1

Federated Search and Recommendation

Federated search systems emerged in late 1990 to support library users to simultaneously search multiple library sources and catalogues [ELG 12]. The first such systems were designed to collect the search criteria from the users, submit search queries to sources to be retrieved, wait for a response to arrive, and finally, filter and organize the query results to be send back to the users. With the maturity of Cloud Computing, Distributed Machine Learning (DML), big data, fog and edge computing technologies, new opportunities for exploiting computing resources on demand appeared, creating huge impacts on many sectors. However, implementing federated search systems is still a challenge for technical solutions, but also governance decisions and policies that need to be implemented and administered.

1.1. Introduction

The authors in [ORC 10] differentiate between Windows Search and Federated Search, as two types of search in Windows 7. Windows Search enables searching of the entire local system and a limited amount of external resources such as shared drives. Federated Search enables the simultaneous search of multiple sources, including document repositories, databases, remote repositories and external Web sites. To enable federated features for websites to be searched by the user, special search connectors need to be implemented and configured for each website [ORC 10]. For example, the selection-based search in MacOS Spotlight requires an index of all items and files that exist in

Chapter written by Dileepa JAYAKODY, Nirojan SELVANATHAN, Violeta DAMJANOVIC-BEHRENDT

the system to be created. One of the biggest issues with the federated search relates to resource accessibility, thus requiring one large index to be created from those multiple sources that constitute federation. Federated search allows for each source to be indexed locally and searched across many sources. Another issue of federated search is about the relevance of results that are obtained from a variety of sources. The relevance of results needs to be synchronized and comparable across sources, in order to merge the results in a meaningful way. In practice, even the results from the same source are not comparable when different search engines are used to retrieve these sources.

In this paper, we present our solution to implement a federated search and recommendation systems for digital platforms, as designed in the ongoing eFactory project. The aim of eFactory is to create a federation of several digital platforms, while overcoming the interoperability between different platforms, tools and services.

1.2. Literature review

The authors in [FEI 09] differentiate between federated search and metasearch. Metasearch engines crawl open websites or return results of all kinds, such as apps, webpages, contacts, and documents that are drawn from different sources. In contrast to metasearch, federated search systems focus on subscribed databases that are processed through an indexing mechanism enabling structured, in-depth, content-oriented retrieval [TAN 07]. A number of attempts to implement federated search happened from the 1990s onwards, e.g. MetaLib and SFX at Boston College [GER 02], MetaLib and SFX with Ex Libris at the Five Colleges Libraries in Massachusetts [MES 07], Georgia Library Learning Online (GALILEO) with its hybrid approach to federated search products [FAN 07], and more. Many interviews shown that the users were unsatisfied with federated search performance, due to limitations in database availability, search speed, retrieval precision, and result comprehensiveness [CAL 05]. Many users preferred simple interfaces and had little interest in advanced searching techniques. In recent years, several attempts to improve federated search were conducted. In 2004, the University of Nevada at Reno experimented with the Google Search Appliance to be used as a federated search utility. This required partnership between Google, the library and a test vendor, but it faced technical and interoperability challenges [TAY 06].

1.3. Problem statement

Federated search is a technique for simultaneous search of multiple data sources using one query and one search interface. The examples of federated search can be found in many sectors, facilitating search of thousands of products, each sorted into different categories, e.g. search of numerous websites serving various purposes and stakeholders, etc. [ALG 19] Majority of research on federated searching is focused on system performances and technical development, system usability testing, user interaction, interface customization, authentication, design, database communication protocol, system platform [DOR 04]. The two fundamental components of federated search systems include an index that is a reference to the data to be searched and a search algorithm. These two components interact using either search-time, index-time or a federated search interface [ALG 19].

– **Search-time merging** runs separate search algorithms on each data source that is searched. It uses multiple indices and aggregates the returned search results into a final list, which is then presented to the user. Obtaining the results can be slow, if the central search engine needs to wait until all of the local search engines have responded.

– **Index-time merging** requires a central index to be built for all of the data that need to be included in the search results. It requires one search engine and one index, and is faster than search-time merging. To aggregate data from multiple sources and formats into a single index, interoperability issues need to be addressed.

– **Federated search indexing** is an extension of the search-time merging method that requires a robust search solution to index different types of content in different indices and create the unified federated search interface.

1.4. Federated search and recommendation in eFactory

Our solution to a federated search and recommendation system design in eFactory, requires the following: firstly, to achieve a technical federation of digital platforms, we design a federated search capability. Secondly, we collect the results of federated search to support federated recommendations. Finally, federated search and recommendations are used to support advanced matchmaking and agile network creation mechanisms in the platform ecosystem. Federated search in eFactory is about searching for partners, products and services. The search criteria and the results are collected to support the recommendation processes, based on different techniques of information pattern matching, e.g. information retrieval and similarity matching techniques, which are both based on Machine Learning (ML) and data analysis. After recommendation algorithm identifies the most suitable partners, products and services, the user evaluates the results based on selected indicators, e.g. cost, reliability, quality, etc. In the final step, the user makes the decision and initiates a business transaction.

1.5. Design of a federated search and recommendation

In eFactory, we follow the index-time merge architectural approach to implement federated search (Figure 1). Such approach requires content from all base platforms to be acquired into a central index in the eFactory platform. The four base platforms in eFactory are NIMBLE, COMPOSITION, vf-OS and DIGICOR. The eFactory Data Spine is designed as an integration middleware, enabling toolsets and services from the base platforms to connect with each other and offer their capabilities to the users of these platforms. For the implementation of the Search Index, we use Apache Solr (https://lucene.apache.org/solr/) that enables distributed indexing, replication and load-balanced querying, automated failover and recovery, configuration, etc.

The major disadvantages of the index-time merge architecture are about the data connectors that need to be implemented to integrate different types of data sources. In addition, acquiring the content from different repositories and data sources requires considerable efforts; e.g., it needs to be done using scheduled read-only processes that need to be designed and implemented at the data integration layer. This also requires a decision about the frequency of the data ingestion into a central index. Data ingestion frequency needs to be configured hourly, daily or weekly, depending on the data velocity of the base platforms.

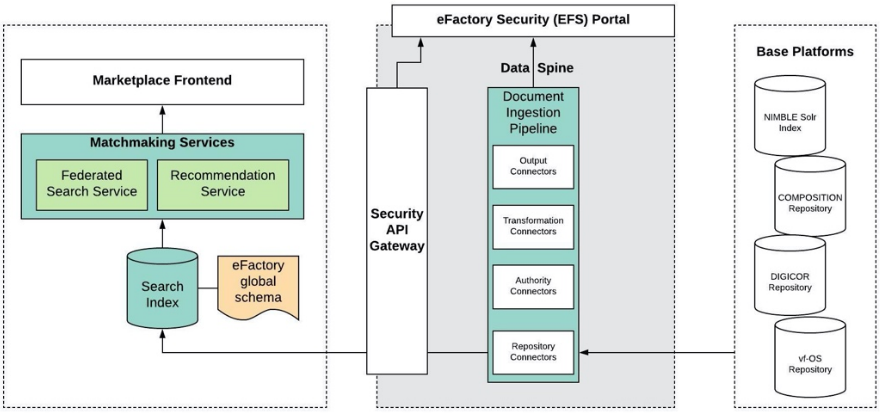

Figure 1.1. *The index-time merge architecture in eFactory*

Furthermore, the user interaction data in eFactory are stored using the User Activity Log Service, which listens to all user interaction events generated from the eFactory portal, e.g. purchased items, etc. The user interactions data are also stored using Solr Index, to support ML model creation. The outcomes of the matchmaking are presented to the users through an intuitive UI that is integrated in the eFactory portal.

1.6. Conclusion

Combining federated search, ML and recommendation systems in an innovative approach that can deliver better results for businesses by improving user experience, increasing user engagement through incentives and rewards mechanisms, boost conversion rates and operationalization of businesses via digital platforms. However, there are still many questions to be explored in the intersection of these fields, e.g.: Which recommendation algorithms are more suitable for federated digital environments? What is the minimum amount of data to run an effective recommendation system? How symbolic AI methods can compensate for the missing data and better support recommendation?

Acknowledgments

This research has been co-funded by the European Commission within the H2020 project NIMBLE, No. 723810 and the H2020 DT-ICT-07-2018-2019 project eFactory, No. 825075.

1.8. References

[ELG 12] ELGUINDI A.C., SCHMIDT, K., "Discovery systems, layers and tools, and the role of the electronic resources librarian". *Electronic Resource Management,* 2012, p. 109-139.

[TAY 06] TAYLOR, M., "Using the Google Search Appliance for federated searching*", Internet Reference Services Quarterly*, 10(3), 2006, p. 45–55.

[ORC 10] ORCHILLES, J., "Introduction to Windows 7", *Microsoft Windows 7 Administrator's Reference,* p. 1–91, doi:10.1016/b978-1-59749-561-5.00001-2, 2010.

[FEI 09] FEI, X., "Implementation of a Federated Search System: Resource Accessibility Issues", *Serials Review – SER REV. 35*, 2009, p. 235-241.

[TAN 07] TANG, R., HSIEH-YEE, I., ZHANG, S., "User Percep. of MetaLib Comb. Search: An Inv. of How Users Make Sense of Fed. Search," *Intern. Ref. Serv. Q. 12*, 2007, p. 211–236.

[DOR 04] DORNER, D.G., CURTIS, A.M., "A Comparative Review of Common User Interface Products," *Library Hi Tech 22,* no. 2, 2004, p. 182–197.

[GER 02] GERITTY, B., LYMAN, T., TALLENT, E., "Blurring Services and Resources: Boston College's Implementation of MetaLib and SFX", *Ref. Serv. Rev.30*, no.3, 2002, p. 229–241.

[MES 07] MESTRE, L.S., TURNER, C., LANG, B., MORGAN, B., "Do We Step Together, in the Same Direction, at the Same Time? How a Consortium Approached a Federated Search Implementation", *Internet Reference Services Quarterly 12*, no. 1/2, 2007, p. 111–132.

[FAN 07] FANCHER, L., "Wanted, Dead or Alive: Federated Searching for a Statewide Virtual Library", *Internet Reference Services Quarterly 12*, no. 1/2, 2007, p. 133–158.

[CAL 05] CALHOUN, K., "An Integrated Framework for Discovering Digital Library Collections", *Journal of Zhejiang University Science 6A*, no. 11, 2005, p. 1318–1326.

[ALG 19] ALGOLIA, "What is Federated Search?", URL: https://blog.algolia.com/what-is-federated-search/, 2019.