Personalized search on news articles

Project group 6

Hamza Benjelloun, Yuqi Liu, Artin Mirzaian, Fynn van Westen
KTH Royal Institute of Technology
May 11, 2021

Contents

List of Figures

1. Abstract

By creating user profiles with implicit feedback (history of visited URLS) it is possible to personalize search results for users. And once a multitude of such profiles are stored in the search engine it is possible to use a collaborative filtering algorithm that calculates the similarity between items using users' interactions with those items. By calculating similarity between items the system can then recommend related items once a user has clicked on an item. It was found that by representing a user in document space by only looking at the user's 10 latest visited URLS it is possible to re order the search results in a way that leads to more relevant documents (for the user) appearing at the top of the results. Furthermore, a qualitative analysis showed that for popular items, the related items that were recommended by the system were found relevant.

2. Introduction

The topic of personalized information retrieval, or PIR, is a sub-field of the general Information Retrieval domain. The individual nature of information need makes the problem of information retrieval a ambiguous one. This means that the use of the information, returned by the system, depends on the individual problem and interest of the querying user. Two users who submit an identical query to the system might have very different informational needs. A famous example for this from literature is the query "Java". Depending on the users context this simple query could refer either to the programming language, the Indonesian island or something related to any of these, like java coffee named after the island. This problem is rooted in the ambiguity of language and words.

To allow users to find the information intended, feedback mechanisms were invented. One of the most famous ones is the rocchio algorithm [13], which utilizes direct feedback from users about the informational content of the retrieved results from the user for its purpose. Due to reasons like convenience, usability etc., development of passive methods utilizing implicit information, instead of explicit user feedback becomes more and more popular. These techniques rely on user profiles and histories, as described in various publications [5] [10]. Once a multitude of such profiles and histories are combined, more possibilities such as collaborative filtering approaches to infer user specific relevance by comparison to other users metadata, become possible and can provide remarkable results.

This report consists of two folds, first an algorithm to rerank the search results retrieved from search engine for personalization and an algorithm to recommend related items once a user has clicked on an item.

3. Background

Previous work on personalized search engines can be divided into two main aspects. First the collection of user's personal data, by observing the interactions or background information. The scope in which the collection of data can be conducted is defined in matters of time and source (web, disk, etc.). Secondly the collected information has to be accumulated to represent the user profile. Then the search engine would use the user profile to rerank or filter the search results. Since our work also includes non-personalzied recommendation (RS), the recent popular recommender systems without being personalized can be classified into content based RS and collaborative filtering RS.

3.0.1 Dataset

In this article the Microsoft News Dataset (MIND) [14] for news recommendation collected from Microsoft news services. It contains about 160.000 news articles in English language and over 15 million impression logs created by one million users. The news articles contain amongst more title, abstract, body, category and entities. While the user specific data contains click events, non-clicked events and historical news click behaviors of anonymized users before the impression. In this article specific interest lies on the history data of users, utilized in the recommendation by collaborative filtering.

3.1 Personalized search

3.1.1 User Interest collection

Many studies on personalized search have been using the recorded history of user interactions with the system. For short-term personalized systems, Sriram et al. [9] has conducted a system following the current user session. For users who retrieve documents for short-term information needs, this approach may provide a more useful result. However, the session data often gives non-ideal results due to the sparse data it holds compared to the long-term personalization. Teevan et al. [11] has proposed a long-term user interests model, which tracked rich data from not only visited website history but also indexing the files on hard drive, emails and etc. Dou et al. [2] suggested that both long-term and short-term user interests models improve the performance of personalized search results and help building the user profile. The short-term approach is particularly applicable for histories and current informational needs, while the long-term approach can be used to build an overall profile of the user, his preferences and circumstances.

3.1.2 User profile Representation

Most of the previous research seeks to represent the user based on his history, for a short term representation. The general idea is to create a representation of the user in the document space by aggregating all documents the user has visited. The representations differ in the way of assigning the weight, for example in the article Personalizing Web Search using Long Term Browsing History [7], the author proposes 3 different ways of weighting, the first is with term frequency TF, the second is TF-IDF and the third is BM25.

There are other more advanced methods that propose to consider the user as two vectors, one static that is related to the age, the country and the language that the user speaks and the other one is dynamic, it uses one of the weighing methods already presented [4]. In addition, there are those who try to model the changes in the preferences of the users, they use the scheme of exponential decrease as described in the article Adaptive web search based on user profile constructed without any effort from users [10]. In this way, a user's preference expressed a long time ago has less weight than the ones expressed recently.

3.2 Recommendation Systems

A recommendation system attempts to detect the most suitable content, based on some domain specific criteria. Most recommendation systems use similarity as a proxy to give recommendations. There are mainly two types of approaches, collaborative filtering and content-based filtering. The former is based on a set of user interaction with items, while the latter uses item content descriptions and user thematic profiles. There are models that only use collaborative filtering for example in the article "How to build a Recommendation Engine quick" [12] and models that use only content-based approach as illustrated in the article "Trends in content-based recommendation"[6]. But the most powerful are those that use a hybrid model that combines both. In this report we will focus on a collaborative filtering approach. The idea of collaborative filtering is to base its recommendations on the behaviour of other users of a system. The critical assumption of collaborative filtering is that if a group of users agree about the rating or relevance of some issues then these users are likely to have the same opinion on a different issue [3]. In this report we will more specifically focus on a collaborative filtering algorithm that is based on a item-item model. The item-item model is a type of collaborative filtering that calculates the similarity between items using users' interactions with those items. A common approach with an item-item model is to look at co-occurrences and make predictions based on the co-occurrence matrix. Further explanation of how an item-item model can be used for recommendation will be explained in the method.

4. Methods

The used system architecture, see Figure 4.1, to conduct experiments consists of multiple aspects. First the elastic search engine which stores and retrieves the news documents and user information from its indices. Due to parallelisation in the work process, we have two servers in parallel communicating with each other. First a flask web-server which implements a narrow set of API connections and functions as a gateway to the elastic search engine. And secondly a node web-server hosting the frontend for the user and fetching information from the flask gateway server. The information flow can be described as follows. A user issues a search query on the frontend to the node server. The frontend server connects to the corresponding API interface on the backend, gateway server. This server retrieves information from the elasticsearch indices and aggregates the retrieved information into a final results which is send back as a response to the frontend server who displays the response to the user. Other than that the backend server is also where the personalization is conducted by incorporating the users history in the result, which is described in more detail following. Once the results are obtained and the user clicks on one the recommendation system will recommend similar articles based on collaborative filtering. The idea behind this is to avoid the lock-in phenomenon, due to information bubbles resulting from too dominant bias in the user profile.

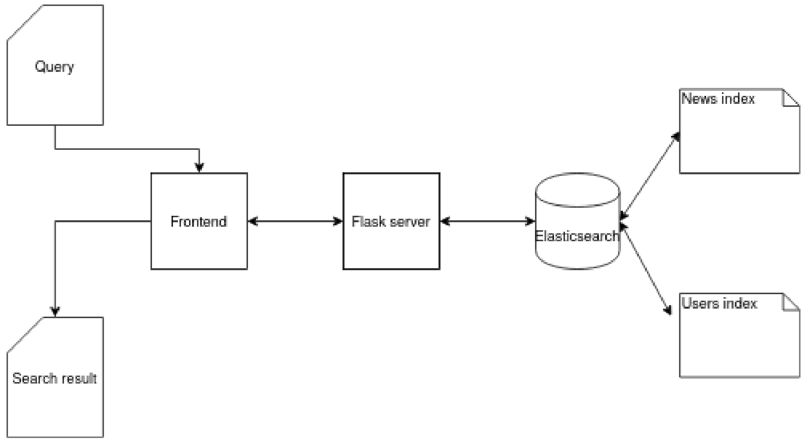

Figure 4.1: Network architecture

4.1 News Recommendation

The recommendation system is based on collaborative filtering, more specifically an item-item model as described in [12] and the corresponding section3.2. The algorithm is as follows:

1. When a user has clicked on a news article X, we search for all users who have X in their history. This defines our foreground population.
2. We look for co-occurrence in our foreground which gives us indication that the news article Y can be relevant.
3. We search for the users that have Y in their history but not X, this will define our background population.
4. We then calculate the score for each Y with the following equation,

$$Y_{score} = (\frac{FG_{hits}}{FG_{total}} - \frac{BG_{hits}}{BG_{total}}) * \frac{\frac{FG_{hits}}{FG_{total}}}{\frac{BG_{hits}}{BG_{total}}} \tag{4.1}$$

where,

$$FG_{hits} = \text{ Number of users with X and Y in their history} \tag{4.2}$$
$$FG_{total} = \text{ Number of users with X in their history} \tag{4.3}$$
$$BG_{hits} = \text{ Number of users with Y but without X in history} \tag{4.4}$$
$$BG_{total} = \text{ Number of users without X in history} \tag{4.5}$$
$$\tag{4.6}$$

Clarifications:

- The variables in equations 4.2 - 4.5 can be visually interpreted in Figure 4.2.
- In step 2 it is possible to define a parameter *min_docs* which will regulate the sensitivity of the algorithm. The *min_docs* parameter will determine the minimum amount of users in the foreground who needs to have Y in their history in order for Y to be considered to be possible relevant.

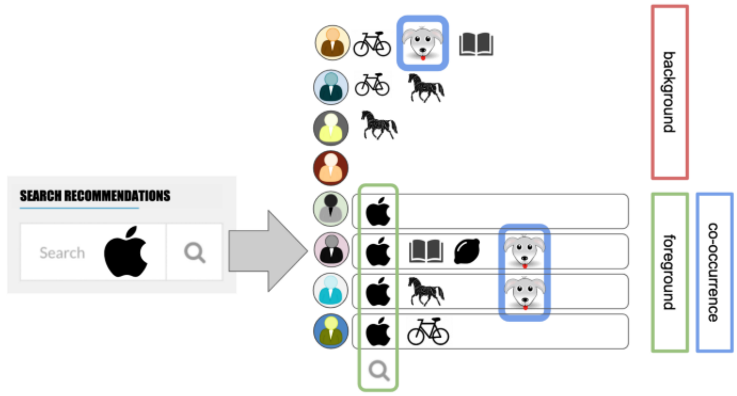

Figure 4.2: Visual explanation of variables for recommendation scoring function taken from [12]

- The Elasticsearch specific JLH scoring function [1] utilized in step 4 consists of the following two parts. To the left of the multiplication we have a absolute change which would favour common news articles whereas to the right a relative change which favours rarer articles. The multiplication therefore, evens out the scoring.

4.2 News Personalization

4.2.1 User and document representation

The main method used to exploit implicit feedback is to look at the click history to build user profiles, in our implementation we look at the 10 last clicks. Each document is represented as 3 TF-IDF vectors and were computed from three different sources. Let the TF-IDF vectors of each document be defined as follows,

- V_{title}: terms of the title.
- $V_{category}$: terms of the category.
- V_{body}: terms of the body.

Let V'_{title} be defined as the average TF-IDF vector over all V_{title} derived from the users 10 latest visited news articles. Define $V'_{category}$ and V'_{body} accordingly. This way the user is represented as a vector u in the document vector space,

$$u = W_b V'_{body} + W_c V'_{category} + W_t V'_{title} \tag{4.7}$$

where W is the associated weight for each of the sources. The choice of weights used in our experiments are based on educated guesses and adjustment by try and error on a personal evaluation. It is to mention that a more sophisticated parameter tuning scheme, like a grid search, could vastly improve the method.

4.2.2 Re-ranking strategy

Re-ranking is performed on the retrieved documents of the regular search method of Elastic Search. For each retrieved document d_i the TF-IDF vector is calculated as,

$$d_i = W_b V_{body} + W_c V_{category} + W_t V_{title} \tag{4.8}$$

Since the user vector u is a representation in the same vector space, the final ranking is calculated using the cosine similarity between result vectors and the user vector. which is following added to its original score assigned by the search algorithm.

$$New\_score_i = (1-p) * old\_score_i + p * cos(u, d_i) \tag{4.9}$$

where, p is the personalization parameter and $p \in [0, 1]$.

4.2.3 Evaluation

Evaluation on the personalized search results depends on the current user's subjective intention, hence the following rating scale is a measurement to give a score to the documents from the search results based on relevancy as described in [8].

- (0) Irrelevant document. The document does not contain any information about the topic.
- (1) Marginally relevant document. The document only points to the topic. It does not contain more or other information than the topic description.
- (2) Fairly relevant document. The document contains more information than the topic description, but the presentation is not exhaustive.
- (3) Highly relevant document. The document discusses the themes of the topic exhaustively.

Cumulative Gain (CG) is the sum values of all the graded search results points using the criteria above. It does not take the position of certain search result into consideration. Discounted cumulative gain (DCG) will reward the relevant results that are located at higher positions and penalized the higher rating results at lower positions. Normalized Discounted Cumulative Gain (NDCG) is the DCG results normalized by ideal DCG. The ideal DCG (IDCG) is used to determine the ideal results score value from the most relevant to the least. By normalizing the results, the harder and the easier queries will be evaluated to a fair stage.

$$CG_n = \sum_{pos=1}^{n} relevance_{pos}$$

$$DCG_n = \sum_{pos=1}^{n} \frac{relevance_{pos}}{log_2(pos+1)}$$

$$NDCG_n = \frac{DCG_n}{IDCG_n}$$

5. Experiments

5.1 Experiment one: Query = "apple"

The aim of this experiment is to investigate if the system returns different results depending on the user's profile. The profile of the user consists of the user's personal history of clicked articles in this system.

User B in this experiment was interested in healthy fruits and had a history consisting of articles related to different types of fruits.

The experiment was to examine what type of articles User B retrieves from the search engine when searching for "apple" and varying the personalization factor p mentioned in the method. The majority (23 out of 25) of news articles retrieved from Elastic Search are related to the company Apple whilst as one can expect from User B's history, User B is interested in news articles about the fruit Apple. The relevancy for each news articles has been rated for User B for the query "apple" and thus a NDCG scoring can be calculated. In Figure 5.1 the NDCG score is displayed for different values of p.

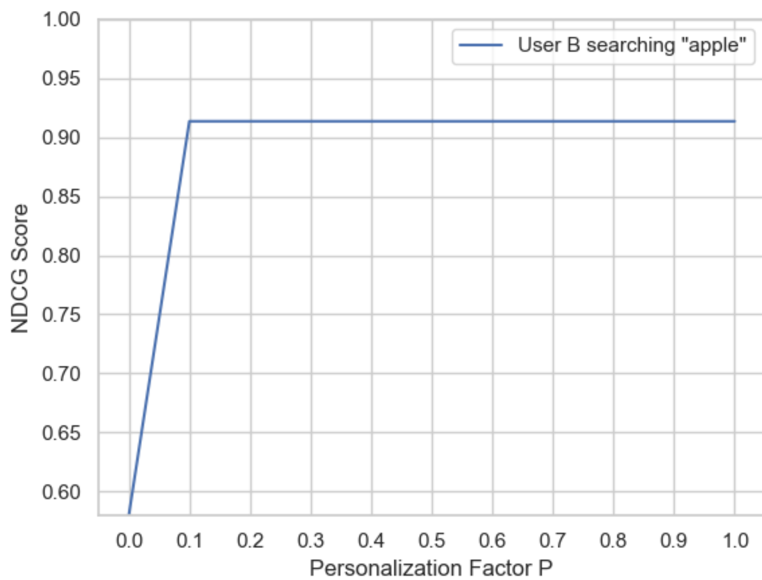

Figure 5.1: NDCG score for User - A for query "apple" when varying p.

5.2 Experiment two: Query = "markets"

The goal of this experiment is to show the different results retrieved by the system, depending on an individual user profile. These profiles consist exclusively of the users personal history of clicked articles in this system. The query "markets" was chosen due to its ambiguous meaning as financial markets or physical markets like farmers markets.
User A in this experiment was interested in food, shopping, farming and sports. While User F was interested in finance, sports, stocks, health and diet. Therefore each of them has corresponding results in his histories look back window8.2. Each user issued the query "markets" to the system and scored his individual, as well as the general, results on a 0 to 3 scale for to interpret the systems performance as an NDCG score. As described in section 4.2.2 the final result ranking is accumulated from the personalization and the general results, according to factor P. Further the influence of factor P on the results was investigated, the result can be seen in 5.2.

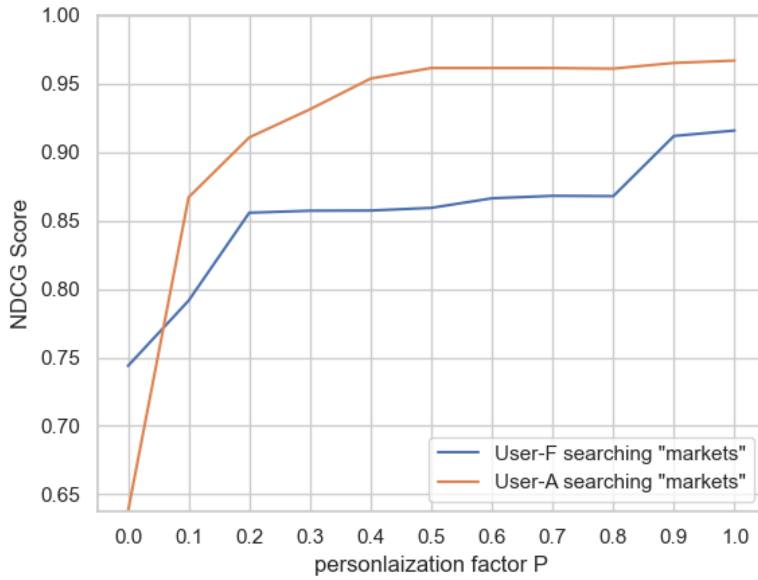

Figure 5.2: NDCG score to P value relation

5.3 Experiment three: Recommendations

In this experiment the search engine was evaluated on how well it recommends related news based on a currently selected article. Two different news articles were chosen for this purpose. One of the articles chosen was about Trump with the title *Trump blasts 'Never Trump' Republicans as 'human scum'*. The recommended articles can be seen in Figure 5.3. The other article was about Taylor Swift with the title *Taylor Swift Praises Madonna for 'Outstanding' N.Y.C. Concert: 'M Gave It Everything'*. The recommended articles are shown in Figure 5.4.

6. Discussion

6.1 Apple Query - Experiment

As seen in Figure 5.1 there is a significant increase of the NDCG score for User - A when p goes from 0 to 0.1. However, as p continues to grow, the NDCG score stays constant. The explanation for this could be due to only 2 out of 25 news articles are rated as relevant for User - A and therefore, once the personalization factor p has overcome a threshold where the fruit related articles are top ranked, the ranking of the articles does not change.

Figure 5.3: A screenshot of the recommended articles for the news article "Trump blasts 'Never Trump' Republicans as 'human scum'".

6.2 Market Query - Experiment

The purpose of this experiment was to validate if the employed personalization techniques reliably retrieve information of specific interest for an individual user. Further not just a difference between personalized retrieval and the results without such influence, this experiment was also conducted to investigate the influence of different user profiles for the same query.
As it can be seen in figure 5.2 the increase of the personalization factor P steadily increases the systems performance in this experiment. There is a visible difference between the two users which can be traced back to the specifics of our data-set.
From the results it becomes evident, that the user specific interests embedded in the history are considered in the results. Each user gets specific results for his informational needs. It also remains to say, that during the experiments it has been noticed that the results can be susceptible already to small changes in the history. It is believed that the nature of the document corpus and the chosen weighting of document parts are responsible for this. It then was deducted that a proper weight initialization scheme should be performed as a separate experiment.

6.3 Recommendation - Experiment

The purpose of this experiment was to see the recommendation performance of the system. In this experiment a qualitative evaluation was made by the authors. For the first article with the title *Trump blasts 'Never Trump' Republicans as 'human scum'* the system recommended relevant articles as 9/10 of them were about Trump, see Figure 5.3. However, the recommendations for the article about Taylor Swift was less successful, as none of the recommended articles were considered relevant. The difference in results can be explained by the

Figure 5.4: A screenshot of the recommended articles for the news article "Taylor Swift Praises Madonna for 'Outstanding' N.Y.C. Concert: 'M Gave It Everything'".

difference in popularity of the articles. As described in the method, the recommendation is solely based on the behaviour of other users that have visited the same article and therefore, if a news article has few clicks, the recommendation will often lead to non relevant results. Further investigation showed that the number of users who had visited the Taylor Swift article was 168 while for the Trump article 3377. One can imagine that as the data grows with more users and more clicks, the recommendation algorithm will give have better performance. The evaluation of the suggestions was conducted on a personal base and are described more in the discussion section7.

7. Conclusion

As shown in the results, building user profile in information retrieval system can improve personalized search results based on the user's interests.

We explored two different methods in our project for improving the search results of a search engine that were based on implicit feedback (clicks) from the users. One of the methods is based on collaborative filtering where the system recommends news articles once a user has clicked on a certain article, the other method is reordering the search results based on a TF-IDF representation of the user using his history.

Although we have not rigorously evaluated our personalized search engine, the results we present show that using clicks as implicit feedback can work in practice and that our methods have improved the search quality of the search engine.

8. Appendix

8.1 Experiment "apple" query User profile:

User B:

- Is corn a fruit, a vegetable, or a grain? » foodanddrink>foodnews
- 12 Foods You're Probably Refrigerating That You Actually Shouldn't » foodanddrink>tipsandtricks
- How to stop throwing away your veggies and fruit » foodanddrink>tipsandtricks
- 8 Clear Signs You're Not Eating Enough Vegetables » health>nutrition
- 10 of the Healthiest Fruits for Your Body » health>nutrition
- This farm in Australia grows six different fruits on one tree » video>viral

8.2 Experiment "markets" query User profiles:

User A:

- Food For Thought Prepares For Its Rock-A-Belly Fundraiser » foodanddrink>newstrends
- The best food truck in every state » foodanddrink>restaurantsandnews
- Here's where Colorado's top high school football recruits will play college football »sports>more_sports
- 10 things we learned during college football weekend » sports>football_ncaa
- Farmer's market on wheels delivers fresh, local produce to 'food deserts' » foodanddrink>newstrends
- The Best Grocery Shopping Tips of All Time » foodanddrink>tipsandtricks

- Order your farm-fresh local turkey soon if you want the best selection » foodanddrink>newstrends
- The indoor farming revolution » news>newsworld
- 50 fascinating facts about farming in America » foodanddrink>newstrends
- 10 best cities to visit in winter » travel>travelarticle

User F:

- Wealthy investors are bracing for a sharp stock sell-off in 2020 » finance>markets
- These Are the 7 Worst Diet Mistakes for Weight Loss (That Can Actually Cause Weight Gain!) » health>weightloss
- The F-Factor Diet Promises the "Secret to Permanent Weight Loss," But Is It Legit? » health>weightloss
- Giants hire Gabe Kapler to replace Bruce Bochy as manager » sports>baseball_mlb
- Giants are linked to Gabe Kapler in manager search. That's a problem » sports>baseball_mlb
- Bitcoin bounces back from a 5-month low » finance>markets
- 100 bold predictions for NBA season » sports>basketball_nba
- WeWork Wrestled With SEC Over Metric Just Before It Scrapped IPO » finance>finance-companies
- Software stocks were thought to be a can't-miss trade this year but now they are struggling » finance>finance-top-stocks
- 7 Dow Stocks That Billionaires Love » finance>finance-top-stocks

Bibliography

[1] Significant terms aggregation | Elasticsearch Guide [7.12] | Elastic.

[2] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 581–590, New York, NY, USA, 2007. Association for Computing Machinery.

[3] Michael D Ekstrand, John T Riedl, and Joseph A Konstan. *Collaborative filtering recommender systems*. Now Publishers Inc, 2011.

[4] Deepak Singh Tomar & Dhirendra Pratap Singh Jaytrilok Choudhary. An efficient hybrid user profile based web search personalization through semantic crawler, 2019.

[5] Chunyan Liang. User profile for personalized web search. In *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 3, pages 1847–1850. IEEE, 2011.

[6] Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. Trends in content-based recommendation, August 2020.

[7] Filip Radlinski Nicolaas Matthijs. Personalizing web search using long term browsing history, February 2011.

[8] Eero Sormunen. Liberal relevance criteria of trec -: Counting on negligible documents? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 324–330, New York, NY, USA, 2002. Association for Computing Machinery.

[9] Smitha Sriram, Xuehua Shen, and Chengxiang Zhai. A session-based search engine. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 492–493, New York, NY, USA, 2004. Association for Computing Machinery.

[10] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684, 2004.

[11] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 449–456, New York, NY, USA, 2005. Association for Computing Machinery.

[12] Jan Teichmann. How to build a Recommendation Engine quick and simple, August 2020.

[13] Mark Van Uden. Rocchio: Relevance feedback in learning classification algorithms, 2002.

[14] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online, July 2020. Association for Computational Linguistics.