Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

Current Content Discovery for Module Teaching

by

Guanyuming He

Email: guanyuming.he24@imperial.ac.uk
GitHub username: esemsc-gh124
Repository: https://github.com/ese-ada-lovelace-2024/irp-gh124

Supervisors:

Sean O'Grady
Rhodri Nelson

8th July 2025

Abstract

TBD. **Keywords:** content discovery, information retrieval, LLM, search engines, case method, Business school,

1 Introduction

The project aims to improve personal information retrieval tools for a specific usage: retrieving latest business news for business school teaching. In this introduction, section 1.1 explains why it is important in business school teaching; section 1.2 gives a broad review of the history and current methods of information retrieval. Then, section 1.3 describes how I aim to improve the current status of personal information retrieval systems.

1.1 Project background

Since the emergence of the first Business schools in the late 19th century [46, 38], several distinct pedagogical teaching strategies have been applied. First institutionalized at Harvard Business School [18, 6] in the early 20th century, a method about teaching students with real world business cases (will be called *case method* in the rest of the thesis) has been found more effective and engaging [49, 4, 26] than many traditional, for example, big lecture based, teaching methods. Thus, it has found wide adoption across the world [47, 12].

Despite its aforementioned adoption and performance in business school teaching, the case method faces a significant constraint: the availability and collection of timely and relevant case material [42, 32]. In particular, Christensen identifies in his classical article that instructors have to conduct "extensive preparation" [8] for it. Another factor contributing to this constraint of case method is the ever-evolving business world and the necessity of the latest information: Clark argues that learned skill will lose value quickly in five years, and then it is critical for students to be up to date to remain relevant in the business world [9]; McFarlane emphasises the importance of updated cases, as otherwise students could be disengaged or discouraged [27].

1.2 Past advancements in information retrieval

The thesis summarizes the challenges in information retrieval into two general problems: (1) collect/gather information (2) identify/select desired information from gathered sources.

1.2.1 Problem (1)

During the past two centuries, a number of key developments have profoundly expanded one's capacity to gather information about the world. In the early 19th century, the transmission of information was still traditional — carried by person on paper or simply remembered. The invention of telegraph in the 1840s by Morse, Cornell, and Henry [40, 25], notably with Morse's first telegraph message, "*What hath God wrought?*" in 1844 [29], marked a paradigm shift. This technological innovation was considerably improved by Bell's telephone in 1876 [51, 14], enabling communication directly by human voice, instead of encoded Morse code.

Wired communication is critically constrained by geographical features on where the wires were laid. Around the late 19th century, Marconi's experiments with wireless telegraphy [13] and the first successful transatlantic signal in 1901 [3] introduced electromagnetic wave-based wireless communication, eventually accumulating into the world's first voice broadcast by radio in 1906

[45]. These milestones collectively redefined the temporal and spatial boundaries of information gathering.

In parallel to improving the weaknesses of wireless communication [17, 53, 1], reseachers then worked on a theory of information. Hartley observed a logritham pattern of information capacity [19] and then Shannon expanded on it to first define *bits* and *entropy*, giving a formal mathematical theory of information [43] in 1948. Meanwhile, engineers were experimenting with alternative modulation techniques, notably frequency modulation (FM), and the concepts were formalized in the 1930s [20].

1.2.2 Problem (2)

Although solutions to problem (1) had enormously improved during these times, substantial progress in problem (2) would have to wait until digital computers were invented in the 1940s [52], which were made to repeat tedious operations fast. Actually, the term *information retrieval (IR)* was not invented until Mooers coined it in 1950 [28]. The nature of information reveals two subproblems of problem (2): (2.1) how to process (extract useful information from) potentially unstructured data. (2.2) how to related human queries to the content a user actually wants.

According to Griffiths and King, early IR systems (in the USA) were mainly transferring human computing activity (bookkeeping in a library or database) into digital forms, by various sorting, indexing, and searching algorithms [16]. The form slowly went from “offline, batch” computing to “online, real-time” computing, and finally became “distributed, networked, and mass computing”, thanks to the Internet and various of its applications [16, 24]. Then commercial search engines emerged in the 1990s, opearting on the scale of the whole Internet [41, 30].

The aforementioned sorting, searching, and indexing algorithms were partial solutions to problem (2.1) and (2.2), with fundamental designs like inverted index [39, chap. 2], [56, sect. 2], boolean search [39, chap. 1], and various clever ideas such as spelling correction (stemming) [39, chap. 3.3], positional indexing [56, sect. 3]. Because the database can be extremely large, specific index construction [39, chap. 4], [56, sect. 5] compression [39, chap. 5] algorithms have appeared. Also, people saw ample opportunity to explore distributed computing, with the most notable model probably being MapReduce [10].

Nevertheless, these inventions were still mostly doing textual processing, and machines struggled to “understand” information. Thus, advanced and complex search queries are often needed to achieve desired performance [2, 34]. The research on making computers understand natural language (NLP), images and visuals (computer vision), etc. were consequently and gradually being integrated into search engines, since 2000, notably with Google’s Panda, Penguin, and Hummingbird algorithms that aim to understand web content better to rank them in searches [31].

One major breakthrough in NLP is Vaswani et al.’s transformer [48]. Building upon it, OpenAI created generative pre-trained transformers (GPTs) trained on massive amount of data [33], which, after a few iterations, evolved into a chatbot facing the general public, ChatGPT, and it was perhaps then when LLMs started to recieve massive amount of public attention, where LLM stands for large language model, language models that are *large* in trained data and parameters.

Although public attention is not the same as the importance of an idea, LLMs have undoubtedly entered and transformed many areas, including daily life, business and the industry, and research [11]. One strong appeal of LLMs is that they could process unstructured natural language input and generate natural language output in return with remarkable resemblence to

what a human would say [55, 23]. However, because of the inherent limit of neural networks, some argue that they could not formally reason about what they output [37, 22, 36]. Indeed, hallucination [21, 35] and other forms of distortion of facts, is a big problem of LLMs.

Despite the drawbacks of LLMs, they are currently the most successful tool in NLP, and naturally various attempts have been made to integrate LLMs with search engines [54, 44, 50]. Specifically, Xiong et al. proposes to categorize them into "LLM4Search" and "Search4LLM", where "A4B" means using A to improve B [54]. Here is where my thesis will build upon. Now, with the help with LLMs that can easily process frivolous natural language input, I plan to integrate them together to boost an individual's information retrieval further, especially in the area of business school teaching content discovery.

1.2.3 Gaps I aim to fill

The contribution I aim make is not about combining LLMs and IR, a direction numerous studies have explored. I identify these gaps in the state-of-the-art IR tools, with or without LLMs:

1. They are still semi-automatic, be it search engines or LLMs. One would have to give a prompt first and wait for answers. I plan to make a mostly automatic system where the IR process will be run automatically based on some configuration.
2. LLM's training data is often outdated. The information retrieved cannot reflect the new events happened very recently. Even if existing tools like RAG can make LLM search the Internet with a search engine, users often lack control over the content the search engine indexes and returns, and may often end up with less relevant or outdated data.

1.3 Goals

Based on the gaps, I plan to develop a software system with these goals:

1. By combining LLMs and search engines, compensate each other's weakeness in information retrieval.
   (a) The system shall improve search engines on prompt engineering to the extent that a user would only need to give a rough and vague natural language description of the target information to the system.
   (b) The system shall improve LLMs on result credibility and interpretibility. If pure LLMs lack the two properties, then combination with deterministic and well-designed ranking and searching algorithms in search engines will compensate for that.
2. Enhance the automation of the current general public information retrieval tools, e.g., Google search, ChatGPT, to the extent that a user would only need to occasionally configure the system, instead of giving search prompts each time.
3. Specially tailor the system to business teaching related information, aiming to gather information from authentic and reliable sources.
4. The system shall support up-to-date information retrieval. More precisely, the system shall support the user to specify a time range of information, and the user is allowed to set the end of the range to the current time.
5. The system could support user feedback and learning from it to provide information more close to one's need.

1.3.1 Not included in the goals

On the other hand, these possible targets are beyond the scope of my project:

1. Providing a method to rank business information. This is rather subjective and I believe is better left for the user to judge.
2. Enforcing strict security measures. Because the use group of the system is limited to business school teaching teams, enforcing output security or censorship of harmful or adult content is not a goal.

2 Methods

2.1 Architecture

Based on the goals, the system is designed to be a chain of tools invoking each other.

What directs the system is the user's configuration, which is expected to include

1. A time range of information to retrieve.
2. A natural language description of the types of information to retrieve.
3. How the retrieved information is sent to the user.
4. A frequency of running the tool.

Based on the configured running frequency, the tool runs, which

1. Use LLMs to translate the configured description and generate a list of search engine prompts.
2. The prompts are fed to search engines, with the configured time constraint.
3. The search results are gathered and processed. How the information is processed and filtered may rely on other AI systems like a rule-based expert system.
4. The processed results may be summarized by LLMs.
5. The results are sent to the users via the configured ways.
6. Optionally, the user gives feedback to the results. The tool can thus improve on the search engine prompts, the information processing, and LLM parameters, based on the feedback. The improvement algorithm may use something like recommendation algorithms. Then, the tool can iterate on the user's configuration automatically.

Figure 1 demonstrates my architecture design and the workflow of the system.

What directs the system is the user's configuration, which is expected to include a time range of information to retrieve, a natural language description of the types of information to retrieve, how the retrieved information is sent to the user, and a frequency of running the tool.

2.2 Implementation detail

Different from in project plan, here the details of the implementation, instead of the directions and plans, are provided.

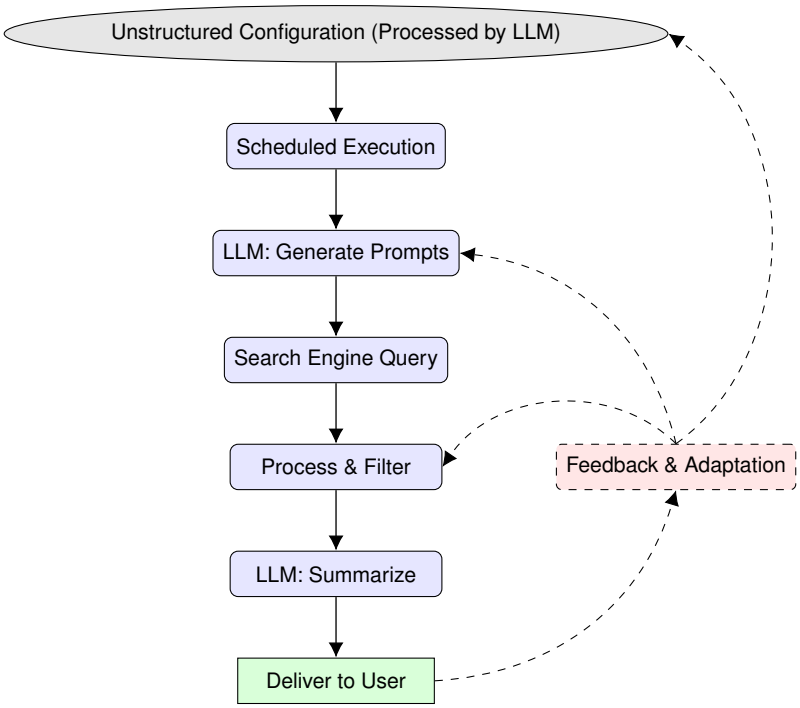

Figure 1: Architecture and operational flow of the software system

2.2.1 Configuration format

LLMs have demonstrated unprecedented ability to process unstructured data [7, 5]. Thus, it is desirable for the system to accept unstructured configuration from the user. The user may give a natural language description of the wanted information, give some examples, e.g., URLs, or simply feed the LLM the teaching slides and videos from previous classes.

Nevertheless, there are a few configurations that must be formatted.

1. A range of time for the information retrieved.
2. (Optional) A list of sources (e.g. websites) to always include or exclude.
3. The destination of the collected information (e.g. email addresses of teaching staff).

Of course, LLMs can still try to infer them from the given unstructured data, but they must be presented to the user for her to decide and correct, because these configurations are strict and a little error will have great effects.

Another important output from unstructured configuration is the search prompts or insturctions generated by the LLMs. As this output direct controls the information retrieved. LLMs for this task will be specifically tuned to ensure

- Diversity of results.
- A specific distribution of results. E.g., a more authentic website will contribute relatively more information.

2.2.2 Scheduled Execution

Based on the configured running frequency, the system will run every once a while automatically. The scheduling component must handle different frequencies (daily, weekly, monthly) and manage execution timing to avoid overwhelming external APIs or generating excessive costs.

This involves implementing a robust scheduling system that can handle failures, retries, and load balancing across multiple users.

2.2.3 Custom search engine

One problem of search engine APIs is that they have ungenerous search limits [15], allowing about 1000 queries per month, which is not enough for frequent testing and prototyping.

As a result, I develop a custom search engine that is much less powerful but much more specific, crawling only a limited subset of website domains, instead of the whole Internet.

The search engine is implemented in C++ with `libcurl` for url handling, `lexbor` for HTML parsing, and `xapian` for search indexing.

The search engine's algorithm is similar to the mainstream ones: Starting from initial urls, scrap each. And for every url found in each, scrap them too. However, one difference is that my engine will select a set of domains as first-class domains. Only those urls that match the domains will contribute to new scrap urls. That is, if a url does not match the domains, all urls found within it will not be processed.

2.2.4 Content Processing and Filtering

Search results will be gathered and processed through a multi-stage filtering system. This involves content deduplication, relevance scoring, source credibility assessment, and quality filtering. The processing pipeline will need to handle various content formats including articles, reports, videos, and social media posts.

The filtering system will combine rule-based approaches (such as keyword matching and source whitelisting) with machine learning techniques for relevance scoring and content classification. Natural language processing techniques will be employed to extract key information and assess content quality. The system will also need to handle different languages and cultural contexts in business information.

2.2.5 LLM-Based Summarization

Processed results will be summarized using LLMs with business-focused prompting strategies. The summarization component must balance comprehensiveness with conciseness, ensuring that key business insights are preserved while making information digestible for educators. This involves developing prompting strategies that can identify the most relevant information for case method teaching and present it in a structured format.

The summarization process may involve extractive techniques (selecting key sentences from original content) and abstractive techniques (generating new summaries). Different summarization styles may be needed for different types of content and user preferences.

2.2.6 Delivery and User Interface

Results shall be sent to users via configured methods, which may include email summaries, web dashboards, or integration with existing educational platforms. The delivery component shall handle different user preferences and ensure information is presented in a format suitable for educational use.

2.2.7 Feedback and Adaptation System

The system shall collect user feedback on result quality and relevance, using this information to improve future searches. This involves designing feedback mechanisms that capture both explicit user ratings and implicit behavioral signals. Possible algorithms include recommendation algorithms and other machine learning models.

2.3 Technical Considerations

The system will be a high-level program, which calls for simply programming languages like Python. On the other hand, the system may have specific performance requirements, especially for LLMs and other machine learning algorithms. The likely result is a combination of different programming languages and technical frameworks.

2.4 Evaluation and Validation

System effectiveness will be evaluated through multiple metrics including information retrieval precision and recall, user satisfaction surveys, time savings measurements, and comparison with manual search processes. Evaluation will involve collaboration with business educators to assess the practical utility of discovered content for case method teaching.

The system will be tested with different types of business information requests and across different time periods to ensure robust performance. A/B testing may be employed to compare different algorithmic approaches and user interface designs.

3 Results

4 Discussion

5 Conclusion

References

[1] E. F. W. Alexanderson. Transatlantic radio communication. *Proceedings of the American Institute of Electrical Engineers*, 38(10):1077–1093, 1919.

[2] Muhammad Bello Aliyu. Efficiency of boolean search strings for information retrieval. *American Journal of Engineering Research*, 6(11):216–222, 2017.

[3] W. J. G. Beynon. Marconi, radio waves, and the ionosphere. *Radio Science*, 10(7):657–664, 1975.

[4] Kevin M. Bonney. Case study teaching method improves student performance and perceptions of learning gains. *Journal of Microbiology & Biology Education*, 16(1):21–28, 2015.

[5] William Brach, Kristián Košťál, and Michal Ries. The effectiveness of large language models in transforming unstructured text to standardized formats, 2025.

[6] Todd Bridgman, Stephen Cummings, and Colm McLaughlin. The case method as invented tradition: revisiting harvard's history to reorient management education. In *Academy of Management Proceedings*, volume 2015, page 11637. Academy of Management Briarcliff Manor, NY 10510, 2015.

[7] Andry Castro, João Pinto, Luís Reino, Pavel Pipek, and César Capinha. Large language models overcome the challenges of unstructured text data in ecology. *Ecological Informatics*, 82:102742, 2024.

[8] C Roland Christensen. Teaching and the case method harvard business school. `https://www.hbs.edu/teaching/case-method/Pages/default.aspx`, 1987.

[9] Tom Clark. Case method in the digital age: how might new technologies shape experiential learning and real-life story telling? LSE Impact of Social Sciences, 2016.

[10] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[11] Qinxu Ding, Ding Ding, Yue Wang, Chong Guan, and Bosheng Ding. Unraveling the landscape of large language models: a systematic review and future perspectives. *Journal of Electronic Business & Digital Economics*, 3(1):3–19, 2023.

[12] Rebekka Eckhaus. Supporting the adoption of business case studies in esp instruction through technology. *Asian ESP Journal*, 14:280–281, 2018.

[13] Gabriele Falciasecca. Marconi's early experiments in wireless telegraphy, 1895. *IEEE Antennas and Propagation Magazine*, 52(6):220–221, 2010.

[14] J.E. Flood. Alexander graham bell and the invention of the telephone. *Electronics and Power*, 22:159–162, 1976.

[15] Google Developers. Usage limits, 2025. Accessed: 2025-06-09.

[16] J.-M. Griffiths and D.W. King. Us information retrieval system evolution and evaluation (1945-1975). *IEEE Annals of the History of Computing*, 24(3):35–55, 2002.

[17] Brian N. Hall. The british army and wireless communication, 1896–1918. *War in History*, 19(3):290–321, 2012.

[18] John S Hammond. *Learning by the case method*. Harvard Business School Boston, MA, 1980.

[19] R. V. L. Hartley. Transmission of information. *Bell System Technical Journal*, 7(3):535–563, 1928.

[20] Raymond A. Heising. Modulation methods. *Proceedings of the IRE*, 50(5):896–901, 1962.

[21] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2), January 2025.

[22] Andrei Kucharavy. Fundamental limitations of generative llms. In *Large Language Models in Cybersecurity: Threats, Exposure and Mitigation*, pages 55–64. Springer Nature Switzerland Cham, 2024.

[23] Pranjal Kumar. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260, 2024.

[24] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. A brief history of the internet. *SIGCOMM Comput. Commun. Rev.*, 39(5):22–31, October 2009.

[25] Kenneth B Lifshitz. *Makers of the Telegraph: Samuel Morse, Ezra Cornell and Joseph Henry*. McFarland, 2017.

[26] Alberto Lusoli. Teaching business as business: The role of the case method in the constitution of management as a science-based profession. *Journal of Management History*, 26(2):277–290, 2020.

[27] Donovan A McFarlane. Guidelines for using case studies in the teaching-learning process. *College Quarterly*, 18(1):n1, 2015.

[28] Calvin Mooers. The theory of digital handing of non-numerical information and its implications to machine economics. In *Proceedings of the meeting of the Association for Computing Machinery at Rutgers University*, 1950.

[29] Samuel Finley Breese Morse. First telegraph message. Retrieved from the Library of Congress, `https://www.loc.gov/item/mcc.019/`, May 1844.

[30] Rhoda Okunev. *History of the Internet, Search Engines, and More*, pages 9–16. Apress, Berkeley, CA, 2023.

[31] Akshita Patil, Jayesh Pamnani, and Dipti Pawade. Comparative study of google search engine optimization algorithms: Panda, penguin and hummingbird. In *2021 6th International Conference for Convergence in Technology (I2CT)*, pages 1–5, 2021.

[32] Sandeep Puri. Effective learning through the case method. *Innovations in Education and Teaching International*, 59(2):161–171, 2022.

[33] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[34] Shivangi Raman, Vijay Kumar Chaurasiya, and S. Venkatesan. Performance comparison of various information retrieval models used in search engines. In *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–4, 2012.

[35] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models, 2023.

[36] Walid S Saba. Stochastic llms do not understand language: towards symbolic, explainable and ontologically based llms. In *International conference on conceptual modeling*, pages 3–19. Springer, 2023.

[37] Erin Sanu, T Keerthi Amudaa, Prasiddha Bhat, Guduru Dinesh, Apoorva Uday Kumar Chate, and Ramakanth Kumar P. Limitations of large language models. In *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, pages 1–6, 2024.

[38] Steven A Sass. *The pragmatic imagination: A history of the Wharton School, 1881-1981*. University of Pennsylvania Press, 2016.

[39] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[40] Mischa Schwartz and David Hochfelder. Two controversies in the early history of the telegraph. *IEEE Communications Magazine*, 48(2):28–32, 2010.

[41] Tom Seymour, Dean Frantsvog, Satheesh Kumar, et al. History of search engines. *International Journal of Management & Information Systems (IJMIS)*, 15(4):47–58, 2011.

[42] Binod Shah. Case method of teaching in management education. *Research Gate*, 2019.

[43] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001.

[44] Xiang Shi, Jiawei Liu, Yinpeng Liu, Qikai Cheng, and Wei Lu. Know where to go: Make llm a relevant, responsible, and trustworthy searchers. *Decision Support Systems*, 188:114354, 2025.

[45] Elliot N. Sivowitch. A technological survey of broadcasting's "pre-history," 1876–1920. *Journal of Broadcasting*, 15(1):1–20, 1970.

[46] John-Christopher Spender. The business school in america: a century goes by. *The future of business schools: Scenarios and strategies for*, pages 9–18, 2020.

[47] Carlos JO Trejo-Pech and Susan White. The use of case studies in undergraduate business administration. *Revista de administração de empresas*, 57:342–356, 2017.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[49] Klára Vítečková, Tobias Cramer, Matthias Pilz, Janine Tögel, Sascha Albers, Steven van den Oord, and Tomasz Rachwał. Case studies in business education: an investigation of a learner-friendly approach. *Journal of International Education in Business*, 18(2):149–176, 2025.

[50] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. Freshllms: Refreshing large language models with search engine augmentation, 2023.

[51] Thomas A. Watson. How bell invented the telephone. *Proceedings of the American Institute of Electrical Engineers*, 34(8):1503–1513, 1915.

[52] Martin H. Weik. The eniac story. *Ordnance*, 45(244):571–575, 1961.

[53] JONATHAN REED WINKLER. Telecommunications in world war i. *Proceedings of the American Philosophical Society*, 159(2):162–168, 2015.

[54] Haoyi Xiong, Jiang Bian, Yuchen Li, Xuhong Li, Mengnan Du, Shuaiqiang Wang, Dawei Yin, and Sumi Helal. When search engine services meet large language models: Visions and challenges. *IEEE Transactions on Services Computing*, 17(6):4558–4577, 2024.

[55] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data*, 18(6), April 2024.

[56] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM Comput. Surv.*, 38(2):6–es, July 2006.