Large Language Models and Recommendation Systems: A Proof-of-Concept Study on Public Procurements

Roberto Nai[1][0000-0003-4031-5376], Emilio Sulis[1][0000-0003-1746-3733], Ishrat Fatima[1][0000-0002-3807-410], and Rosa Meo[1][0000-0002-0434-4850]

Department of Computer Science - University of Turin

Abstract. In legal informatics research, decision support systems can be a valuable tool for practitioners facing a growing volume of data. An expert system based on information retrieval and a recommender system can benefit from the application of Large Language Models to improve the quality of results. This paper proposes a general framework based on Retrieval-Augmented Generation for addressing integrated recommendation systems with generative models in public procurement. Moreover, we addressed a practical application by adopting real datasets in the legal domain. To illustrate the feasibility of the approach, a proof-of-concept has been presented in the context of public procurement management within an Italian case study. The study and evaluation phases have been supervised by domain experts in the legal field to ensure robust analysis and relevance.

Keywords: Large Language Models applications · Information Retrevial · Public Procurements

1 Introduction

An area of great interest in legal informatics research concerns decision-support systems. Most activities in law work are normally carried out by domain experts, who are faced with an increasing amount of data and possible risks of incurring errors, such as not reaching all documents of interest.

In order to facilitate the work of legal practitioners, expert systems have been proposed by research on Information Retrieval (IR) and Recommendation Systems (RS). Such systems facilitate the work of practitioners by leveraging computational capabilities, big data management, and data mining. In this context, the most used methods concern knowledge extraction and identification of documents from large legal datasets. In recent years, the impact of Large Language Models (LLMs) in Natural Language Processing (NLP) has also grown in the direction of exploiting their potential to improve RSs [27]. The adoption of RSs in the context of legal informatics represents a significant challenge to improve the efficiency of legal work organizations.

In this paper, we focus specifically on the problem of extracting similar legal documents to facilitate the instruction of administrative process. In particular, legal experts often have to track down the referenced legislation given a specific field manually, relying on experience and knowledge gained in prior work. This method, however, is relatively time-consuming and obviously not without risk of omissions and gaps. Increasing computational capabilities allow the volume of regulations to be managed to try to get to all similar documents quickly and accurately. A legal area of great interest for the impact on public administration is the public procurement process. The creation of new tenders often requires consideration of previous tenders already published in the specific target area. Domain experts confirm that they have to spend a lot of time to achieve the broadest and most correct knowledge possible without any support tool.

To address this gap, we focus on an application task of a real-world problem, proposing a general framework for the suggestion of similar public procurements adopting the latest generative technologies. In particular, we propose the adoption of Retrieval-Augmented Generation (RAG) [9], i.e. an approach in Natural Language Processing (NLP) that combines information retrieval techniques with text generation. A RAG-based system takes advantage of information retrieval results to inform and enhance text generation, thereby enabling the production of more accurate and relevant responses to user queries.

Through a proof-of-concept study related to public procurement management in an Italian case study, we demonstrate the feasibility of the proposed approach by presenting the initial results. In addition, we add an analysis of execution time, which is a relevant issue in LLMs. All phases of the study and evaluation have been supervised by domain experts in the legal field.

In the remainder of the paper, Section 2 concerns the description of the background with related work and case study. Section 3 describes the methodological framework, while Section 4 details the initial output of the proof-of-concept study. Section 5 concludes the paper.

2 Background

2.1 Related Work

The typical goal of RSs is to suggest a ranking list of items on a specific topic [19]. Some recent works have explored the use of LLMs in the field of RS [6,12]. The objective of providing support in searching for relevant textual content and making decisions is the basis of conversational recommendation systems [16].

The combination of retrieval-based and generation-based models has been proven significant to enhance the ability of RSs to provide relevant suggestions [5]. In [4], authors studied the adoption of ChatGPT in RSs by obtaining good performance in recommendations and a good understanding of item similarity with four datasets (e.g., the widely-adopted MovieLens benchmark dataset). Similar results have been obtained by [11], which explored a solution to the ranking task by designing the prompting template and conducting extensive experiments on two widely-used datasets. In our work, we propose a pipeline similar to the previous two papers, although we use a legal dataset as well as a dataset derived *ad hoc* for the current research. Accordingly to [26], adopting benchmark datasets can lead to problems in evaluating RSs. In fact, the following two potential problems may arise: i) Such datasets are relatively small in scale compared to real-world industrial datasets and may not fully reflect the recommendation capability of LLMs; ii) The items in these datasets may have related information that appeared in the pre-training data of LLMs.

In the legal field, RSs have been proposed to address different kinds of tasks. For instance, a support system offers sources of law suggestions with a prototype in the context of Immigration Law [25], while [10] focused on companies' sustainability reports to assists auditors and financial investors. In [17], the authors explored using RS in the legal field with SBERT and LaBSE BERT [7] . Frameworks for legal recommendations may exploit Bidirectional Encoder Representations from Transformers (BERT) and Skip-Recurrent Neural Network (Skip-RNN) models [29]. In [22], the authors proposed a system designed to extract legal arguments from a user's brief and recommend case law opinions of high relevance. These systems have been applied to the legal domain without using LLMs. Our work investigates the combination of RSs and generative models in the legal field.

2.2 Case Study

The Italian agency The current research is based on a legal dataset concerning the public procurement process in Italy. The whole dataset has been retrieved and collected from the National Authority for Anti-Corruption (ANAC), which is an autonomous Italian administrative agency tasked with preventing corruption in the Italian public sector. The agency tracks procurement notices issued by national public administrations in an Open Data catalogue. The main information provided by the agency concerns the description of contract authorities (the public administrations that initiate the contract) and economic operators (the contractors that win the contract). In particular, the agency also displays notices online, i.e. a website[1] hosts data on public procurement cases collected since 2007. In addition, only data on public procurements with a value greater than 40,000 euros are considered.

The structure of the dataset The complete dataset has been organized in three tables. The first table considered by the ANAC dataset is `Procurement`, which records the procurement announced across the different Italian regions. Each procurement is distinguished by an alphanumerical identifier *CIG* (the key ID value) and a *subject*, e.g. a textual description of it. Procurements are categorised by a *type* (Works, Supplies, Services), a *sector* (Ordinary or Extraordinary), a *Common Procurement Vocabulary* (CPV) code to specify the type of goods acquired [1], and a *selection criterion* for determining the winning contractors. A second table includes the relevant information of the `Contract Authority`, which records the identifier *ID*, the *name*, and their *type* (e.g., municipality, hospital,

[1] `https://dati.anticorruzione.it/opendata`

school, etc.) of the Public Administrations (PA) that created the procurement. A third table considers the detail of each `Economic Operator` by including the identifier of the records the *ID* and *name* of the contractors that win the procurement. The full ANAC dataset collected from different sections of the main website has been made publicly accessible in the repository of the current work[2].

Dataset filtering Following the suggestion of domain experts involved in the case study, we focused the attention on a significant subset of the dataset. In particular, we considered the public procurement process spanning from 2016 to 2022 in Northern Italy, i.e. 8 regions[3]. The reason for this choice involves both qualitative and quantitative aspects related to the number of procurements and their heterogeneity. In fact, the period 2016 - 2022 aligns well with the operational duration of the Italian public procurement code, which was enacted in April 2016 and abrogated on 31 March 2023[4]; this enabled us to gather consistent data. The total number of procurements extracted from the ANAC catalogue for the 8 regions of Northern Italy from 2016 to 2022 is 992,137, constituting the dataset on which this paper is based.

3 Experimental Setup

Since our goal can be classified as a *question answering* task [3], we based our method on the Retrieval-Augmented Generation (RAG) approach. RAG is a novel approach in the field of natural language processing (NLP) that combines the strengths of information retrieval (IR) and advanced language generation technologies. This technique is designed to enhance the capability of large language models in generating responses that are not only relevant but also contextually accurate [18]. The core of a RAG system is the integration of two components: a *retriever* and a *generator*; the retriever is tasked with fetching relevant pieces of information or documents from a database or corpus based on a given query, while the generator, which is typically an LLM, has the role to synthesise the retrieved information and the original query to produce a coherent response [18]. We refer to [15] for a deep understanding of the RAG architecture. Thus, by augmenting LLM's generative capabilities with information retrieval accuracy, RAG systems offer a useful tool for a wide range of applications, including response systems, which is the focus of our research.

General framework The general framework of the present research includes several steps, which can be summarized by Figure 1. The proposed pipeline starts with the acquisition of the dataset. In our proof-of-concept, this step involves retrieving the relevant data from the online database of the agency involved (ANAC). Data processing and filtering allow us to obtain the texts of interest

[2] Repository of the current research: `https://github.com/roberto-nai/NLDB2024`

[3] The regions involved are: Aosta Valley, Piedmont, Liguria, Lombardy, Emilia-Romagna, Veneto, Friuli-Venezia Giulia, Trentino-Alto Adige

[4] Legislative Decree n. 50 of 2016, superseded by Legislative Decree n. 36 of 2023

and the metadata necessary to set up the system. Subsequently, the embedding phase transforms the texts into vectors.

The following steps concern contextual research, which includes the user formulating a question to query the reference data. In the example in the Figure, a public official (user) is looking for procurements related to the purchase of syringes for public hospitals. Based on the query[5], the search in the database is carried out by the question system that transforms the textual query in an embedding and, following the best similarity score, retrieves and returns the texts (procurement object) to the user in the form of an answer. Finally, the system proposes a ranking of the most relevant documents that are returned to the user, which will assess their actual relevance.

Processing steps The initial step of our framework considers the collection and pre-processing of relevant data, which are appropriately filtered to serve as the basis for the RAG system. As the ANAC's Open Data catalogue is divided into more than 200 CSV files, a script has been written to automatically download them all; the Python source code is publicly available in the above-mentioned repository of the current research. The Open Data, appropriately filtered by region and year (see Section 2.2), were saved in the underlying database of the RAG system. This process of indexing the dataset into the database is enhanced by *embeddings*, which can be used to transform the texts into a high-dimensional space, making the retrieval process more efficient due to semantic similarity. Following the retriever's configuration, the focus shifts to the generator component, which relies on LLM. This step is crucial for crafting coherent and contextually appropriate responses, drawing on both the user's input and the texts fetched by the database as answers. Integrating the retriever and generator marks a crucial step in the RAG pipeline. It combines the information fetched by the database with LLMs' natural language processing prowess. The final steps involve fine-tuning and evaluating the RAG to ensure it meets specific performance criteria; to do this, the proposed results were evaluated in combination with their position and a user relevance score based on the metadata of the recommended tender (procurement type, sector, PA type, etc. as described in Section 2.2).

Evaluation To ensure the effectiveness and relevance of our approach, we evaluated the system using the *Normalized Discounted Cumulative Gain* (NDCG) metric [24]. This evaluation method allows quantitatively assessing how well the system ranks recommended items in terms of their relevance to users. Two key pieces of information were used to evaluate the NDCG: the ordered list of recommended items and the relevance values for each item. The recommended items are ordered according to their relevance in the system (using the typical cosine similarity measure [13]), and a relevance value is entered for each item to reflect its importance or usefulness to the user.

[5] Examples of query arguments (in Italian) can be found in the repository of the current search: `https://github.com/roberto-nai/NLDB2024`

To calculate NDCG, we first compute the *Discounted Cumulative Gain* (DCG) using the formula $DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$, where rel_i is the relevance of the item at position i, and k is the number of top items considered. The *Ideal DCG* (IDCG) is the maximum possible DCG obtained by arranging the items in the perfect order of relevance. NDCG is then the ratio of the actual DCG to the IDCG, normalized in the range between 0 and 1, calculated as $NDCG@k = \frac{DCG@k}{IDCG@k}$. This metric measures how effectively the recommendation system ranks the different items according to their relevance.

Technology In terms of technology, the LangChain framework[6] has been used to implement the RAG system; LangChain facilitates the connection between information retrieval skills and the generative capacity of LLMs by reducing coding time. We opted for the adoption of Elasticsearch [8], which is the most used search engine software that also supports dense vectors [28] (a generalisation of embeddings generated via LLM), according to the DB-Engines Ranking of Search Engines [2] as of March 2024. The calculation of the *NDCG@k* index can be performed using the Python library scikit-learn[7]. As LLMs rapidly scale up to gigantic models [21], their deployment as the core of RAG systems has become hard. This is mainly due to the fact that LLMs are computationally expensive, and implementing them on standard hardware is often not feasible. To address this issue, numerous companies, such as Cohere[8] and OpenAI[9], have initiated providing access to their proprietary LLMs via a range of APIs. The proposed framework adopts the Cohere and OpenAI APIs, which support bulk calls of up to 96 and 2048 texts per call, respectively. This is relevant when the amount of data to be transformed into embedding is important. Cohere offers the possibility of generating embeddings with two dimensions: `embed-multilingual-v3.0` (1024) and `embed-multilingual-light-v3.0` (384), while OpenAI offers three embeddings: `text-embedding-ada-002` (1536), `text-embedding-3-small` (1536) and `text-embedding-3-large` (3072).

For the indexing phase, each of the five models mentioned above has been tested. Afterwards, 10 queries were executed for each model used in the RAG, and the top 10 responses were compared with respect to the NDCG. In addition to the RAG system, queries were also executed on a textual version of the database, used as a performance baseline; again, the top 10 responses obtained from the database algorithm (based on BM25 [20]) have been evaluated against the NDCG.

To avoid computational waste, a small cache system was implemented to save user query embeddings and prevent recomputation after crashes or freezes. A native LangChain functionality stores embeddings on the local file system, minimizing API calls. Exploring other cache system solutions is beyond the scope of this research.

[6] `https://www.langchain.com`
[7] `https://scikit-learn.org`
[8] `https://cohere.com/`
[9] `https://openai.com/`

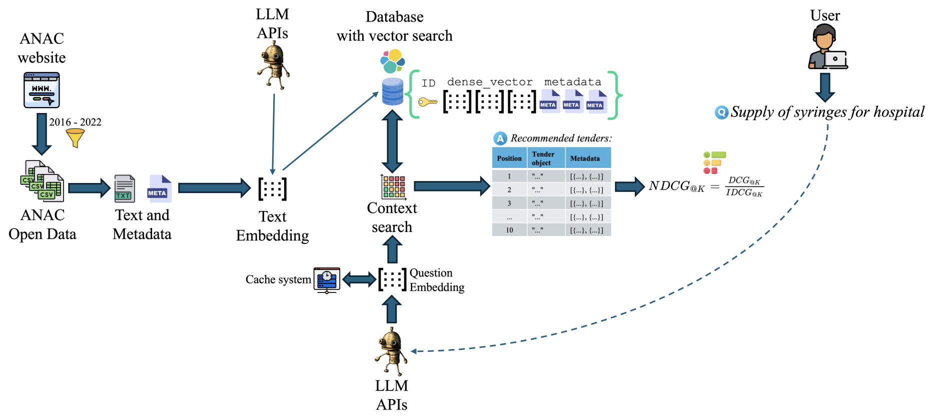

Fig. 1. Pipeline adopted for the question answering task with LLM; the workflow starts from the left with the collection of data, then the creation of the embeddings, the saving in the database and then the presence of a question (query) and the search for a contextual response. Full size image available at `https://github.com/roberto-nai/NLDB2024`.

4 Overall Results

4.1 LLM results

Table 1 presents a comparative analysis of retrieval model performance using the NDCG metric at a cutoff of 10. The columns in the table represent the results obtained in the various implementations of the RAG: BM25 using text mode; Cohere$_{\texttt{light}}$ and Cohere$_{\texttt{large}}$ using respectively `embed-multilingual-light-v3.0` and `embed-multilingual-v3.0`; OpenAI$_{\texttt{ada-002}}$ using `text-embedding-ada-002`; OpenAI$_{\texttt{3-small}}$ using `text-embedding-3-small`; OpenAI$_{\texttt{3-large}}$ using `text-embedding-3-large`.

An initial observation indicates that OpenAI$_{\texttt{3-large}}$ model consistently outperforms the other models with the highest average NDCG@10 score of 0.848, highlighting its superior ability to rank relevant documents at the top of the search results. OpenAI$_{\texttt{ada-002}}$ model has the second-highest average score of 0.837, underlined in the table, suggesting that while it is less effective than the OpenAI$_{\texttt{3-large}}$ model, it still significantly improves upon the baseline BM25 and the OpenAI$_{\texttt{3-small}}$ model. As far as Cohere is concerned, neither model performed better than the OpenAI models in our task; finally, it can be seen that Cohere$_{\texttt{light}}$ model (which has the smallest dimension) does not outperform BM25; this result was also highlighted in [14].

In any case, while historically robust, the BM25 model shows its limitations in this comparison, especially against the more sophisticated models provided by LLMs.

Overall, the results imply that more advanced and larger models tend to yield better relevance ranking performance, though the degree of improvement can vary by query and models adopted. This suggests a nuanced landscape where model selection for information retrieval tasks should consider the trade-offs between computational resources and the incremental gains in retrieval effectiveness.

Table 1. Results (NDCG10) for every experiment. The best score is marked in **bold** and the second best is underlined.

Query	**BM25**	**Cohere$_{\texttt{light}}$**	**Cohere$_{\texttt{large}}$**	**OpenAI$_{\texttt{ada-002}}$**	**OpenAI$_{\texttt{3-small}}$**	**OpenAI$_{\texttt{3-large}}$**
1	0.762	0.775	0,879	0.789	0.833	0.832
2	0.803	0.762	0.842	0.822	0.814	0.837
3	0.749	0.661	0.845	0.785	0.806	0.701
4	0.732	0.901	0.774	0.909	0.925	0.968
5	0.775	0.631	0.817	0.836	0.739	0.799
6	0.900	0.811	0.772	0.846	0.813	0.854
7	0.918	0.816	0.844	0.825	0.833	0.788
8	0.768	0.912	0.795	0.909	0.844	0.916
9	0.823	0.778	0.786	0.770	0.827	0.905
10	0.824	0.872	0.877	0.881	0.843	0.880
Avg. NDCG10	0.805	0.792	0.823	<u>0.837</u>	0.827	**0.848**

4.2 Evaluation

The evaluation of the proposed approach has been carried out qualitatively by legal domain experts, who were involved in all stages of the design process. In our proof-of-concept, the tool has been prepared based on a specific need of legal practitioners, who need to have specific support in the preliminary stage of the legal process a picture of the legislation as complete as possible related to a legal sector or topic. In particular, according to the analysis of the results proposed by the implemented automated system, the system worked properly by offering a relevant and accurate suggestion. The tool has shown to be reliable and, above all, fast, compared to the typical pipeline that involves manual searching, often on paper documents, which is also likely to be incomplete and partial. Domain experts appreciate the possibility of having such a tool to support their daily work, although they confirm the great importance of human control over the proposed results. A quantitative evaluation has not been implemented, given the limited number of cases of this proof-of-concept. It is our intention to propose an indicator-based evaluation in a future extended version of this work.

4.3 Timing of the experiments

Our observations show that API interactions are mostly stable, with occasional service request errors. Each API call consistently took around 300ms across both companies tested. Service errors and latency times can be influenced by factors such as input length and server workload, which vary over time. The `text-embedding-3-large` model took the most time for both creating embeddings and computing results. The total size of the database with embeddings is about 120 GB. All the computations were carried out by an ARM architecture-based chip with 3.2 GHz speed (10-Core CPU / 24-core GPU) and 32 GB of RAM.

5 Conclusions

This paper details creating a decision support system for legal practitioners to manage growing data volumes. The approach combined a recommendation system with generative model technologies in a proof-of-concept study using a real public procurement dataset. The results demonstrated the application's feasibility in the legal field. Although the implementation demands significant computational effort, the performance benefits are highly valuable to domain experts.

The conclusion highlights a specific request from domain experts for an easy-to-manage application using commonly used equipment. The proposed solution is to develop a web browser-based application for managing results, enabling continued testing and feedback for improvement. Future work includes implementing a service using Flask and ReactJS to facilitate web-based tool usage.

As for other future work, we want to introduce the use of further evaluation metrics in addition to the NDCG and explore in more detail a qualitative evaluation by domain experts, adding more queries to submit to the system. On a technological level, we plan to compare the proposed RAG framework with other state-of-the-art ones, such as Fast-RAG[10]. Regarding LLMs, we want to explore the use of open LLMs such as LLaMA [23] instead of proprietary tools (as are Cohere and OpenAI).

Finally, given that the general framework presented here in the legal field and the related proof-of-concept results have shown the tool's usefulness, the application can be tested in other domains, such as healthcare.

5.1 Acknowledgements

The authors thank the Department of Management for supporting this research with domain experts in public tenders and, in particular, Professor Gabriella Margherita Racca and Dr. Francesco Gorgerino for contributing data analysis.

References

1. CPV codes and nomenclatures. https://simap.ted.europa.eu/web/simap/cpv (2022), visited: 2022-12-01

[10] https://github.com/intellabs/FastRAG/

2. DB-engines ranking of search engines. `https://db-engines.com/en/ranking/search+engine` (2023), visited: 2023-02-01
3. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051 (2017)
4. Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., Sun, Z., Zhang, X., Xu, J.: Uncovering chatgpt's capabilities in recommender systems. In: Zhang, J., Chen, L., Berkovsky, S., Zhang, M., Noia, T.D., Basilico, J., Pizzato, L., Song, Y. (eds.) Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023. pp. 1126–1132. ACM (2023). https://doi.org/10.1145/3604915.3610646
5. Di Palma, D.: Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In: Zhang, J., Chen, L., Berkovsky, S., Zhang, M., Noia, T.D., Basilico, J., Pizzato, L., Song, Y. (eds.) Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023. pp. 1369–1373. ACM (2023). https://doi.org/10.1145/3604915.3608889
6. Fan, W., Zhao, Z., Li, J., Liu, Y., Mei, X., Wang, Y., Tang, J., Li, Q.: Recommender systems in the era of large language models (llms). CoRR **abs/2307.02046** (2023). https://doi.org/10.48550/ARXIV.2307.02046
7. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. arXiv preprint arXiv:2007.01852 (2020)
8. Gormley, C., Tong, Z.: Elasticsearch: the definitive guide: a distributed real-time search and analytics engine. " O'Reilly Media, Inc." (2015)
9. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: International conference on machine learning. pp. 3929–3938. PMLR (2020)
10. Hillebrand, L., Pielka, M., Leonhard, D., Deußer, T., Khameneh, T.D., Kliem, B., Loitz, R., Morad, M., Temath, C., Bell, T., Stenzel, R., Sifa, R.: sustain.ai: a recommender system to analyze sustainability reports. In: Grabmair, M., Andrade, F., Novais, P. (eds.) Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, ICAIL 2023, Braga, Portugal, June 19-23, 2023. pp. 412–416. ACM (2023). https://doi.org/10.1145/3594536.3595131
11. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems. In: Goharian, N., Tonellotto, N., He, Y., Lipani, A., McDonald, G., Macdonald, C., Ounis, I. (eds.) Advances in Information Retrieval. pp. 364–381. Springer Nature Switzerland, Cham (2024)
12. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J.J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems. In: Goharian, N., Tonellotto, N., He, Y., Lipani, A., McDonald, G., Macdonald, C., Ounis, I. (eds.) Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14609, pp. 364–381. Springer (2024). https://doi.org/10.1007/978-3-031-56060-6_24
13. Jurafsky, D., Martin, J.H.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition. Prentice Hall series in artificial intelligence, Prentice Hall, Pearson Education International (2009), `https://www.worldcat.org/oclc/315913020`
14. Kamalloo, E., Zhang, X., Ogundepo, O., Thakur, N., Alfonso-Hermelo, D., Rezagholizadeh, M., Lin, J.: Evaluating embedding apis for information retrieval. arXiv preprint arXiv:2305.06300 (2023)

15. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
16. Manzoor, A., Jannach, D.: Towards retrieval-based conversational recommendation. Inf. Syst. **109**, 102083 (2022). https://doi.org/10.1016/J.IS.2022.102083
17. Nai, R., Meo, R., Morina, G., Pasteris, P.: Public tenders, complaints, machine learning and recommender systems: a case study in public administration. Computer Law & Security Review **51**, 105887 (2023)
18. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
19. Resnick, P., Varian, H.R.: Recommender systems - introduction to the special section. Commun. ACM **40**(3), 56–58 (1997). https://doi.org/10.1145/245108.245121
20. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval **3**(4), 333–389 (2009)
21. Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., Zhang, E., Child, R., Aminabadi, R.Y., Bernauer, J., Song, X., Shoeybi, M., He, Y., Houston, M., Tiwary, S., Catanzaro, B.: Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model (2022)
22. Thomas, M., Vacek, T., Shuai, X., Liao, W., Sanchez, G., Sethia, P., Teo, D., Madan, K., Custis, T.: Quick check: A legal research recommendation system. In: NLLP@KDD (2020)
23. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
24. Wang, Y., Wang, L., Li, Y., He, D., Liu, T., Chen, W.: A theoretical analysis of NDCG type ranking measures. CoRR **abs/1304.6480** (2013), `http://arxiv.org/abs/1304.6480`
25. Winkels, R., Boer, A., Vredebregt, B., van Someren, A.: Towards a legal recommender system. In: Hoekstra, R. (ed.) Legal Knowledge and Information Systems - JURIX 2014: The Twenty-Seventh Annual Conference, Jagiellonian University, Krakow, Poland, 10-12 December 2014. Frontiers in Artificial Intelligence and Applications, vol. 271, pp. 169–178. IOS Press (2014). https://doi.org/10.3233/978-1-61499-468-8-169
26. Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., Xiong, H., Chen, E.: A survey on large language models for recommendation. CoRR **abs/2305.19860** (2023). https://doi.org/10.48550/ARXIV.2305.19860
27. Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., et al.: A survey on large language models for recommendation. arXiv preprint arXiv:2305.19860 (2023)
28. Zhao, W.X., Liu, J., Ren, R., Wen, J.R.: Dense text retrieval based on pretrained language models: A survey. ACM Transactions on Information Systems **42**(4), 1–60 (2024)
29. Zheng, M., Liu, B., Sun, L.: Lawrec: Automatic recommendation of legal provisions based on legal text analysis. Computational Intelligence and Neuroscience **2022** (2022)