
UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

# Implementation of an LLM-Based Chatbot for Academic Venue Recommendations by exploiting RAG

Diploma Thesis

**Efthymia Koustika**

**Supervisor:** Eleni Tousidou

September 2025


UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

# Implementation of an LLM-Based Chatbot for Academic Venue Recommendations by exploiting RAG

Diploma Thesis

**Efthymia Koustika**

**Supervisor:** Eleni Tousidou

September 2025


ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

# Δημιουργία ενός Chatbot βασισμένου σε LLM για Σύσταση Ακαδημαϊκών Μέσων Δημοσίευσης αξιοποιώντας τη μεθοδολογία RAG

Διπλωματική Εργασία

**Ευθυμία Κουστίκα**

**Επιβλέπουσα:** Ελένη Τουσίδου

Σεπτέμβριος 2025

Approved by the Examination Committee:

Supervisor **Eleni Tousidou**
Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Member **Michael Vassilakopoulos**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Aspassia Daskalopulu**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

# Acknowledgements

Firstly, I would like to express my sincere gratitude to my professor and supervisor, Eleni Tousidou for her guidance and invaluable feedback through the course of this project. Also, I would like to thank the other two members of the committee, Michael Vassilakopoulos and Aspassia Daskalopulu, for their time and effort they gave to my thesis review.

I am also grateful for my family and friends, who showed unwavering patience, support and understanding through all the years of my academic journey. Their continuous hard work to uplift me in every setback I had, was the cornerstone of all my achievements. Finally, a special thank you to my friend and colleague through my academic years, Aristea Koutsomarkou and my partner Pavlos Gkougkoulis who made this journey a core memory for the rest of my life.

# DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Efthymia Koustika

Diploma Thesis

**Implementation of an LLM-Based Chatbot for Academic Venue Recommendations by exploiting RAG**

**Efthymia Koustika**

# Abstract

In our age, the rapid increase of academic publications has made it more challenging to identify appropriate venues for submitting scientific papers. Conventional recommendation systems often rely on keyword matching or citation-based heuristics, which fail to capture the semantic nuances of research topics. This thesis proposes an academic venue recommendation system in the form of a chatbot, built on a Large Language Model (LLM) enhanced with Retrieval-Augmented Generation (RAG). By incorporating retrieved context into the LLM's input, RAG reduces hallucinations and grounds responses in relevant sources, improving reliability and precision. We used the DBLP V14 dataset after extensive filtering, including venue name normalization. Precomputed embeddings for titles, abstracts and keywords were stored in PostgreSQL and indexed with FAISS for efficient semantic retrieval. The system was evaluated under three setups: a traditional machine learning model (TF-IDF+SGD), an LLM-only chatbot and five RAG-based variants. Results show that the basic RAG pipeline delivers the best recommendation quality, highlighting the effectiveness of combining semantic retrieval with LLM reasoning.

**Keywords:**

Academic Venue, Publication, Recommendation System, Large Language Model, Retrieval - Augmented Generation, Chatbot

Διπλωματική Εργασία

**Δημιουργία ενός Chatbot βασισμένου σε LLM για Σύσταση Ακαδημαϊκών Μέσων Δημοσίευσης αξιοποιώντας τη μεθοδολογία RAG**

**Ευθυμία Κουστίκα**

# Περίληψη

Στην εποχή μας, η ραγδαία αύξηση των ακαδημαϊκών δημοσιεύσεων έχει καταστήσει πιο δύσκολη την εύρεση κατάλληλων ακαδημαϊκών χώρων δημοσίευσης για την υποβολή επιστημονικών άρθρων. Τα συμβατικά συστήματα σύστασης συχνά στηρίζονται στην αντιστοίχιση λέξεων-κλειδιών ή σε ευριστικές μεθόδους βασισμένες σε αναφορές, οι οποίες δυσκολεύονται να αποτυπώσουν τις λεπτές σημασιολογικές διαφορές των ερευνητικών θεμάτων. Στην παρούσα εργασία προτείνεται η υλοποίηση ενός Συστήματος Συστάσεων για ακαδημαϊκούς χώρους δημοσίευσης σε μορφή διαλογικού παραθύρου (chatbot), βασισμένο σε ένα Μεγάλο Γλωσσικό Μοντέλο (LLM) στο οποίο έχει εφαρμοστεί η τεχνική Retrieval-Augmented Generation (RAG). Ενσωματώνοντας τα δεδομένα που ανακτήθηκαν στο κείμενο εισόδου του LLM, το RAG μειώνει τις παραισθήσεις και βασίζει τις απαντήσεις του μοντέλου σε σχετικές πηγές, βελτιώνοντας την αξιοπιστία και την ακρίβεια. Χρησιμοποιήσαμε το σύνολο δεδομένων DBLP V14 έπειτα από εκτεταμένο φιλτραρίσμα, περιλαμβανομένης της κανονικοποίησης ονομάτων χώρων. Οι προ-υπολογισμένες διανυσματικές αναπαραστάσεις για τίτλους άρθρων, περιλήψεις και λέξεις-κλειδιά αποθηκεύτηκαν στην PostgreSQL και δημιουργήθηκαν ευρετήρια μέσω του FAISS για αποτελεσματική σημασιολογική ανάκτηση. Το σύστημα αξιολογήθηκε μέσω τριών διαμορφώσεων: ένα παραδοσιακό μοντέλο μηχανικής μάθησης (TF-IDF + SGD), ένα Σύστημα Διαλόγου βασισμένο σε LLM και πέντε παραλλαγές του βασισμένες στο RAG. Τα αποτελέσματα δείχνουν ότι η βασική διαδικασία RAG παρέχει την υψηλότερη ποιότητα συστάσεων, υπογραμμίζοντας την αποτελεσματικότητα του συνδυασμού της σημασιολογικής ανάκτησης με τη συλλογιστική του LLM.

**Λέξεις-κλειδιά:**

Ακαδημαϊκός Χώρος Δημοσίευσης, Δημοσίευση, Σύστημα Συστάσεων, Μεγάλο Γλωσσικό Μοντέλο, Παραγωγή με Ενίσχυση μέσω Ανάκτησης, Σύστημα Διαλόγου

# Table of contents

# List of figures

# List of tables

# Abbreviations

| | |
|---|---|
| LLM | Large Language Model |
| RAG | Retrieval-Augmented Generation |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| SGD | Stochastic Gradient Descent |
| MRR | Mean Reciprocal Rank |
| MVO | Mean Venue Overlap |
| FAISS | Facebook AI Similarity Search |
| GTE | General Text Embeddings |
| NLP | Natural Language Processing |
| ML | Machine Learning |
| SQL | Structured Query Language |
| ANN | Approximate Nearest Neighbor |

# Chapter 1

# Introduction

The area of academic publishing has witnessed tremendous growth in recent decades, with researchers across different fields producing an ever-increasing volume of scientific papers. This expansion, while a positive indicator of global research activity, has created significant challenges in terms of information organization, retrieval and dissemination. Among these, one of the most crucial challenges researchers face is the selection of an appropriate venue to submit their work.

Choosing a publication venue is not a trivial process. It requires alignment with the scope and standards of the journal or conference, awareness of current trends in the field and knowledge of where similar work has been previously published. This is especially difficult for early-career researchers or those working in interdisciplinary areas, who may lack the necessary familiarity with the landscape.

Existing solutions to assist with venue selection typically rely on simple heuristics, keyword matching, or statistical learning methods that leverage surface-level text features. However, such approaches often fail to capture the semantic complexity of research topics or to adapt dynamically to new scientific developments.

In parallel, recent advances in artificial intelligence-particularly in the field of natural language processing-have enabled the emergence of Large Language Models (LLMs). These models exhibit impressive capabilities in understanding and generating human-like text, suggesting their potential to support more context-aware systems. However, LLMs also have limitations, particularly when it comes to grounding their responses in current or domain-specific information.

These challenges highlight the broader need for systems that are capable of interpreting

the content of research work while being aware of its contextual relevance within the scholarly landscape.

## 1.1 Subject of thesis

This thesis addresses the problem of recommending suitable academic venues for newly written scientific papers based solely on their abstract. As outlined in the previous section, researchers often struggle to determine where to submit their work, particularly in interdisciplinary or unfamiliar domains. Existing automated methods, based on keyword matching or shallow learning models, often lack the semantic depth and adaptability required for accurate and context-aware recommendations.

So, in this project, we examine the utilization of LLMs enhanced through a Retrieval-Augmented Generation (RAG) framework as a solution to this issue. The proposed system generates venue recommendations by first retrieving semantically similar papers from a vast academic corpus and then using these examples as context to guide the predictions of the language model. The anticipated outcome of this retrieval mechanism is to grant relevance and grounding, thereby assisting the model in steering clear of generic or hallucinated suggestions and generating outputs that correspond with real publication patterns.

In the next chapters, we will be presenting the design and implementation of this system, involving the preprocessing of a large-scale academic dataset, the generation of semantic embeddings for paper abstracts and venue names and the development of a retrieval component based on vector similarity. These elements are integrated with a language model capable of producing venue suggestions in natural language. We also examine how variations in the retrieval and reranking process influence the overall quality of the recommendations.

In doing so, we explore both the capabilities and limitations of RAG-based architectures in supporting decision-making tasks within academic publishing, highlighting the technical challenges of semantic retrieval, prompt design and evaluation alignment.

### 1.1.1 Contribution

The contribution of this thesis can be summarized as follows:

1. Investigated limitations of existing venue recommendation approaches and studied relevant work in the field.

2. Designed a retrieval-augmented generation (RAG) system that combines large language models with contextual retrieval of semantically similar papers.

3. Preprocessed and curated a large academic dataset and applied normalization techniques to ensure venue consistency.

4. Developed and compared several system configurations, including traditional machine learning and LLM-based methods.

5. Proposed and applied evaluation metrics to assess both the precision of the recommendations and the contextual alignment.

## 1.2 Thesis Structure

Following this introductory chapter, the remainder of the thesis is organized as follows: Chapter 2 presents a review of related work, covering previous research on venue recommendation, retrieval-augmented generation and applications of large language models in scholarly contexts. Chapter 3 provides the necessary background theory, outlining the principles of large language models, retrieval-augmented generation and related concepts. Chapter 4 describes the dataset and tools used, detailing their characteristics and the software frameworks employed. Chapter 5 explains the methodology, including data preprocessing, embedding generation, index construction, baseline models and the design of both LLM-only and RAG-based recommenders. Chapter 6 presents the evaluation setup and experimental results, followed by a comparative analysis of the different approaches. Finally, Chapter 7 discusses limitations and proposes potential directions for future research.

# Chapter 2

# Related Work

## 2.1 Introduction

This chapter reviews related work on academic venue recommendation, with emphasis on techniques relevant to this thesis. It covers early keyword-based methods, supervised classification approaches and more recent systems using semantic embeddings. It also discusses the role of Large Language Models (LLMs) and the development of Retrieval-Augmented Generation (RAG) as a means to improve contextual accuracy in text generation.

## 2.2 Traditional Approaches to Venue Recommendation

Early research on venue recommendation framed the task as a classification or matching problem, using textual features such as TF-IDF or bag-of-words representations of paper titles and abstracts combined with classifiers like SVMs or logistic regression. For instance, Luis M. de Campos et al. [1] built topic-based profiles for venues using clustering and information retrieval techniques to match new papers against them, demonstrating the efficacy of content-focused approaches.

In addition, scholars explored metadata-enhanced models that leverage information beyond mere text. Hiep Luong et al.[2] created co-authorship graphs to support venue recommendations to emphasize the role of co-authorship and institutional collaboration in guiding recommendation accuracy, showing that integrating author networks boosts selection relevance.

Among hybrid approaches that combine structural metadata with author behavior, one

notable contribution is the PAVE model [3], which formulates venue recommendation as a graph-based problem. In this method, a heterogeneous co-publication network is constructed where nodes represent authors and venues and edges capture various relationships such as co-authorship frequency, author-venue interactions and researcher seniority. Personalized venue suggestions are generated through a biased random walk with restart, favoring venues that exhibit stronger graph connectivity to a target author.

## 2.3 Neural and Embedding-Based Recommendation Systems

A prominent example of this type of recommendation systems is GraphConfRec [4], which employs a graph neural network over a heterogeneous scholarly graph integrating paper content, authorship, citations and venues. By jointly embedding these modalities, the model achieved recall@10 of 0.58 and MAP of 0.336, significantly outperforming content-only baselines.

Other systems explore document-level semantic embeddings. Arman Cohan et al. [5] train a transformer-based model using citation links as supervision, producing science-domain embeddings that boost performance on classification and recommendation tasks.

Stergiopoulos et al. [6] present a large-scale academic recommender system that combines topic-level graph modeling, clustering and deep learning. Using the DBLP AMiner Citation Network, they build a weighted graph of fields of study (fos) from their co-occurrence in papers and apply K-means clustering to group related topics. For each user, relevant fos are identified via keyword similarity and the candidate set is expanded through cluster and graph connections. The final ranking is produced by a hyperparameter-tuned CATA++ model, which is a collaborative dual attentive autoencoder integrating matrix factorization, collaborative filtering and attention mechanisms.

## 2.4 Language Models in Scholarly NLP Tasks

Large Language Models (LLMs) pretrained on scientific corpora have advanced several tasks within scholarly NLP by capturing deeper semantic structures.

SciBERT [7] is a domain-specific variant of BERT, pretrained on 1.14 million scientific

papers. It consistently outperforms general-purpose BERT in tasks such as sequence tagging, dependency parsing and paper classification, thanks to its vocabulary and embedding space adapted to scientific language.

Another study [8], compares SciBERT with other models like SciNCL and SPECTER2. Their findings show that fine-tuned domain-specific embeddings significantly improve classification across 123 Open Research Knowledge Graph categories, achieving an F1-score of 0.7415.

Furthermore, recent studies have begun to explore the use of open-weight language models such as Mistral in domain-specific scholarly tasks. For instance, Elliot Bolton et al. [9] evaluate the performance of Mistral 7B in the context of clinical question answering, comparing it with several biomedical-focused models on datasets such as MedQA. The results show that Mistral achieves a competitive accuracy of 63%, outperforming various domain-tuned models in general consumer health queries. This suggests that, mid-sized, general-purpose LLMs can offer strong baseline performance in specialized academic tasks. However, the study also highlights the fact that while Mistral excels in efficiency and general comprehension, it can still underperform compared to larger closed-source models in tasks requiring deep domain expertise.

## 2.5 Retrieval-Augmented Generation (RAG)

In recent years, the paradigm of Retrieval-Augmented Generation (RAG) has gained substantial traction as a means of enhancing the factual accuracy and contextual relevance of LLMs. While originally introduced for open-domain question answering, RAG [10] has since been adapted for use in academic and scientific tasks, where domain grounding and information traceability are essential. Several recent studies illustrate how RAG-based systems can be successfully employed across different scholarly contexts.

Guangzhi Xiong et al. in their study [11] systematically evaluate RAG systems in the medical domain. The authors introduce the Medical Information Retrieval-Augmented Generation Evaluation (MIRAGE) benchmark, which comprises 7,663 clinical QA questions drawn from five medical datasets. They assess 41 different RAG configurations-combining various corpora, retrievers and backbone LLMs, with over 1.8 trillion prompt tokens tested. Their findings demonstrate that RAG techniques can improve accuracy by up to 18% compared to

chain-of-thought prompting, elevating the performance of models like GPT-3.5 and Mixtral to GPT-4 levels.

Additionally, Ahmet Yasin Aytar et al. [12] present RAGAS, a retrieval-augmented framework tailored for scholarly data science tasks. Their system integrates domain-specific embeddings, semantic chunking of scientific documents and bibliographic metadata parsing using tools like GROBID. Unlike more generic RAG setups, RAGAS emphasizes high-quality text segmentation and structured metadata usage to improve retrieval granularity. In evaluations involving real academic navigation tasks, such as finding related literature or identifying relevant authors, RAGAS demonstrates clear improvements in both relevance and usability.

Finally, the work of Xianrui Zhong et al.[13] explores the use of RAG in a chemistry-specific setting. The authors construct a benchmark based on curated corpora including PubMed, PubChem and textbooks and evaluate multiple RAG configurations for complex chemistry-related question answering. Their modular approach allows for testing different retrieval models and document sources. Experimental results show that RAG-based models achieve an average 17.4% improvement in factual correctness compared to LLMs operating without retrieval. This work is quite important as it demonstrates the adaptability of RAG to highly technical scientific domains, where accurate grounding is essential.

# Chapter 3

# Background Theory

## 3.1 Traditional Machine Learning Baseline

Before the widespread adoption of deep learning and transformer-based models, classical machine learning pipelines offered practical and effective solutions for text classification tasks. One of the most well-established approaches combines Term Frequency-Inverse Document Frequency (TF-IDF) vectorization with a linear classifier such as Stochastic Gradient Descent (SGD).

TF-IDF transforms textual input into high-dimensional sparse vectors that reflect the relative importance of terms. It downweighs common terms that appear frequently across documents while emphasizing rarer, more discriminative words. This representation is particularly suitable for tasks where lexical signals play an important role in distinguishing between classes. When applied to academic abstracts, TF-IDF highlights terms that are informative for identifying the thematic focus of a paper.

The TF-IDF score for a term $t$ in document $d$ within a corpus $D$ is defined as:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \cdot \log\left(\frac{N}{|\{d' \in D : t \in d'\}|}\right) \tag{3.1}$$

where $\text{TF}(t, d)$ denotes the frequency of term $t$ in document $d$, $N$ is the total number of documents in the corpus $D$ and the denominator counts how many documents contain term $t$. This formulation assigns higher weights to terms that are frequent in a specific document but rare across the entire collection.

To classify these representations, a linear model such as the SGDClassifier is often used. SGD is computationally efficient for large datasets and high-dimensional input spaces. It up-

dates model weights incrementally using mini-batches, allowing it to scale effectively while remaining interpretable. The model learns to associate term patterns with specific publication venues based on training data.

The weight update rule for SGD is given by:

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \mathcal{L}(\theta_t; x_i, y_i) \tag{3.2}$$

where $\theta_t$ represents the model parameters at iteration $t$, $\eta_t$ is the learning rate and $\mathcal{L}$ is the loss function evaluated on training sample $(x_i, y_i)$.

This type of pipeline has been widely adopted and studied in the literature. For example, [14] evaluate TF-IDF in combination with linear classifiers for SMS spam detection and show that it outperforms several deep learning baselines in terms of accuracy and training efficiency. In another case, Schäfermeier et al. [15] propose a venue recommendation approach that represents papers with TF-IDF features and applies Non-negative Matrix Factorization (NMF) for dimensionality reduction.

Despite its strengths, this traditional approach has clear limitations. It cannot capture semantic similarity between texts that use different vocabulary to describe the same concepts. Furthermore, its reliance on surface-level word features limits its ability to generalize to new or interdisciplinary venues that lack term overlap with the training set.

However, the TF-IDF + SGD Classifier pipeline serves as a baseline in this thesis, providing a point of comparison for more advanced methods involving semantic embeddings and large language models.

## 3.2 Semantic Representations & Dense Vector Retrieval

Traditional representations of text, such as Term Frequency-Inverse Document Frequency (TF-IDF), rely on sparse vector encodings based on word frequency statistics. While computationally efficient, these methods fail to capture the semantic content of language, treating each word as an independent token and ignoring syntax, word order and contextual relationships. This makes them insufficient for tasks that require understanding beyond surface-level lexical overlap.

To address this, recent advances in natural language processing have shifted focus toward semantic embeddings, which are dense, fixed-size vector representations that encode

the meaning of words, sentences, or documents. These embeddings are generated by neural models trained on large corpora and are designed to position semantically similar texts close together in the vector space. This allows for effective comparison of textual units based on meaning rather than exact wording [16].

These embeddings are typically trained using objectives such as contrastive learning or masked language modeling, allowing the model to learn relationships between different text spans. Unlike static embeddings (e.g., Word2Vec or GloVe), which assign the same vector to a word regardless of context, transformer-based models produce context-aware embeddings, enabling nuanced semantic representation.

A common approach to generating semantic embeddings is through sentence transformers, which extend pre-trained transformer architectures such as BERT [17]. Sentence Transformers enable efficient sentence-level embeddings using architectures such as Siamese or triplet networks during training to ensure that the generated vectors reflect semantic similarity. These models process input sequences and output contextual embeddings, which are then aggregated (e.g., via mean pooling) to form a single vector per sentence or document. In this thesis, we use the GTE-small model for this purpose, encoding both paper abstracts and venue names into a shared embedding space.

To compare embeddings for similarity, cosine similarity is commonly used. It measures the cosine of the angle between two vectors and is defined as:

$$\cos(\theta) = \frac{u \cdot v}{|u||v|} \tag{3.3}$$

A cosine similarity close to 1 indicates high semantic similarity, even if the texts use different vocabulary. This makes such methods particularly valuable in academic venue recommendation, where similar research topics may be expressed using varied terminology.

Once all documents are represented as dense vectors, the recommendation task becomes a nearest neighbor search in high-dimensional space. Given a query vector (e.g., the embedding of a paper abstract), the goal is to efficiently retrieve the most similar vectors from a large index. This is known as Approximate Nearest Neighbor (ANN) search. ANN methods trade exactness for computational efficiency, making them ideal for large-scale semantic retrieval.

## 3.3 Large Language Models (LLMs)

Large Language Models (LLMs) represent a transformative advancement in natural language processing. These models are typically based on the Transformer architecture introduced by Vaswani et al. [18], which employs self-attention mechanisms to learn contextual representations of input sequences. As shown in Figure **??**, the Transformer consists of an encoder-decoder architecture that relies on multi-head attention, feed-forward layers and residual connections combined with normalization to model long-range dependencies between tokens.

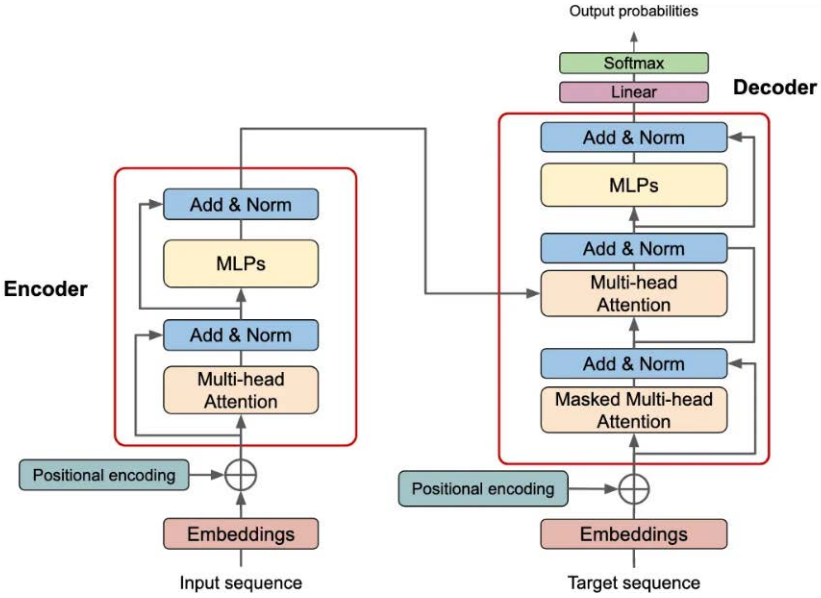


Figure 3.1: Transformers architecture[1]

Trained on massive text corpora, LLMs learn generalizable language patterns, making them effective for a wide range of tasks such as summarization, classification, question answering and text generation, often with little or no task-specific supervision. Well-known examples include GPT-3 [19], LLaMA [20] and PaLM [21], each varying in scale, architecture and training data composition. These models support in-context learning, where task adaptation occurs directly through prompt engineering rather than gradient-based fine-tuning.

[1]Source: `https://www.researchgate.net/publication/392138444_Augmenting_Orbital_Debris_Identification_with_Neo4j-Enabled_Graph-Based_Retrieval-Augmented_Generation_for_Multimodal_Large_Language_Models`

### 3.3.1 Capabilities

LLMs exhibit strong generalization capabilities, allowing them to perform a diverse set of language-related tasks without extensive retraining. Through prompt-based interactions, they can adapt to new tasks, follow instructions and synthesize knowledge from different sources. Their ability to reason over complex text, generate coherent language and identify relationships between concepts makes them useful in applications ranging from chatbots and content summarization to recommendation systems and semantic search engines.

Furthermore, LLMs can interpret natural language queries and generate relevant responses by leveraging their internal representations of language structure and meaning. This flexibility allows developers to build systems that can interact with users using free-form language rather than rigid rule-based inputs.

### 3.3.2 Limitations

Despite their expressive power, LLMs face two major challenges in practical applications. First, they are prone to hallucinations such as, generating factually incorrect or unverifiable content [22]. This is a significant concern when the model is expected to provide accurate and trustworthy information.

Second, LLMs are inherently ungrounded. Since they rely on static training data and lack access to external sources unless explicitly augmented, they struggle to reflect current knowledge or adapt to domain-specific requirements during inference. As a result, their recommendations or predictions may be disconnected from the most relevant or up-to-date information.

These limitations have motivated the development of hybrid methods such as Retrieval-Augmented Generation (RAG), which combine the language generation capabilities of LLMs with the contextual precision of document retrieval systems.

## 3.4 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a hybrid framework that integrates retrieval-based methods with large language models to address the limitations of purely generative approaches.

### 3.4.1 Overview of RAG

Instead of relying solely on the model's internal parameters to generate output, Retrieval-Augmented Generation (RAG) augments the generation process by retrieving relevant documents from an external knowledge corpus. As illustrated in Figure 3.2, the retrieved texts are combined with the original query and passed to the language model, grounding its output in external evidence and reducing the risk of hallucination [23]. As mentioned in the previous chapter, this architecture was first introduced by Patrick Lewis et al. [10], in the context of open-domain question answering.

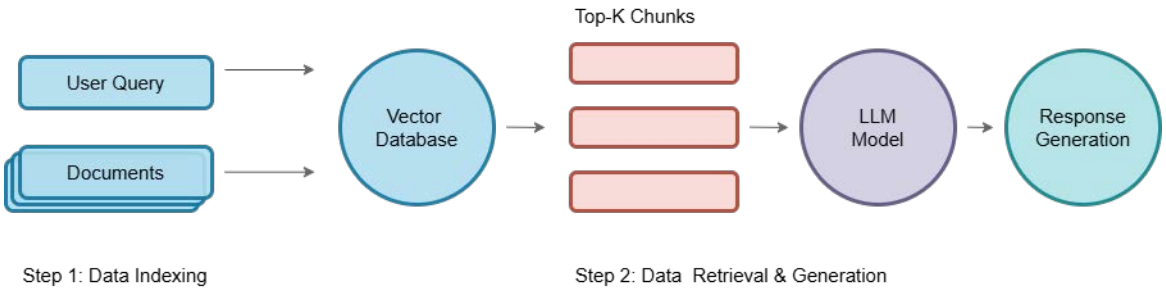


Figure 3.2: Basic RAG pipeline

Their model combined a dense retriever (DPR) with a sequence-to-sequence generator (BART), showing notable improvements in factual accuracy and response relevance. The two-stage pipeline involves:

1. Retrieving top-$k$ semantically similar documents given a query and
2. conditioning a generative model on both the query and retrieved passages to produce an answer.

Subsequent works have extended this paradigm across domains such as biomedical QA, legal reasoning, and customer support, demonstrating RAG's general applicability. Surveys such as Gao et al. [24] have classified RAG systems into categories such as naive, modular and advanced architectures and have emphasized enhancements like reranking, hybrid retrieval, and the use of knowledge graphs. An example of such an enhanced design is shown in Figure 3.3, where retrieved results are reranked before being passed to the language model.

Recent adaptations explore RAG in recommendation systems. For instance, ER2ALM incorporates retrieval-aware prompting for improved item recommendation using retrieved metadata and descriptions [25]. In academic venue recommendation, the RAG framework

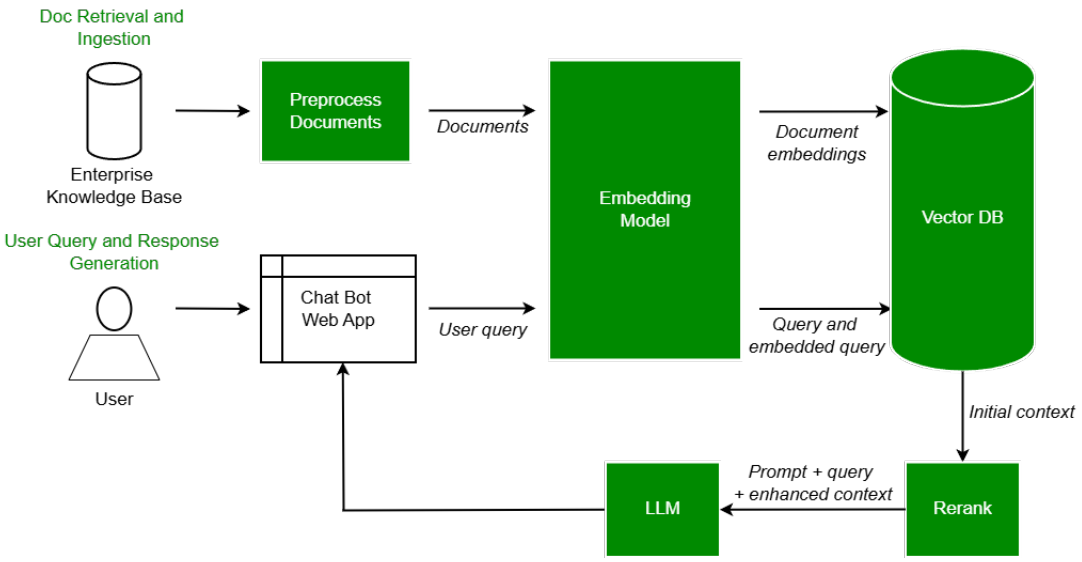


Figure 3.3: RAG pipeline with reranking

addresses core challenges of semantic mismatch and hallucination. By grounding predictions in real, semantically similar publications retrieved from a scholarly corpus, RAG enables contextually aware and evidence-backed venue suggestions. This grounding mechanism is particularly crucial in research domains where terminologies vary and model hallucinations can yield misleading recommendations.

### 3.4.2 Fine-Tuning vs. RAG

While fine-tuning large language models is another strategy for adapting them to domain-specific tasks, it is often less practical in this context. Fine-tuning requires large labeled datasets, substantial computational resources and frequent retraining to remain up to date with new data. Moreover, a fine-tuned model tends to encode knowledge statically, which makes it less flexible in rapidly evolving research areas. In contrast, RAG provides a more efficient and adaptive solution by combining a general-purpose LLM with an external retrieval component. Instead of modifying model parameters, RAG dynamically incorporates current and semantically relevant literature into the prediction process, reducing hallucinations and improving factual grounding without the prohibitive cost of large-scale fine-tuning.

As seen in the recent work of Ovadia et al. [26], who evaluate the two approaches across knowledge-intensive tasks, RAG consistently outperforms unsupervised fine-tuning. Their findings indicate that RAG is superior not only for incorporating novel information but also for reinforcing knowledge already present in the model, supporting its role as a more flexible

and resource-efficient alternative to fine-tuning.

## 3.5 Impact Factor

Beyond thematic relevance, researchers also consider the reputation of publication venues when selecting where to submit their work. This dimension is often captured through bibliometric indicators such as the Journal Impact Factor (IF). This indicator approximates the visibility and perceived prestige of a venue within the scientific community and has influenced decisions in academic publishing form a long time [27].

The Impact Factor is one of the most widely recognized metrics, defined as the average number of citations received in a given year by articles published in a venue during the preceding two years:

$$\text{IF}_y = \frac{\text{Citations in year } y \text{ to items published in years } y-1 \text{ and } y-2}{\text{Number of citable items published in years } y-1 \text{ and } y-2} \tag{3.4}$$

Higher values indicate venues whose recent publications are frequently cited and therefore considered more influential. While the metric has well-known limitations, such as field-dependence and susceptibility to manipulation, it remains a central indicator of venue prestige [28].

In the context of recommendation systems, such scores can be incorporated as auxiliary signals alongside topical similarity. For example, Schäfermeier et al.[15] propose venue ranking frameworks that integrate topical alignment with venue-specific profiles, while other works suggest using citation-based metrics to rerank candidate venues. This combination reflects the realistic decision process of authors, who aim not only for thematic fit but also for maximal academic visibility.

## 3.6 Evaluation Metrics

To assess the performance of the venue recommendation system, a set of standard and task-specific evaluation metrics is used. These metrics aim to capture various aspects of recommendation quality, including accuracy, ranking quality and contextual alignment with the retrieved material.

### 3.6.1 Top-K Accuracy (Hit Rate)

Top-K accuracy [29] measures whether the correct venue appears within the top K predicted venues.

$$\text{Top-K Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\delta(y_i \in \hat{Y}_i^{(K)}) \tag{3.5}$$

where $N$ is the total number of test instances, $y_i$ is the ground truth venue for instance $i$, $\hat{Y}_i^{(K)}$ is the set of top-$K$ predicted venues and $\delta(y_i \in \hat{Y}_i^{(K)})$ is an indicator function equal to $1$ if $y_i$ is included in the top-$K$ predictions and $0$ otherwise.

Essentially, for each test instance, if the ground-truth venue appears within the top K predictions, it is considered a "hit". The final score is reported as the percentage of instances with at least one hit in the top K. It is a straightforward metric that provides insight into how often the model's suggestions include a relevant venue within the top K positions.

### 3.6.2 Mean Reciprocal Rank (MRR)

MRR evaluates the position of the first correct recommendation in the ranked list [30].

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i} \tag{3.6}$$

where $N$ is the total number of test instances and $\text{rank}_i$ denotes the position of the ground truth venue $y_i$ in the ranked list of predictions for instance $i$.

For each test case, the reciprocal rank is defined as the inverse of the rank at which the first correct venue appears. The mean of these values across all test instances provides an aggregate measure of how early relevant venues are being suggested. MRR places greater emphasis on the correct venue that appears near the top of the recommendation list.

### 3.6.3 Mean Venue Overlap (MVO)

MVO is a custom metric introduced to measure semantic consistency between the final venue recommendations of the model and the venues associated with the retrieved context.

$$\text{MVO} = \frac{1}{N}\sum_{i=1}^{N}\left|\left\{v \in \hat{Y}_i \;\middle|\; \max_{c\in C_i}\cos\big(e(v), e(c)\big) \geq \tau\right\}\right| \tag{3.7}$$

where $N$ is the number of test instances, $\hat{Y}_i$ is the list of predicted venues for instance $i$ returned by the LLM and $C_i$ is the set of unique venues appearing in the retrieved context for that instance. The function $e(\cdot)$ denotes the normalized embedding representation of a venue name and $\tau$ is the cosine similarity threshold used to determine whether a predicted venue is considered semantically similar to any retrieved venue. For each instance, the metric counts how many of the predicted venues meet this similarity criterion, and reports the average count across all instances.

In other words, a higher MVO indicates that the model's suggestions are more aligned with the material it was provided, reflecting better contextual grounding. This metric is especially valuable in RAG setups, where the relevance of retrieved documents to the final output is a key consideration. The design of this metric draws inspiration from the semantic alignment approach used by Schäfermeier et al. [15], who measure the similarity between a paper's topic distribution and venue topic profiles in their explainable venue recommendation framework.

# Chapter 4

# Data and Tools

## 4.1 Dataset Description

The dataset used in this thesis is the DBLP V14 Citation Network [31], a large-scale academic corpus focusing on computer science literature. This dataset comprises approximately 5.26 million research articles, enriched with nearly 36.6 million citation links and spans publications over several decades, up to early 2023. It is distributed in JSON format and includes rich metadata for each publication, enabling comprehensive bibliometric analysis and recommendation tasks.

Each entry in the dataset contains detailed bibliographic information such as the title, abstract, publication year and the name of the venue where the paper appeared. Author information is also included, with full names, institutional affiliations and unique identifiers. Citation-related metadata such as the number of citations received, references to other papers and fields of study with their relevance weights are also part of the dataset. Additional attributes include the document type, page ranges, volume, issue, DOI, ISSN and the language in which the paper was written.

One of the key features of the DBLP V14 dataset is its suitability for scholarly recommendation systems. The availability of both structured metadata and unstructured text (e.g., titles and abstracts) makes it possible to apply a range of learning-based techniques. The inclusion of raw venue names provides a natural label space for classification or retrieval-based tasks focused on venue recommendation.

The dataset is publicly accessible through the AMiner Citation Network repository and it has been widely used in academic research involving citation analysis, co-authorship network

modeling and publication trend detection.

## 4.2 Tools and Frameworks

A variety of tools and frameworks were used throughout the thesis for data preprocessing, embedding generation, similarity search, LLM orchestration and evaluation. These tools were chosen based on their scalability, flexibility and support for modern NLP and ML workflows.

### 4.2.1 PostgreSQL

PostgreSQL is an open-source Relational Database Management System (RDBMS) known for its stability, extensibility and compliance with SQL standards [32]. It supports advanced features such as user-defined types, full-text search and transactional integrity, making it well-suited for both analytical and transactional workloads.

In this thesis, PostgreSQL was used primarily during the preprocessing stage for storing and manipulating intermediate data derived from the DBLP V14 dataset. This included the storage of filtered and cleaned paper metadata, normalized venue names and precomputed semantic embeddings. Its support for efficient indexing and bulk operations made it well-suited for handling large-scale data ingestion and transformation tasks.

It is important to note that PostgreSQL was not used during the runtime of the recommendation system itself. Instead, it functioned as a back-end storage solution during data preparation and evaluation set construction, prior to embedding indexing and retrieval via other components such as FAISS.

### 4.2.2 Facebook AI Similarity Search (FAISS)

FAISS [33] is a library developed by Facebook AI Research for efficient similarity search and clustering of dense vectors in high-dimensional spaces. It is designed to work for billion-scale datasets and supports both exact and approximate nearest neighbor (ANN) search. FAISS achieves this by combining various indexing strategies, such as inverted file (IVF) indexing, product quantization (PQ) and graph-based structures like Hierarchical Navigable Small World (HNSW), depending on the performance and accuracy requirements.

In the context of this thesis, FAISS was used to build a dense vector index over paper abstract embeddings. This enabled fast retrieval of semantically similar documents based

on cosine similarity or dot product. The library's support for both CPU and GPU backends facilitated scalable experimentation, especially during repeated evaluations of venue recommendations. Its Python bindings and compatibility with PyTorch make it a natural fit for integration with modern neural embedding pipelines.

### 4.2.3 Sentence Transformers and GTE-small

Semantic embeddings for paper abstracts and venue names were generated using the GTE-small model, accessible through the sentence-transformers Python library. Sentence transformers extend traditional transformer-based language models such as BERT by introducing a pooling mechanism to produce fixed-size dense vector representations for entire sentences or documents [16]. These representations are optimized to capture semantic similarity, allowing for meaningful comparison between textual inputs even when surface vocabulary differs significantly.

The GTE-small sentence-embedding model is a lightweight, multilingual encoder optimized for tasks involving semantic textual similarity, information retrieval and question answering. It is trained using contrastive objectives on multilingual corpora, enabling it to place semantically related texts close together in the embedding space. Despite its compact size, GTE-small offers a favorable trade-off between accuracy and computational efficiency, making it well-suited for large-scale retrieval applications such as the academic venue recommendation system developed in this thesis.

### 4.2.4 Ollama and Mistral LLM

To facilitate large language model (LLM) inference in a self-hosted and reproducible manner in this thesis, we employed the Mistral language model via the Ollama framework. Ollama is an open-source system designed for running LLMs locally, enabling developers to load, run and interact with state-of-the-art models without relying on external APIs or cloud services. It provides a container-like interface for managing LLMs, supports model quantization for efficient deployment and offers GPU acceleration where available.

The Mistral model itself is an open-weight, decoder-only transformer architecture developed to achieve strong language understanding and generation performance while maintaining computational efficiency. Specifically, the model used in this thesis is the mistral-7B-instruct variant, which has been fine-tuned using instruction-following datasets to improve

alignment with human intent. Its architecture is similar to LLaMA-style transformers, incorporating techniques such as grouped-query attention (GQA) and sliding window attention for faster inference [34].

In this work, Mistral serves as the generative backbone for both the LLM-only and Retrieval -Augmented Generation (RAG) pipelines. In the LLM-only setup, it is prompted with a paper abstract and asked to predict a suitable venue. Running the model through Ollama ensures consistency, low-latency inference and data privacy during experimentation.

### 4.2.5 LangChain

LangChain [35] is a modular framework designed to streamline the development of applications powered by large language models. In this thesis, LangChain was employed as the orchestration layer that coordinates various components of the LLM-based pipeline, including the prompt, the retrieval logic, the reranking and the generative response synthesis. The hierarchical layers it provides us with, enable the integration of retrieval mechanisms (such as FAISS), reranking models and local inference backends (such as Ollama) into flexible and reusable chains.

LangChain provides a standardized interface for implementing RAG workflows, supporting both declarative chain construction and imperative control via Python. It also allows dynamic prompting, result parsing and contextual memory management, making it particularly useful in experimental and iterative setups.

Through its compatibility with embedding models, vector databases and LLM interfaces, LangChain significantly reduces the complexity involved in integrating disparate components. This facilitated rapid prototyping and evaluation of multiple RAG configurations while maintaining reproducibility and modularity.

### 4.2.6 Python Libraries

The implementation of this thesis is based on the Python programming language. Python offers extensive support for modular experimentation, data manipulation, model training and evaluation, making it the best choice for building and testing recommender systems.

Several libraries were used to support various stages of the pipeline:

- `scikit-learn` was used for implementing traditional machine learning baselines,

particularly the TF-IDF vectorizer and the SGDClassifier for venue prediction.

- `pandas` and `numpy` facilitated efficient manipulation of structured data and numerical arrays, essential for preprocessing, filtering and analysis.
- `json` was used extensively to parse and write intermediate representations and results.
- `RapidFuzz` for fuzzy string matching during venue normalization, chosen for its speed and accuracy in comparing large numbers of textual labels.
- `faiss` provided the Python interface to the FAISS library, enabling approximate nearest neighbor search over dense vector representations.
- `sentence-transformers` offered a high-level API to load and use pretrained models like GTE-small for generating semantic embeddings.
- `torch` served as the backend for running transformer-based models used in embedding generation and LLM inference.
- `langchain` was used for composing modular LLM pipelines including retrieval and generation components.

# Chapter 5

# Methodology

## 5.1 Overview

This chapter describes the pipeline developed for academic venue recommendation using traditional machine learning, LLMs and RAG strategies. Each component of the pipeline is designed to evaluate a different aspect of model capability, including word similarity, semantic relevance and grounded generation.

## 5.2 Data preprocessing

To ensure that any downstream modeling and analysis is based on reliable, consistent and semantically meaningful data, a comprehensive preprocessing pipeline was implemented. The raw DBLP V14 dataset, although rich in information, contains a variety of inconsistencies, irrelevant records and noise that must be addressed before any meaningful computation can take place. This multi-stage preprocessing process was designed to clean, filter, normalize and enrich the data, ultimately improving its quality.

### 5.2.1 Data filtering and insertion into PostgreSQL

The first step involved extracting and filtering relevant entries from the original DBLP V14 JSON dataset, which contains over 5 million records. The data was filtered based on a set of quality and relevance criteria:

1. **Presence of abstract**: Papers without abstracts were excluded, as abstracts serve as

the primary textual feature for venue prediction.

2. **Citation count**: Records with zero citations were removed to eliminate potentially low-impact or incomplete entries.

3. **Language filtering**: Non-latin documents, were discarded to maintain consistency in language-based semantic modeling.

4. **Venue availability**: Only entries containing a valid venue.raw field were retained, as venue prediction requires labeled data.

After applying these filters, the remaining records were batch-inserted into a PostgreSQL database named AMINER_V14, which served as the central repository for further processing and querying. The core publication metadata was stored in a table named papers, with structured columns capturing fields such as id, title, abstract, authors, keywords, fos, n_citation and additional bibliographic metadata.

### 5.2.2 Venue normalization via Fuzzy Matching

One of the most significant challenges in working with this dataset is the inconsistency in venue naming. The same academic venue may appear in multiple variants due to typographical errors, inconsistent abbreviation styles, or inclusion of metadata such as publication years and volume numbers. These inconsistencies inflate the number of unique class labels, introduce noise and severely hinder the performance of any model tasked with venue classification or recommendation.

To address this issue, a dedicated normalization routine was implemented to systematically clean and consolidate raw venue names. Initially, raw venue strings were processed to remove noise: they were lowercased, stripped of accents and non-alphabetic characters and cleared of year tokens, volume indicators and acronyms enclosed in parentheses. Additionally, uninformative entries, such as “unknown”, “null”, or “test”, were filtered out entirely.

Following this initial cleaning, fuzzy string matching was applied to group venue names that referred to the same publication outlet but differed slightly in their textual form due to typos, formatting inconsistencies, abbreviations, or minor variations in naming. The RapidFuzz library was chosen for this step due to its speed and accuracy, outperforming alternatives like FuzzyWuzzy in terms of performance [36]. A similarity threshold of 93% was empirically selected after experimentation.

This process reduced the number of unique venue entries significantly. From an initial set of over 36,024 raw venue names extracted from the dataset, the normalization procedure reduced this to approximately 24,788 unique standardized venue names. This process eliminated inconsistencies such as formatting variations, noise and duplicates. Subsequently, a filtering step retained only venues with at least 100 associated publications, resulting in a final set of roughly 3,500 high-frequency venues used for the task at hand. This progressive refinement was essential for reducing label noise and controlling any label imbalance.

### 5.2.3 Text embeddings

To enrich the dataset with semantically meaningful representations of each paper's content, a dense vector embedding was computed for three core textual fields: the title, abstract and keywords. For this purpose, the open-source model **thenlper/gte-small** was used. This model produces fixed-size, 384-dimensional vectors that effectively capture the semantic essence of textual inputs.

In addition to per-paper embeddings, sentence embeddings were also generated for all distinct normalized venue names in the dataset. These canonical venues were retrieved from the venues_norm table in the PostgreSQL database, embedded in batches using the same gte-small model and saved as a JSON file.

A key design decision was to precompute all embeddings and store them. This reduces latency at query time and separates computationally expensive processes from downstream components. Since all computations were carried out locally on a personal laptop, encoding was performed on an NVIDIA GeForce RTX 3050 Ti Mobile GPU, which helped accelerate the embedding process over millions of entries.

## 5.3 Index Construction

To enable efficient semantic retrieval of similar academic papers, we constructed a high dimensional vector index using FAISS (Facebook AI Similarity Search). This index is central to the retrieval operations used in both LLM-based and RAG models.

### 5.3.1 FAISS Index Construction

Although the underlying metadata and embeddings were stored in a PostgreSQL database, the system required a more specialized solution for large-scale semantic retrieval. PostgreSQL excels at structured queries and relational joins, but it is not optimized for high-dimensional vector similarity search. Even with extensions such as pgvector, performance degrades significantly as the dataset grows, making SQL alone impractical for millions of embeddings.

For this reason, FAISS was adopted to handle Approximate Nearest Neighbor (ANN) search efficiently. After generating sentence embeddings for each paper abstract using the thenlper/gte-small model, the resulting dense vectors were assembled into a unified NumPy array, restricted to papers that met the preprocessing criteria (non-empty abstract, known normalized venue and passing citation and language filters). FAISS provides a suite of indexing algorithms tailored to high-dimensional vector spaces and for this project the IndexFlatIP structure was selected. This index computes similarity using the inner product, which directly corresponds to cosine similarity when vectors are normalized.

The index was constructed from the full array of abstract embeddings and serialized to disk for reuse. During inference and evaluation, this design allowed the index to be memory-mapped, avoiding repeated recomputation or data loading. Depending on the available runtime environment, FAISS was configured to exploit either CPU or GPU resources to accelerate both indexing and query operations.

### 5.3.2 Metadata Storage for Indexed Papers

In addition to the vector index itself, the metadata corresponding to each indexed paper was stored. For every embedded abstract, a structured metadata object was constructed containing:

- `id`
- `title`
- `abstract`
- `venue_norm`
- `n_citation`

These metadata entries were saved in a parallel JSON file, aligned with the vector index by maintaining consistent order between embedding vectors and their corresponding metadata entries. The separation of vector data from descriptive metadata was done to achieve a modular structure that simplifies debugging, evaluation and downstream experimentation. It also ensures that the retrieval step remains lightweight and focused solely on fast similarity computation.

For the RAG with Impact-Factor reranking experiment described later in this chapter, the metadata was further enriched with an additional attribute representing a proxy impact factor score for each venue. This score was derived from citation statistics to approximate venue-level prestige, enabling reranking strategies that incorporate not only semantic similarity but also the relative standing of publication venues.

## 5.4 Model Architecture and System Design

In this section are included the details of the various venue recommendation models developed in this thesis, including a traditional baseline, a standalone LLM-based generator and four RAG pipelines. Each model adopts a certain methodology, ranging from shallow feature-based learning to grounded generation using external context.

### 5.4.1 TF-IDF & SGD Classifier

As a foundational baseline for the venue recommendation task, a traditional machine learning model was implemented using TF-IDF vectorization combined with a linear classifier trained via Stochastic Gradient Descent (SGD). While more simplistic than embedding-based or generative approaches, this baseline provides a useful benchmark for assessing the effectiveness of more complex models.

The training data for this baseline was drawn from the preprocessed DBLP V14 corpus. To prevent evaluation leakage, all paper IDs included in the held-out evaluation sets were excluded from training. Abstracts from the remaining papers were streamed using the ijson library and converted into sparse feature vectors using the TfidfVectorizer, with the vocabulary limited to the 50.000 most frequent terms. This constraint balanced computational efficiency with representational coverage.

Venue labels were normalized and encoded into integer class indices using LabelEncoder,

restricted to venues with at least 100 publications. The resulting feature matrix and label array were used to train an SGDClassifier from the scikit-learn library using default hyperparameters.

The baseline design was loosely inspired by the work of Kobs et al. in their paper called ”Where to Submit? Helping Researchers to Choose the Right Venue” [37]. While the structure of their traditional model was adopted, exact replication was not possible due to missing implementation details. Instead, several decisions were tailored to the specific characteristics of the DBLP V14 dataset and the available computational resources, such as setting the vocabulary cap and handling label imbalance.

### 5.4.2 LLM-Based Recommender

The LLM-Based Recommender was designed to evaluate whether a pretrained large language model, operating without access to external retrieval mechanisms, could infer suitable academic venues directly from a paper’s abstract. This setup relies entirely on the model’s internal knowledge and its ability to semantically generalize across academic language.

For this purpose, the open-source **Mistral** model was deployed locally using the Ollama framework, which allows running language models on consumer-grade hardware without the need for cloud services or external APIs. This local deployment ensured full control over inference, reproducibility and data privacy during evaluation. The model was served via a personal laptop equipped with an NVIDIA GeForce RTX 3050 Ti Mobile GPU.

Prompt construction and interaction with the LLM were managed using the LangChain framework. LangChain was employed as the orchestration layer to manage prompt construction, LLM invocation and response post-processing. Central to this architecture is the use of a LangChain chain object, which is a compositional abstraction that links a structured prompt template with a specific LLM instance to form a callable unit. In other words, when invoked with an input (e.g., an abstract), the chain automatically formats the input according to the prompt, sends it to the LLM for inference and returns the generated response.

In our project, two such chains were defined:

1. **Venue Recommendation Chain**: This chain consisted of a prompt template that asked the LLM to generate a ranked list of ten suitable academic venues based solely on a given abstract. There were two different of these templates, one for the LLM-only recommender and one for RAG recommenders, as shown in the Figures 5.1 and 5.2.

The chain was responsible for formatting the input abstract, invoking the model via Ollama and returning the LLM's output as plain venue names.

```
llm_only_prompt = PromptTemplate.from_template("""
You are an expert research assistant. Based only on the
following paper abstract, suggest the top 10 academic
venues where this paper could be submitted.

Abstract:
{abstract}

Respond with a simple list of full venue names only,
one per line.
""")
```

Figure 5.1: Prompt template for the LLM-only recommender.

```
rag_prompt = PromptTemplate.from_template("""
You are an expert academic assistant. Based on the abstract of
a new paper and a list of similar papers, recommend the most
appropriate publication venues.
Only consider venues where similar papers were actually published.
Respond with a ranked list of 10 academic venue names. Use the
full official name of each venue. Do not explain or comment
on your choices.

New paper abstract:
{abstract}

Similar papers (with venues):
{similar_papers}

Format your response as a simple list:
- Venue Name 1
- Venue Name 2
- Venue Name 3
...
""")
```

Figure 5.2: Prompt template for the RAG recommenders.

2. **Venue Expansion Chain**: This secondary chain was used when the LLM's predicted venues were informal or abbreviated. As shown in the Figure 5.3, it invokes the model with another prompt, asking it to expand or normalize a venue name to its full formal citation form. This ensured reliable semantic matching between predicted and actual venue names during evaluation. It is important to note that, this chain was used exclusively as a part of the evaluation phase to improve the accuracy of metric calculations

and did not play a central role during the venue recommendation process itself.

```
expansion_prompt = PromptTemplate.from_template("""
You are a smart assistant that converts abbreviated or informal venue names
into their standardized full academic names.
Only respond with the formal name used in academic citations, without any
commentary or extra information.
If the name is already complete, return it exactly as is.

Venue name:
"{venue}"
""")
```

Figure 5.3: Prompt template for the Venue Expansion Chain.

Each chain was initialized by combining a PromptTemplate object with the Ollama-backed LLM instance, producing a self-contained module for structured inference. The figure above shows the exact prompt definitions used to implement these chains in practice.

To ensure robustness during large-scale evaluation, each LLM call was wrapped in a thread-safe execution pattern with a configurable timeout (set to 120 seconds). Using Python's concurrent.futures, each chain invocation ran in isolation, allowing the system to gracefully recover from timeouts or errors without crashing. This mechanism was particularly important for the scalability and reliability of the evaluation pipeline, given the high number of sequential inference calls involved.

### 5.4.3 Retrieval-Augmented Generation (RAG)

RAG is a technique that combines information retrieval with generative language models. Rather than relying solely on the model's internal knowledge, RAG pipelines augment the model's input with relevant external documents retrieved based on the user's query. This setup improves factual grounding and domain specificity, especially in knowledge-intensive tasks. In the context of academic venue recommendation, RAG allows the system to ground its predictions in semantically similar research papers, thus enhancing the relevance of the suggested venues.

#### Basic RAG

At first, a straightforward RAG approach is implemented, where the retrieval step is based purely on embedding similarity and no reranking mechanism is applied to adjust or prioritize the selected documents.

For each evaluation instance, the abstract of a paper is encoded using the same sentence embedding model that was used during FAISS index construction. The resulting vector is queried against the FAISS index to retrieve the top 50 nearest neighbors, all of which correspond to valid papers' abstracts. From these, up to 10 unique-venue documents with non-empty abstracts are selected to serve as the retrieval context.

The retrieved context is then embedded into a prompt alongside the new abstract. This prompt instructs the model to recommend 10 academic venues based on the provided similar papers. The language model used for generation is again served locally via Ollama and orchestrated through LangChain. Notably, the prompt is carefully structured to elicit a ranked list of full venue names, without any commentary, justification, or additional output. This controlled format was chosen specifically to simplify our evaluation process, particularly the use of vector-based similarity matching for metric computation. However, in a real-world application, the prompt could be adapted to support more naturalistic or interactive responses, depending on user-facing requirements.

**RAG with Keywords-Augmented Context**

An additional experiment explored the use of keywords as supplementary semantic signals during retrieval. While previous RAG variants relied exclusively on paper abstracts to build the FAISS index, the DBLP dataset also provides author-assigned or system-extracted keywords for many publications. These keywords can encapsulate core research themes in a more concise form than abstracts, potentially guiding retrieval toward more thematically aligned papers.

In this setup, the keyword field was preprocessed and embedded using the same model as the abstracts and titles. For papers that included keywords, their embeddings were combined with abstract embeddings to produce a joint representation. This was done by computing a weighted average of the normalized vectors, where the abstract embedding was given weight $0.9$ and the keyword embedding weight $0.1$. This balance was chosen after experimentation, with the goal of preserving the rich semantic content of abstracts while still incorporating the topical focus provided by keywords. For papers lacking keywords, the abstract embedding alone was used.

Retrieval proceeded in the same way as in the Basic RAG configuration: for each query abstract, the top 50 nearest neighbors were retrieved from the FAISS index and up to 10

unique venues were selected to form the LLM context. The downstream prompting and generation process was unchanged. In this way, the experiment tested whether enriching the representation of documents with keywords could improve contextual grounding and, ultimately, venue recommendation quality.

**RAG with citation-based reranking**

This approach builds upon the basic RAG variant by reusing the same dense retrieval and prompt generation mechanisms, while incorporating a simple reranking heuristic based on citation counts. The goal is to examine the relevance of the retrieved context by favoring more influential papers, measured by their number of citations, under the assumption that such papers are more likely to reflect high-quality publication venues.

As in the baseline RAG setup, each query abstract is embedded using the thenlper/gte-small model and used to perform a similarity search over the FAISS index of paper embeddings. From the top 50 retrieved entries, the top 10 are selected based on descending citation count, rather than raw similarity score. This approach was designed to explore whether emphasizing more influential or widely-cited papers, rather than strictly the closest in embedding space, could yield more informative retrieval contexts and potentially lead to improved venue recommendations.

These reranked entries are used to build the context passed to the language model, using the same prompt template as before. Each context entry includes a paper title and its corresponding venue. The model is then instructed to generate a ranked list of 10 academic venues, strictly based on the provided context.

**RAG with cross-encoder reranking**

In this variant of the RAG pipeline, another reranking strategy is applied using a cross-encoder model. This method aims to refine the selection of contextual documents by directly scoring the semantic relevance between the query abstract and each of the 50 candidates using joint encoding.

The initial retrieval step remains the same. Then, to rerank the retrieved candidates, a cross-encoder model (cross-encoder/ms-marco-MiniLM-L-6-v2) is used. This model receives each candidate abstract along with the query abstract as input pairs and outputs a scalar relevance score. Unlike bi-encoder similarity, which computes dot products of independent

embeddings, the cross-encoder processes both texts jointly, usually allowing it to capture fine-grained semantic relationships. Then, the top 10 papers according to these scores are selected as the final context set for the LLM.

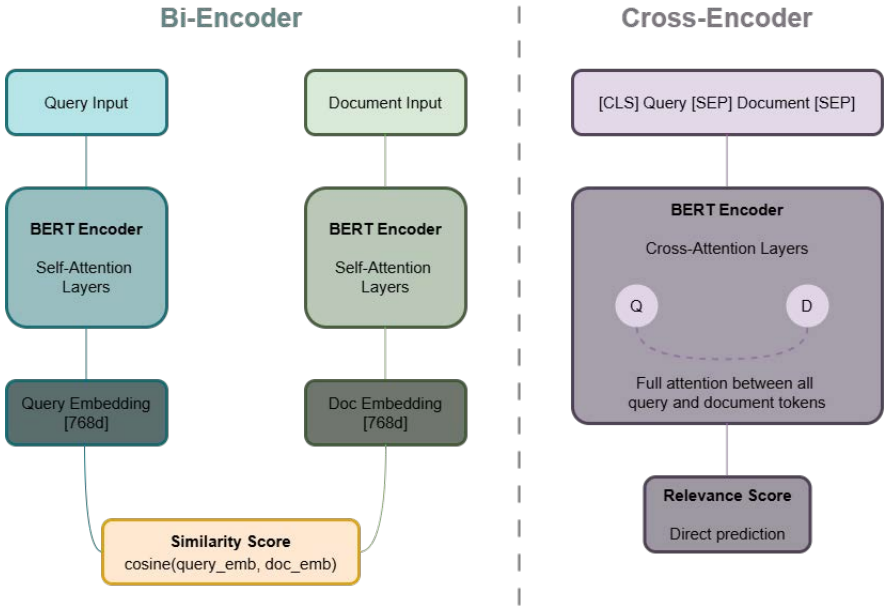


Figure 5.4: Bi-Encoder vs Cross-Encoder

It is important to note that while cross-encoders are more expressive, they are also quite expensive. Unlike bi-encoder models, which can precompute embeddings for large corpora and perform fast vector operations, cross-encoders must process each query-candidate pair jointly at runtime. This makes them considerably slower and less scalable for large-scale retrieval tasks. Moreover, their effectiveness in RAG pipelines depends on multiple factors beyond expressiveness. These include the semantic diversity of the retrieval candidates, the degree to which the cross-encoder has been pre-trained or if it was fine-tuned for a certain topic. The example in Figure 5.4 use BERT as the encoder to illustrate the difference between Bi-Encoders and Cross-Encoders.

**RAG with Impact-Factor reranking**

As a final variant, we introduce a reranking strategy inspired by the concept of journal impact factors. The official Journal Impact Factor (JIF), as maintained by Clarivate, was not available in the DBLP dataset. Instead, we constructed a proxy measure of venue impact based on citation data, using the smoothed mean citation count of all papers published at a given venue. Specifically, for a venue $v$ with $n_v$ papers and $s_v$ total citations, the proxy score

was computed as:

$$\text{Impact}(v) = \frac{\alpha \cdot \mu + s_v}{\alpha + n_v} \tag{5.1}$$

where $\mu$ denotes the global mean citation count across all venues and $\alpha$ is a smoothing parameter that controls shrinkage toward the overall mean. This Bayesian-style smoothing procedure ensures that venues with only a few papers do not receive artificially high or low values due to statistical noise, a well-known issue in bibliometric evaluation.

The use of smoothed citation-based indicators is consistent with practices in the scientometrics literature. Patricia Pérez-Hornero et al. [38] emphasize the instability of citation-based averages for small journals and recommend shrinkage estimators to improve robustness. More recently, Gu Jiaying et al.[39] employed Empirical Bayes adjustments to stabilize journal-level impact metrics in economics. Our formulation follows this rationale, yielding a proxy impact factor that is robust, comparable across venues and interpretable as a prestige signal.

In the pipeline, this proxy impact factor was integrated into the reranking stage of the RAG system. Following the initial FAISS retrieval of the top 50 most similar papers, candidates were reranked according to the following score:

$$\text{FinalScore}(d) = \lambda \cdot \text{Sim}(q, d) + (1 - \lambda) \cdot \text{Impact}(\text{venue}(d)), \tag{5.2}$$

where $\text{Sim}(q, d)$ is the cosine similarity between the query abstract and a candidate document $d$, and $\lambda$ is a weighting hyperparameter. This blending allowed the system to jointly account for semantic closeness and venue prestige, with higher-impact venues favored when similarity scores were otherwise comparable. After experimentation with different values, $\lambda = 0.9$ was chosen for this project.

From the reranked list, the top 10 unique venues were selected to construct the context provided to the LLM. This context was then passed into the same prompt template as in previous RAG variants, instructing the model to generate a ranked list of 10 academic venues. In this way, the system leverages both semantic relevance and an impact-inspired prestige signal, aiming to improve recommendation quality beyond purely similarity-driven retrieval.

# Chapter 6

# Experimental Evaluation

## 6.1 Evaluation Setup

To objectively assess the performance of the venue recommendation models, we established a comprehensive evaluation protocol. The primary evaluation metrics used are **Top-K Accuracy (Hit Rate)**, **Mean Reciprocal Rank (MRR)** and **Mean Venue Overlap (MVO)**. These metrics are well-suited for ranked recommendation tasks, as they capture different dimensions of predictive performance: correctness, ranking quality and semantic alignment.

As previously described, Top-K Accuracy evaluates whether the correct venue appears within the top-K predicted venues. MRR measures the reciprocal of the rank at which the correct venue first appears in the output list, placing greater weight on higher-ranked correct predictions. MVO, on the other hand, is a custom metric designed to assess the semantic closeness of predicted venues to the actual venue. This allows the evaluation to reflect cases where the predicted venue is not exactly correct, but still thematically or topically similar to the true venue.

For the evaluation of the system, a collection of **10 evaluation sets** was used, with each one consisting of **100 abstracts** randomly sampled from the preprocessed DBLP V14 corpus. All evaluation instances were held out from any stage of model training or embedding construction to prevent data leakage and ensure the validity of the results.

Each evaluation instance includes the abstract of a paper and its normalized ground truth venue label. In order to allow for a fair comparison, all model outputs were also normalized using the same venue mapping routine that was applied during preprocessing. This step was especially important for predictions generated by LLMs, as they could contain informal or

abbreviated forms of venue names. To ensure consistency, both predicted and actual venues were embedded with the GTE-small model. Cosine similarity was utilized to determine semantic proximity during MVO calculations and to address label matches for Top-K and MRR in cases of string ambiguity.

## 6.2 TF-IDF+SGD Classifier Results

The TF-IDF+SGD Classifier model served as the foundational baseline for the venue recommendation task, relying exclusively on sparse lexical features extracted from paper abstracts. Using scikit-learn's TfidfVectorizer, each abstract was encoded into a high-dimensional feature vector, followed by supervised training of a linear classifier via stochastic gradient descent.

The model performed poorly in practice. The Top-1 Accuracy was just **2%** and even at Top-10, it only reached **10%**. These results clearly indicate that the model struggled to capture the semantic complexity and domain-specific variation present in academic abstracts. Unlike semantic embedding methods, TF-IDF lacks the capacity to account for synonyms, conceptual similarity, or contextual nuance, which is an essential capability when distinguishing between closely related venues.

Another factor contributing to the weak performance is the scale and diversity of the dataset. The corpus contained approximately 2.5 million abstracts mapped to nearly 3,000 distinct venues, creating a highly challenging classification problem. Compared to datasets of similar size but far fewer target categories, this level of label diversity substantially increased the difficulty of achieving accurate predictions using a purely lexical baseline.

| **Top-3** | **Top-5** | **Top-7** | **Top-10** |
|---|---|---|---|
| 2.00 | 5.20 | 7.60 | 10.00 |

Table 6.1: TF-IDF+SGDClassifier performance metrics.

## 6.3 LLM-Only Recommender Results

The LLM-only approach evaluated the standalone capability of a large language model, specifically Mistral, deployed locally using Ollama, to recommend academic venues based

solely on the abstract of a paper, without access to external context or retrieval support.

Compared to the TF-IDF baseline, the LLM-only model demonstrated a clear performance improvement. As it is shown in the table below, its MRR was **0.20**, indicating that correct venues tended to appear mid-way in the predicted lists. However, the MVO was quite low at **1.51**, suggesting limited semantic proximity between predicted and actual venues on average. This shows that while LLMs provide a major leap over purely lexical baselines, they still lack grounding in domain-specific literature, motivating the implementation of RAG.

| Top-3 | Top-5 | Top-7 | Top-10 | MRR | MVO |
|---|---|---|---|---|---|
| 21.51 | 26.63 | 29.54 | 32.76 | 0.20 | 1.51 |

Table 6.2: LLM-only performance metrics.

As seen in Figure 6.1, the model is able to generate plausible chemistry-related venues, but many of them do not exactly align with the ground truth (*"Journal Of Chemical Information And Computer Sciences"*), reflecting the limitations of relying solely on the LLM's internal knowledge.

```
1. Fuel Science and Technology International (Fuels)
2. Combustion and Flame
3. Journal of Physical Chemistry A
4. Energy & Fuels
5. International Journal of Hydrogen Energy
6. Journal of Molecular Liquids
7. Catalysis Today
8. Reaction Kinetics and Dynamics
9. Journal of Computational Chemistry
10. The Journal of Chemical Physics
```

Figure 6.1: Example suggestions of the LLM-only recommender.

## 6.4 RAG Evaluation

This section presents the results and analysis of the RAG models. Each variant integrates retrieved documents into the LLM's prompt to improve semantic grounding and reduce hallucinations. The evaluation follows the same protocol as previously described, using Top-K

Accuracy, MRR and MVO as core metrics. Each variant is assessed across 10 evaluation sets and compared against both the TF-IDF baseline and the LLM-only model.

### 6.4.1 Basic RAG

The first variant of the RAG framework implements a straightforward pipeline, in which the retrieval step is based solely on semantic similarity between the input abstract and pre-indexed paper embeddings. No reranking mechanism is applied in this variant, meaning that the retrieved context is determined exclusively by FAISS-based approximate nearest neighbor search.

For each test instance, the abstract is embedded using the same GTE-small model used during index construction. The resulting vector is used to query the FAISS index, returning the top 50 nearest neighbors. From these, up to 10 documents with distinct venues are selected to form the context passed to the language model. The context is incorporated into a prompt that asks the LLM to recommend ten academic venues. The generation process is managed via LangChain and served locally using Ollama and the Mistral model.

Looking at the results that this setup yielded, a major improvement over the previous 2 baselines is clear. The MRR was measured at **0.28**, indicating moderate effectiveness in placing the correct venue near the top of the predicted list. Additionally, the MVO, which quantifies the average number of venues shared between the predicted list and the retrieved context, was **6.59**, suggesting a reasonably high alignment between the model's output and the provided context.

| **Top-3** | **Top-5** | **Top-7** | **Top-10** | **MRR** | **MVO** |
|---|---|---|---|---|---|
| 31.84 | 37.98 | 42.99 | 49.63 | 0.28 | 6.59 |

Table 6.3: Basic RAG performance metrics.

As seen in Figure 6.2, the Basic RAG method successfully ranked the correct venue (*"Journal of Chemical Information and Computer Sciences"*) at the top of its suggestions, clearly demonstrating the benefit of retrieval-based grounding over the LLM-only approach.

```
1. Journal of Chemical Information and Computer Sciences
2. Computers & Chemical Engineering
3. International Conference on Computational Science
4. RTA (Reliable Software Technologies Symposium)
5. Knowledge Based Systems
6. Entropy
7. Computers & Chemistry
8. Journal of Experimental Algorithmics
9. Electronic Notes in Theoretical Computer Science
10. Journal of Chemical Information and Modeling
```

Figure 6.2: Example suggestions of the Basic RAG recommender.

### 6.4.2 RAG with Keywords-Augmented Context

This experiment extended the Basic RAG setup by enriching the paper embeddings with keyword information in addition to abstracts. As described in the previous Chapter, keyword and abstract embeddings were combined into a joint representation, with a weighting scheme that placed greater emphasis on the abstract. All other steps of the pipeline, including retrieval, prompt construction and LLM generation, remained identical to the Basic RAG method.

The results were very close to those of Basic RAG, with only marginal differences across evaluation sets. Top-10 accuracy reached **48.54%**, and the MRR was **0.27**, essentially on par with the baseline. The Mean Venue Overlap was **6.51**, suggesting that the inclusion of keywords neither harmed nor significantly improved contextual grounding. These findings indicate that while keywords may offer complementary topical signals, much of their information overlapped with what was already captured in abstracts, which limited their additional impact on retrieval and recommendation performance.

| Top-3 | Top-5 | Top-7 | Top-10 | MRR | MVO |
|---|---|---|---|---|---|
| 31.44 | 37.87 | 42.91 | 48.54 | 0.27 | 6.51 |

Table 6.4: RAG with Keywords-Augmented Embeddings performance metrics.

In fact, for the illustrative example run shown earlier in Figure 6.2, this variant produced the exact same ranked venue list as the Basic RAG method. This outcome reinforces the quantitative findings, highlighting that the addition of keywords did not alter the system's recommendations in practice for this case.

### 6.4.3 RAG with Citation-Based Reranking

The second variant of the RAG pipeline introduced a reranking strategy based on citation count. After retrieving the top 50 candidates from the FAISS index, the documents were reordered by descending citation volume and the top 10 were selected to form the retrieval context. This heuristic assumes that highly cited papers are more likely to represent established, reputable venues and thus could guide the LLM toward better recommendations.

However, the results show that this reranking strategy underperformed compared to the basic RAG configuration. The Top-3, Top-5 and Top-10 accuracy scores were consistently lower. The MRR also declined to **0.24**, indicating that the correct venues appeared later in the predicted lists on average. While the MVO remained relatively high at **6.01**, suggesting that the semantic proximity between predicted and actual venues was not entirely lost, the overall effectiveness of this approach was limited. It becomes obvious that citation volume is not always a good proxy for contextual alignment, especially compared to semantic similarity.

| Top-3 | Top-5 | Top-7 | Top-10 | MRR | MVO |
|---|---|---|---|---|---|
| 27.80 | 35.65 | 39.78 | 43.71 | 0.24 | 6.01 |

Table 6.5: RAG with Citation-Based Reranking performance metrics.

Considering the same example as before, we can see in Figure 6.3 that the model produced reasonable venue suggestions, including several chemistry and computational science outlets. However, the correct venue was missed entirely, which is consistent with the quantitative results showing weaker accuracy compared to Basic RAG.

```
1. Computers & Chemistry
2. Computers & Chemical Engineering
3. RTA (Reliable Software Technologies Symposium)
4. Journal of Experimental Algorithmics
5. Electronic Notes in Theoretical Computer Science
6. Journal of Computer-Aided Molecular Design
7. Computer Physics Communications
8. Knowledge Based Systems
9. Journal of Computational Chemistry
10. International Conference on Computational Science (ICCS)
```

Figure 6.3: Example suggestions of the RAG with Citation-Based Reranking recommender.

### 6.4.4 RAG with Cross-Encoder Reranking

The third variant incorporated a more sophisticated reranking mechanism using a cross-encoder model (cross-encoder/ms-marco-MiniLM-L-6-v2). After retrieving the top 50 abstracts using FAISS, each candidate was paired with the query abstract and jointly scored by the cross-encoder for relevance. The 10 most semantically aligned documents were then used to construct the context provided to the LLM.

In terms of retrieval quality and downstream recommendation performance, this variant showed competitive results. Top-3 accuracy improved to **32.55%** and Top-10 accuracy reached **45.22%**, exceeding the citation-based variant. The MRR score recovered to **0.28**, matching the performance of the basic RAG configuration, indicating improved ranking quality. However, the MVO had a value of **6.32**, suggesting that while predictions were more likely to include the correct venue, the reranker prioritized semantically relevant papers at some cases, including those from niche venues.

These findings suggest that cross-encoder reranking succeeds in refining the local relevance of retrieval candidates, helping the LLM focus on more appropriate examples. However, the drop in MVO may reflect a tendency toward more topically precise but less known venue recommendations. Moreover, its computational cost makes it less practical for large-scale deployment.

| **Top-3** | **Top-5** | **Top-7** | **Top-10** | **MRR** | **MVO** |
|---|---|---|---|---|---|
| 32.55 | 38.28 | 41.20 | 45.22 | 0.28 | 6.32 |

Table 6.6: RAG with Cross-Encoder Reranking performance metrics.

As shown in Figure 6.4, the predictions included the correct venue and several closely related alternatives, but the overall list was narrower in scope compared to the Basic RAG method.

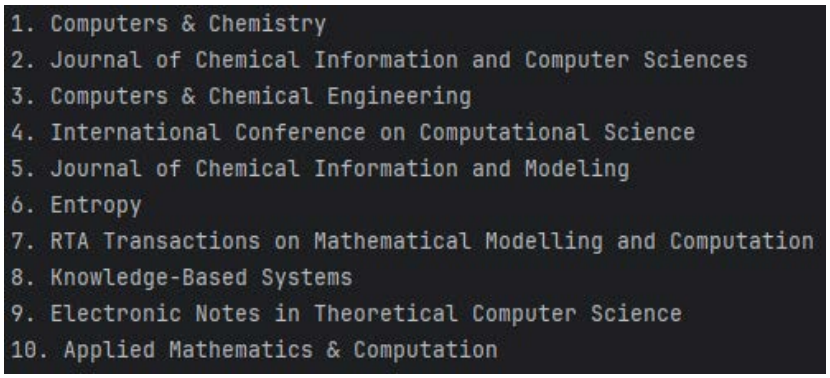


Figure 6.4: Example suggestions of the RAG with Cross-Encoder Reranking recommender.

### 6.4.5 RAG with Impact-Factor Reranking

The final variant of the RAG pipeline incorporates a reranking mechanism that combines semantic similarity with an estimated impact factor score for venues. Since explicit journal impact factors were not available in the DBLP V14 dataset, proxy values were computed using smoothed averages of citation counts per venue, inspired by established bibliometric approaches that recommend shrinkage estimators to stabilize impact factor estimates for smaller journals. This proxy score was normalized to the range $[0, 1]$ and associated with each venue in the metadata.

The results of this method show performance very close to the Basic RAG approach. Top-10 accuracy reached **48.54%**, with an MRR of **0.26**. The Mean Venue Overlap was **6.56**, slightly lower than the Basic RAG variant. These findings suggest that while proxy impact factor information does not dramatically improve overall accuracy, it provides a modest bias toward established venues without degrading contextual alignment.

| Top-3 | Top-5 | Top-7 | Top-10 | MRR | MVO |
|---|---|---|---|---|---|
| 30.35 | 37.79 | 43.71 | 48.54 | 0.26 | 6.56 |

Table 6.7: RAG with Impact-Factor Reranking performance metrics.

As illustrated in Figure6.5, this method successfully retained the ground truth venue within the predicted list and tended to include prestigious publication outlets. However, the improvements over purely semantic retrieval remained marginal, indicating that impact factor–like adjustments may complement but not replace embedding-based similarity in venue recommendation tasks.

```
1. Journal of Chemical Information and Computer Sciences
2. Computers & Chemical Engineering
3. International Conference on Computational Science
4. RTA (Reliable Software Technologies Symposium)
5. Knowledge Based Systems
6. Journal of Computational Chemistry
7. Computers & Chemistry
8. Entropy
9. Journal of Experimental Algorithmics
10. J Cheminformatics
```

Figure 6.5: Example suggestions of the RAG with Impact-Factor Reranking recommender.

In the table 6.8 we have a consolidated view of all methods. It highlights the highest value of each evaluation metric used. In every table, Top-K values are shown in percentage.

| **Method** | **Top-3** | **Top-5** | **Top-7** | **Top-10** | **MRR** | **MVO** |
|---|---|---|---|---|---|---|
| TF-IDF + SGDClassifier | 2.00 | 5.20 | 7.60 | 10.00 | – | – |
| LLM-Only | 21.51 | 26.63 | 29.54 | 32.76 | 0.20 | 1.51 |
| RAG (Abstracts) | 31.84 | 37.98 | 42.99 | **49.63** | **0.28** | **6.59** |
| RAG + Keywords Augmentation | 31.44 | 37.87 | 42.91 | 48.54 | 0.27 | 6.51 |
| RAG + Citation Rerank | 27.80 | 35.65 | 39.78 | 43.71 | 0.24 | 6.01 |
| RAG + Cross-Encoder Rerank | **32.55** | **38.28** | 41.20 | 45.22 | **0.28** | 6.32 |
| RAG + Impact Factor Rerank | 30.35 | 37.79 | **43.71** | 48.54 | 0.26 | 6.56 |

Table 6.8: Performance comparison across all methods and metrics.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusions

The experimental evaluation demonstrated the importance of both semantic modeling and contextual grounding in academic venue recommendation. The TF-IDF baseline highlighted the limitations of purely lexical methods, achieving only minimal accuracy across thousands of venues. In contrast, the LLM-only model leveraged pretrained semantic knowledge to make more plausible predictions, achieving markedly higher Top-K accuracy. However, its predictions often suffered from weak grounding: the model could generate venues that were linguistically or thematically related but not actually aligned with the source literature. This lack of context led to a tendency toward hallucinations and limited semantic overlap with the true relevant venues.

The introduction of RAG directly addressed these shortcomings. By incorporating semantically similar documents retrieved from the FAISS index into the prompt, the LLM was able to anchor its reasoning in domain-relevant examples. This grounding reduced hallucinations, improved factual alignment and enabled the model to produce venue recommendations that were not only more accurate but also more consistent with the established structure of academic publishing. Compared to the LLM-only setup, the basic RAG variant nearly doubled Top-10 accuracy and raised the MVO from 1.51 to 6.59, showing that the predicted venues were much more closely tied to the retrieved evidence.

Another experiment incorporated keywords into the document embeddings, combining them with abstracts to create a joint representation. This was intended to capture both detailed descriptions of content and concise topical labels. The results were very close to the Basic

RAG setup, with Top-K accuracy and MRR remaining essentially unchanged. The MVO was measured at 6.51, slightly lower than the value of the Basic RAG method. This outcome indicates that while keywords introduced additional topical cues, they were largely redundant with the information already present in abstracts and did not provide a discriminative advantage for venue recommendation.

Among the following three strategies, results were mixed. Citation-based reranking underperformed, as citation volume did not reliably reflect topical relevance. Cross-encoder reranking improved Top-3 accuracy and ranking quality, but at the cost of semantic diversity and efficiency. A final variant introduced an score resembling impact factor, computed from citation statistics, to approximate the perspective of researchers who often consider venue prestige alongside topical fit when deciding where to publish. However, this method also showed only marginal differences compared to the basic RAG setup and in some evaluation sets performance was slightly worse. Overall, these findings suggest that while reranking strategies can capture additional dimensions such as influence or prestige, they do not align as strongly with semantic relevance, and therefore provide limited benefits over the basic RAG pipeline.

In summary, the evaluation demonstrates that retrieval augmentation is essential to overcoming the limitations of standalone LLMs in academic venue recommendation. By grounding predictions in retrieved literature, RAG reduces hallucinations, improves factual alignment, and delivers more accurate and semantically coherent recommendations. The limited benefit of reranking strategies further highlights the dominant role of semantic similarity, suggesting that simple, well-designed retrieval pipelines may be as effective as more elaborate ranking schemes. More broadly, the findings underscore the niche nature of venue recommendation, where decisions are shaped by a mix of topical fit, audience and community reputation, which are factors not fully represented in the dataset and therefore beyond the scope of this study.

## 7.2 Future Work

While the proposed framework for academic venue recommendation has demonstrated promising results, there are several avenues for further improvement and expansion.

### Embedding and Model Improvements

One of the most important avenues for future improvement is the embedding model used. Higher semantic fidelity embeddings could be produced by more potent alternatives or refined domain-specific models, even if the current system uses the thenlper/gte-small encoder for both abstracts and venue names. Furthermore, task-specific fine-tuning may be useful for cross-encoder reranking, especially if a small supervised dataset of abstract-venue pairings can be built to inform relevance scoring.

Similarly, future implementations might test out more sophisticated language models, including larger versions of Mistral, LLaMA, or Mixtral, which could enhance the quality of both context-informed and zero-shot recommendations.

### Retrieval Enhancements

Another step for improvement is the enhancement of the retrieval phase. The introduction of hybrid retrieval, which combines sparse retrieval or dense embeddings with symbolic filters, may increase relevance, particularly for specialized or multidisciplinary themes. Moreover, adaptive retrieval techniques could be used, in which the complexity or specificity of the query abstract determines the quantity or kind of documents that are retrieved. A further direction could involve constructing and exploiting heterogeneous graphs that link papers, authors, venues and citation relationships. Such graph structures would enable retrieval strategies that leverage not only semantic similarity but also structural signals from the broader scholarly network, potentially uncovering connections that embeddings alone may miss.

### Reranking Optimization

Despite the fact that the cross-encoder reranking model increased MRR and Top-3 accuracy, its computational cost still prevents real-time use. Future research could examine more scalable reranking options, such as:

- Multi-stage reranking pipelines, such as shallow cross-encoder refinement after bi-encoder trimming.
- Author and co-authorship graphs can integrate signals from collaboration networks and help prioritize venues where similar research communities are active.

- Citation network centrality can be used to rank papers published in venues that occupy central positions in citation graphs higher, reflecting influence beyond raw citation counts.
- Topic–venue alignment can be achieved by applying topic models (e.g., LDA or neural topic embeddings) to assess whether a venue's historical publications match the latent topics of the query abstract.
- Knowledge graphs can explicitly link papers, venues and fields of study, enabling graph-based reranking strategies that capture relationships missed by embeddings alone.

**Evaluation Enhancements**

Lastly, it is also important to evolve the evaluation phase. The evaluation pipeline used in this project was designed for reliability and modularity but remains constrained in scale. Future work should include:

- Larger evaluation sets (e.g., 10,000+ papers).
- Human-in-the-loop assessment, where domain experts judge the quality and relevance of predicted venues.
- Temporal validation to assess whether the system generalizes well to newly published papers.

Additionally, metric design could be expanded beyond Top-K and MRR to incorporate coverage, novelty, and diversity, which are critical in real-world academic recommendation settings.

# Bibliography

[1] Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete. Publication venue recommendation using profiles based on clustering, 2024.

[2] Hiep Phuc Luong, Tin Huynh, Susan Gauch, and Kiem Hoang. Exploiting social networks for publication venue recommendations. In Ana L. N. Fred and Joaquim Filipe, editors, *KDIR 2012 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Barcelona, Spain, 4 - 7 October, 2012*, pages 239–245. SciTePress, 2012.

[3] Shuo Yu, Jiaying Liu, Zhuo Yang, Zhen Chen, Huizhen Jiang, Amr Tolba, and Feng Xia. Pave: Personalized academic venue recommendation exploiting co-publication networks. *Journal of Network and Computer Applications*, 104:38–47, 2018.

[4] Andreea Iana and Heiko Paulheim. Graphconfrec: A graph neural network-based conference recommender system. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, page 90–99. IEEE, September 2021.

[5] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. Specter: Document-level representation learning using citation-informed transformers, 2020.

[6] Vaios Stergiopoulos, Michael Vassilakopoulos, Eleni Tousidou, and Antonio Corral. An academic recommender system on large citation data based on clustering, graph modeling and deep learning. *Knowledge and Information Systems*, 66:1–34, 04 2024.

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text, 2019.

[8] Benjamin Wolff, Eva Seidlmayer, and Konrad U. Förstner. *Enriched BERT Embeddings for Scholarly Publication Classification*, page 234–243. Springer Nature Switzerland, 2024.

[9] Elliot Bolton, Betty Xiong, Vijaytha Muralidharan, Joel Schamroth, Vivek Muralidharan, Christopher D. Manning, and Roxana Daneshjou. Assessing the potential of mid-sized language models for clinical qa, 2024.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[11] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine, 2024.

[12] Ahmet Yasin Aytar, Kemal Kilic, and Kamer Kaya. A retrieval-augmented generation framework for academic literature navigation in data science, 2024.

[13] Xianrui Zhong, Bowen Jin, Siru Ouyang, Yanzhen Shen, Qiao Jin, Yin Fang, Zhiyong Lu, and Jiawei Han. Benchmarking retrieval-augmented generation for chemistry, 2025.

[14] Anoushka Dasgupta and Shideh Mehr. Enhanced mnb method for spam e-mail/sms text detection using tf-idf vectorizer. *American Journal of Mathematical and Computer Modelling*, 9:1–8, 04 2024.

[15] Bastian Schäfermeier, Gerd Stumme, and Tom Hanika. Towards explainable scientific venue recommendations, 2021.

[16] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[19] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[21] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[22] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023.

[23] Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions, 2024.

[24] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[25] Chuyuan Wei, Ke Duan, Shengda Zhuo, Hongchun Wang, Shuqiang Huang, and Jie Liu. Enhanced recommendation systems with retrieval-augmented large language model. *Advances in Artificial Intelligence Research*, 82:1147–1173, 02 2025.

[26] Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in llms, 2024.

[27] Eugene Garfield. The history and meaning of the journal impact factor. *JAMA : the journal of the American Medical Association*, 295:90–3, 02 2006.

[28] Paul Wouters, Cassidy Sugimoto, Vincent Larivière, Marie McVeigh, Bernd Pulverer, Sarah de Rijcke, and Ludo Waltman. Rethinking impact factors: better ways to judge a journal. *Nature*, 569:621–623, 05 2019.

[29] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. On sampling top-k recommendation evaluation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, page 2114–2124. ACM, August 2020.

[30] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. Assessing ranking metrics in top-n recommendation. *Inf. Retr.*, 23(4):411–448, June 2020.

[31] AMiner Team. Dblp-citation-network v14 dataset. `https://open.aminer.cn/open/article?id=655db2202ab17a072284bc0c`, 2023.

[32] Michael Stonebraker and Lawrence A. Rowe. The design of postgres. *SIGMOD Rec.*, 15(2):340–355, June 1986.

[33] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus, 2017.

[34] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock,

Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[35] LangChain Inc. Langchain documentation. `https://python.langchain.com`, 2025. Accessed: 24 August 2025.

[36] Nagwa Elmobark. A comparative analysis of python text matching libraries: A multilingual evaluation of capabilities, performance, and resource utilization,. *International Journal of Environment Engineering and Education*, 7:48–60, 04 2025.

[37] Konstantin Kobs, Tobias Koopmann, Albin Zehe, David Fernes, Philipp Krop, and Andreas Hotho. Where to submit? helping researchers to choose the right venue. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 878–883, Online, November 2020. Association for Computational Linguistics.

[38] Patricia Pérez-Hornero, José Pablo Arias-Nicolás, Antonio A. Pulgarín, and Antonio Pulgarín. An annual jcr impact factor calculation based on bayesian credibility formulas. *Journal of Informetrics*, 7(1):1–9, 2013.

[39] Jiaying Gu and Roger Koenker. Ranking and selection from pairwise comparisons: Empirical bayes methods for citation analysis. *AEA Papers and Proceedings*, 112:624–29, May 2022.