. Her research interests include Collective Intelligent, Computer Vision, and Natural Language Processing. She received her bachelor's degree in Computer Science and Engineering from Guru Gobind Singh Indraprastha University (New Delhi, India).

Amit Arjun Verma is pursuing his Ph.D. from the Computer Science department of the Indian Institute of Technology Ropar. He has completed his M.Tech in Computer Science and Engineering from the National Institute of Technology Meghalaya in 2016. His primary research area is understanding Collective Intelligence via online collaborative portals. He is also interested in developing open-source tools to analyze the dataset of collaborative portals such as Wikipedia, Stack Exchange, and GitHub.

Simran Setia is a PhD Scholar at the department of Computer Science and Engineering, IIT Ropar, India. She completed her Master's degree in Computer Science and Technology from Thapar Institute of Engineering and Technology, Punjab, India.

Sudarshan graduated from the CSA department of IISc and is currently working as an associate professor in the department of computer science and engineering at IIT (Ropar Campus). His areas of interest include collective intelligence, machine intelligence and social computing. He has graduated 4 PhD students and is currently guiding 10. He has recorded 3 online MOOC courses each of 30 hours duration and takes active interest in propagating science through popular science lectures. He has delivered over 120 talks and has conducted more than 30 FDPs in different parts of the country.