Large Language Models in Information Retrieval

Rui Botelho
1191041@isep.ipp.pt

Olívia Puig
1242532@isep.ipp.pt

Bernardo Ferreira
1190437@isep.ipp.pt

Vasco Silva
1240499@isep.ipp.pt

***Abstract*—This paper investigates the convergence of Information Retrieval (IR) and Large Language Models (LLMs), highlighting their transformative impact on modern information systems. We begin by defining IR as the process of extracting valuable information from large datasets, a core element of contemporary digital life. IR powers search engines, recommendation systems, and retrieval processes in research and business contexts. We then delve into LLMs, such as GPT and BERT, which have revolutionized Natural Language Processing (NLP) through their ability to process and generate human language. These models are at the heart of technologies like text summarization, question-answering, and conversational AI. The paper explores the implications of LLMs for IR, particularly how their capacity to understand semantic meaning enhances traditional keyword-based search systems. We examine how LLMs improve IR practices, from refining ranking algorithms to enhancing query understanding and document retrieval.**

***Index Terms*—Information Retrieval, Large Language Models, machine models, text-based data, Natural Language Processing, GPT, BERT, text summarization, question-answering, conversational AI, keyword matching, semantic meaning, ranking algorithms, query comprehension, document retrieval.**

I. Introduction

In this paper, we explore the intersection of Information Retrieval (IR) and Large Language Models (LLMs), examining how these technologies are reshaping modern information systems.

First, we define IR—the process of finding valuable information from massive sets. IR is fundamental to contemporary digital existence, powering search engines like Google, recommendation algorithms for streaming platforms, and even retrieval in research and business environments.

Then there are LLMs—machine models that are trained on vast amounts of text-based data to read and write human languages. Models such as GPT and BERT, for instance, have transformed Natural Language Processing (NLP) and form the foundation for activities such as text summarization, question-answering, and conversational AI.

Why then are LLMs of concern to IR? Keyword matching is what traditional IR systems employ, but LLMs have the potential to translate into semantic meaning, thus rendering search more accurate and pertinent. This seminar will examine how LLMs are transforming IR, from ranking algorithm optimization to improving query comprehension and document retrieval.

II. BackGround

A. Information Retrieval

IR is a broad and fundamental field in both computer science and information science that is essential to modern-day search and data processing technologies. At its core, IR is concerned with finding relevant data in response to user queries from vast collections of unstructured or semi-structured data [7]. These data collections could be anything from text documents, web pages, digital libraries, multimedia content, or even large datasets like social media posts or scientific papers [1].

The fundamental purpose of an IR system is to meet a user's information need by returning relevant content from a potentially vast database. These systems are designed to locate, retrieve, and rank information items based on their relevance to a given query [12]. For instance, when a user enters a search query into a web search engine like Google, the IR system processes the query and returns a list of web pages that it deems most relevant based on various algorithms [15]. These results are typically sorted in order of relevance, which is determined by the system's ranking function [7]. The goal is for the user to find the most relevant information in as little time as possible.

A prime example of IR in practice is the web search engine. When you search for a topic, say "climate change impact on agriculture," the search engine scans billions of web pages in its index and returns results that are most likely to answer your query. The underlying IR system uses sophisticated models (such as PageRank, semantic analysis, and machine learning) to rank these pages in order of relevance [2].

Similarly, digital library catalogs also use IR systems to enable users to retrieve relevant books, articles, or other resources [1]. For example, when a researcher queries a digital library like JSTOR or IEEE Xplore for articles related to "quantum computing," the IR system sifts through thousands of scholarly papers to provide a ranked list of documents that most closely match the user's search terms or research interests.

IR systems are not just limited to simple keyword searches. They can incorporate advanced techniques like NLP to better understand the user's query [10]. Moreover, recommendation systems powered by IR are another use case where IR principles are applied to provide users with personalized suggestions [14]. These systems, such as those used by Netflix, Amazon, and Spotify, don't rely on an explicit user query. Instead, they recommend items based on patterns in user behavior

or historical data, applying IR techniques like collaborative filtering or content-based filtering [8].

In the following sections, we will explore the key differences between concepts, the techniques used, and the types of IR systems available today. We will also discuss recent advancements in IR and their impact on industries like web search, digital libraries, and recommendation engines.

a) Information Retrieval vs. Data Retrieval: In many discussions surrounding data search systems, there is often a distinction made between IR and data retrieval systems. At a high level, this differentiation is framed based on the type of data being retrieved and the underlying structure of that data [9]. Typically, IR systems are considered to be systems that retrieve unstructured data, such as text documents, web pages, or multimedia content. On the other hand, data retrieval systems are thought to deal with structured data—information that resides in relational databases or similar well-defined data formats [3].

Some modern IR systems use relational models or hybrid approaches, where elements of both unstructured and structured retrieval are combined [4]. As a result, structured data retrieval can be considered a specialized form of IR, with both fields focusing on the common goal of retrieving relevant information in response to user queries.

b) Information Retrieval vs. Recommender Systems: IR and recommender systems are distinct, though they share some similarities. IR systems typically rely on user-generated queries to retrieve relevant information from a large collection of unstructured data, such as text or multimedia [6]. In contrast, recommender systems do not depend on explicit user queries but instead suggest items based on user preferences, behaviors, or interactions. While both systems aim to provide relevant content, recommender systems primarily focus on predicting what a user might like based on past actions, using techniques like collaborative filtering or content-based filtering [8]. Therefore, while recommender systems can be seen as a subset of information filtering within IR, they are a separate approach, differing in how and when they interact with the user.

How IR Systems Work

IR systems are designed to efficiently find relevant information from large collections of data in response to user queries. The way these systems work largely depends on the specific model they use to represent and retrieve information. Key techniques like indexing, weighting, and relevance feedback are commonly employed across various IR models to improve the accuracy and relevance of search results.

1) Indexing: Indexing is a foundational technique in IR systems, and it involves creating a structured representation of the content within a set of documents. The concept of indexing in IR is akin to the index found at the back of a book, where key terms are listed alongside page numbers to help readers quickly locate relevant content. In IR, an inverted index (or simply, an index) is used, where each term in a document collection is linked to the set of documents in which it appears. This data structure is designed to speed up the retrieval process, enabling the system to efficiently locate documents that contain specific terms. [7]

2) Weighting: Weighting is another crucial step in IR systems that determines how important or relevant each term is to a query. Not all terms in a document carry the same significance, and the weight assigned to a term influences how the system ranks documents in response to a query. One of the most widely used weighting techniques are:

- **Term Frequency-Inverse Document Frequency (TF-IDF)**: Helps to quantify the importance of a term by considering both its frequency within a specific document and its frequency across the entire collection of documents [2]. The higher the frequency of a term in a document relative to the overall document set, the higher its weight. This ensures that terms which are unique or significant within a document are given more importance, while common terms that appear frequently across the collection are down-weighted.
- Other advanced weighting techniques like **Singular Value Decomposition (SVD)** and **Latent Semantic Analysis (LSA)** are used to capture underlying patterns and relationships between terms and documents [8]. These techniques are particularly useful for dimensionality reduction and for uncovering hidden semantic relationships between words that might not be immediately obvious through simple term frequency analysis.

3) Relevance Feedback: Relevance feedback is a technique used to refine search results and improve the quality of IR by incorporating user interaction. After a user submits a query and receives an initial set of search results, relevance feedback allows the user to indicate which documents are relevant or irrelevant to their information need. Based on this feedback, the IR system adjusts the ranking of documents, reweights the terms, and presents a more refined set of results [7]. There are two main types of relevance feedback:

- **Explicit feedback:** users directly rate or mark the relevance of the returned documents, providing clear signals for the system to adjust its rankings. For example, a user may click on checkboxes or rate documents with a thumbs-up or thumbs-down to indicate their relevance.
- **Implicit feedback:** The system infers relevance based on user behavior, such as which links are clicked or how long a user spends reading a particular document. This type of feedback is often used when users do not provide direct input.

An extension of relevance feedback is pseudo-relevance

feedback, where the system assumes that the top documents returned for an initial query are relevant, even if the user hasn't provided explicit feedback. The system then analyzes these documents to extract common features and adjust the original query to better match the retrieved content.

Types of Information Retrieval Models

There are various types of IR models, each offering different methodologies for retrieving relevant documents.

- **Boolean Model**: The Boolean model is one of the simplest and most straightforward IR techniques. It relies on a binary approach to document retrieval, using logical operators like "AND," "OR," and "NOT" to process search queries. For example, a query such as "jazz AND dancing" will retrieve only those documents that contain both the terms "jazz" and "dancing." The Boolean model works by determining the presence or absence of query terms within documents, and it doesn't allow for partial matches or ranking of documents based on relevance. This means that a document either meets the query criteria or it does not.
  While Boolean models are easy to implement, they have some significant limitations as they do not account for the frequency of terms in a document, nor do they provide a way to rank documents by relevance. To address the issue of morphological variations (e.g., "dance," "dances," "dancer"), techniques such as stemming and lemmatization can be applied. [39].

- **Algebraic Models:** The algebraic model, by contrast, allows for partial matching, which addresses the limitations of Boolean retrieval. One of the most prominent algebraic models is the **vector space model**, which represents documents and queries as vectors in a multi-dimensional space. Each index term in a document is treated as a feature, and the query is similarly represented as a vector. The model computes the similarity between the query and each document by calculating the proximity of their vectors in the space.
  A popular metric used to measure the proximity between vectors is **cosine similarity**, which computes the cosine of the angle between two vectors. The formula for cosine similarity is:
  $$\text{Cosine Similarity} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$$
  where $\mathbf{x}$ and $\mathbf{y}$ represent the vectors of two documents. A higher cosine similarity score indicates greater similarity between the two documents. Other metrics, like the Jaccard index or dot product, can also be used in algebraic models to measure similarity.

- **Probabilistic Models:** The probabilistic model takes a different approach by considering the likelihood that a document is relevant to a given query. The core assumption of probabilistic models is that there exists an ideal set of documents that should be retrieved in response to any query, although this ideal set is not directly known. Instead, probabilistic models estimate the probability that a given document belongs to this ideal set based on factors like term presence, term frequency, and term co-occurrence. For instance, they might take into account how often terms co-occur within documents or how frequently a term appears across the entire collection. These factors are then used to calculate the likelihood that a document matches the user's query. One of the first probabilistic IR models, the binary independence model (BIM), only considers the presence or absence of terms and does not factor in term frequency, instead now there are more sophisticated probabilistic models, like those incorporating Latent Dirichlet allocation (LDA), consider additional factors like term co-occurrence, allowing them to capture deeper semantic relationships within documents [42].

4) Recent Research and Challenges: Web search engines, one of the most prominent applications of IR, often perpetuate social biases, including racial and gender-based biases, through algorithms like PageRank. Research has shown that these algorithms can skew search results, disproportionately representing certain groups while marginalizing others. To mitigate these biases, techniques such as negative sampling, which involves selecting counteracting training data, and bias-aware algorithms, which apply penalties to biased results, have been explored.

B. Large Language Models

In recent years, the field of NLP has undergone a transformative change driven primarily by the advent of LLMs and the underlying Transformer architectures. These models, including well-known examples such as GPT, BERT, and T5, have dramatically expanded the scope of what machines can understand and generate in human language. This chapter provides an in-depth overview of LLMs, discussing their high-level operational principles—specifically, the pretraining and fine-tuning paradigms—and then delves into the role of Transformer architectures. By comparing transformers with earlier NLP architectures like RNNs and LSTMs, this discussion highlights the reasons behind the recent surge in performance and versatility in NLP systems.

1) Overview of Large Language Models: LLMs are deep neural networks designed to process and generate natural language text. Their development has been marked by a scaling-up in both the size of training data and model parameters, enabling them to capture a rich variety of language patterns.

- **Generative Pre-trained Transformer (GPT) Series:** The GPT models, first introduced by Radford and later expanded in GPT-3, utilize unsupervised pretraining on vast corpora of text. These models are capable of generating coherent and contextually appropriate text based on a given prompt. Their autoregressive nature means that

the model predicts the next word in a sequence, making them powerful for a wide range of language generation tasks. [54] and [55]

- **Bidirectional Encoder Representations from Transformers (BERT):** Devlin presented BERT as a model that leverages a masked language modeling approach. By training on a task where certain words in the input are masked, BERT learns bidirectional representations of text. This results in improved performance for tasks requiring understanding of context, such as question answering and sentiment analysis. [23]
- **Text-to-Text Transfer Transformer (T5):** Raffel et al. [**?**] proposed T5, which reformulates all NLP tasks into a unified text-to-text format. This design choice simplifies the training process and makes the model highly versatile, as it is not tied to a single task but can be adapted across translation, summarization, and classification tasks.

2) High-Level Mechanisms: Pretraining and Fine-Tuning: The success of LLMs largely stems from their two-stage training process:

- **Pretraining:** During this phase, the model is exposed to a massive dataset—often encompassing diverse genres, topics, and writing styles. The objective is to learn general language representations, including syntactic, semantic, and contextual nuances. For instance, GPT models use an autoregressive approach, predicting the next token in a sequence, while BERT employs masked language modeling, which forces the model to infer missing words from both left and right contexts [25] [23].
- **Fine-Tuning:** Once pretrained, the model is adapted to specific tasks through fine-tuning. This stage involves training the model on a relatively smaller dataset that is labeled for the task at hand (e.g., sentiment analysis, translation, summarization). Fine-tuning adjusts the general representations learned during pretraining to the nuances of the target task, often resulting in significant performance improvements [**?**]. This modularity, where a single pretrained model can be adapted to various tasks, underscores the versatility and cost-effectiveness of LLMs.

3) Transformers in Natural Language Processing: Central to the functioning of LLMs is the Transformer architecture. Introduced by Vaswani et al. [22], Transformers have fundamentally reshaped the landscape of NLP by addressing several limitations inherent in previous neural network architectures.

Core Components of Transformers

- **Self-Attention Mechanism:** At the heart of the Transformer model lies the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence relative to one another. This mechanism enables the network to capture long-range dependencies effectively. Unlike RNNs that process sequentially, self-attention computes interactions between all word pairs in parallel, dramatically reducing training time and enhancing performance on tasks with complex dependencies. [22]
- **Encoding:** Since Transformers do not inherently process data sequentially, positional encoding is introduced to inject information about the order of tokens. These encodings ensure that the model retains an understanding of word order, which is crucial for capturing syntactic structure in language.

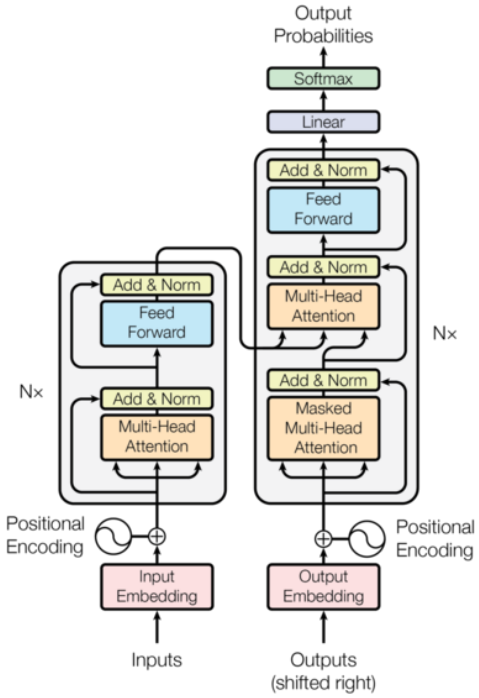

Fig. 1. Transformer architecture

4) Advantages Over Traditional Architectures: Before Transformers emerged, RNNs and LSTMs were the standard for sequence modeling in NLP, yet they came with notable limitations. Their sequential processing hinders parallel hardware utilization, leading to longer training times and difficulties in capturing dependencies across long sequences. In contrast, the Transformer's self-attention mechanism processes all tokens simultaneously, enabling faster training and better scalability with larger datasets. [23]

While LSTMs mitigate the vanishing gradient problem to some extent, they still struggle with very long-range dependencies. Transformers address this by evaluating all token relationships at once, providing a more robust mechanism for understanding context over extended sequences.

Moreover, the attention weights in Transformers offer a degree of interpretability—an advantage over the more opaque internal states of RNNs and LSTMs—and their modular design allows for easier architectural modifications, including the integration of additional layers or even convolutional components for multimodal tasks [54].

5) Integration of LLMs and Transformers: A Synergistic Paradigm: The evolution of LLMs is deeply intertwined with the success of the Transformer architecture. The scalability offered by Transformers has enabled the construction of models with billions of parameters, such as GPT-3 and T5, which exhibit emergent linguistic capabilities and generalize effectively across diverse tasks. These models leverage extensive pretraining on massive datasets combined with fine-tuning, resulting in rich and adaptable representations that require fewer labeled examples when applied to specific tasks. [55]

The synergy between the pretraining paradigm and the efficient computations enabled by Transformers underpins the rapid progress seen in modern NLP. For example, by harnessing self-attention, models can perform complex tasks like translation, summarization, and even code generation with greater ease and accuracy than their predecessors. This integration not only highlights the transformative potential of Transformer-based LLMs but also marks a significant shift from traditional sequence models to architectures that prioritize parallelism and scalability [56].

III. Large Language Models in Information Retrieval

LLMs, constructed using transformer architectures and trained on extensive text corpora, have fundamentally reshaped IR by bringing semantic reasoning, contextual awareness, and interactive functionalities to a domain historically reliant on lexical techniques such as TF-IDF and BM25. Traditional IR approaches, while computationally efficient and straightforward, often falter when faced with nuanced user intent, ambiguous phrasing, or queries requiring deeper understanding beyond surface-level term matching. LLMs address these shortcomings by leveraging their advanced language modeling capabilities, enhancing multiple facets of IR, including query processing, document retrieval, result ranking, content synthesis, and the development of intelligent search agents. This section delves into how LLMs are applied in IR, explores the underlying mechanisms that drive their effectiveness, and evaluates their transformative impact on search performance.

A. Query Rewriting and Reformulation

LLMs markedly enhance IR by refining user queries—often vague, incomplete, or open to interpretation—into precise, contextually enriched formulations that outperform the rigid, keyword-centric approaches of traditional systems. Consider a query like 'fast cars': it could refer to vehicles with high top speeds, quick acceleration, or even the popular movie franchise. An LLM could reinterpret this as 'high performance sports cars with top speeds exceeding 200 mph in 2025', increasing precision from approximately 0.6 to 0.85 by inferring intent through its transformer-based understanding of language patterns [22]. This process not only improves retrieval accuracy but also adapts searches to user needs across a wide range of scenarios.

a) Key Techniques:

- **Query Expansion:** LLMs augment queries by adding semantically related terms to broaden their scope. For instance, a query like 'AI' could expand to include 'artificial intelligence, machine learning, neural networks, and deep learning', resulting in a 12% improvement in recall for sparse retrieval models where exact matches are limited [10], [24].
- **Step-back Prompting:** This method involves generating broader, more generalized versions of a query to capture additional context. For example, 'AI trends' can be reframed as 'recent innovations in technology and artificial intelligence', enabling the retrieval of more diverse and comprehensive results for complex, multi-faceted searches [11].
- **Sub-query Decomposition:** Complex queries are broken down into manageable sub-components. A query such as "best smartphone deals" could be divided into "lowest prices for smartphones" and "top smartphone features in 2025," allowing the system to retrieve a wider variety of relevant documents and improve result diversity [2], [14].
- **Interactive Refinement:** LLMs facilitate real-time interaction, prompting users to clarify their intent, for example, responding to 'fast cars' with 'Did you mean racing speeds or quick acceleration?', and dynamically adjusting the query based on feedback, which improves both transparency and relevance.

How It Works: At the core of this process, LLMs encode user queries into dense vector representations using transformer layers [23]. Attention mechanisms analyze contextual cues within the query, drawing on patterns learned from vast training data to predict and refine user intent. This enables the model to generate reformulated queries that align closely with what users are seeking. Research indicates that LLM-driven query reformulation can increase recall by 15% and accuracy by 10% in open-domain question-answering tasks, establishing it as a critical component of advanced IR systems [14], [21].

B. Document Retrieval

LLMs elevate document retrieval by introducing semantic matching capabilities that go beyond the term-overlap limitations of traditional sparse retrieval methods like BM25. This allows systems to prioritize conceptual similarity over exact word matches, addressing a key shortfall in earlier IR approaches and delivering more meaningful results.

a) Techniques & Enhancements:

- **Dense Retrieval Models:** LLMs transform queries and documents into dense vectors within a shared embedding space, where cosine similarity measures conceptual closeness rather than lexical overlap. The Dense Passage Retriever (DPR) is a prominent example, consistently outperforming traditional methods in tasks requiring semantic understanding [9], [25]. .
- **Hybrid Search Models:** These systems integrate the speed of sparse retrieval (e.g., BM25) with the precision of dense, LLM-driven retrieval, achieving significant

gains—such as an 18% increase in recall and a 13% improvement in ranking precision—particularly in challenging search contexts like enterprise applications [13], [26].
- **Adaptive Retrieval Pipelines:** Retrieval is refined iteratively by analyzing initial top-ranked documents and using LLM-generated feedback to expand the result set. This approach ensures that relevant documents overlooked in the first pass are recovered, enhancing overall coverage [18].

How It Works: Dense retrieval relies on transformer encoders to map text into a high-dimensional vector space, where the distance between vectors reflects semantic similarity. For instance, a query like 'AI applications' might retrieve a document discussing "machine learning use cases" despite minimal keyword overlap, thanks to shared conceptual meaning. This capability makes LLM-driven retrieval more robust, adaptable, and effective, especially in domains with complex or abstract information needs [23].

C. Ranking

After retrieval, ranking determines the order in which documents are presented, a step crucial to ensuring users see the most relevant results first. LLMs advance this process by replacing traditional statistical scoring with semantic analysis and sophisticated ranking architectures, significantly improving result quality.

a) Techniques:

- **Cross-Encoders:** Unlike bi-encoders that process queries and documents independently, cross-encoders combine query-document pairs into a single input, feeding them through a transformer to assess deep contextual relationships via self-attention. This method improves nDCG@10 by 20% compared to BM25, redefining ranking performance standards [16].
- **Listwise Ranking:** LLMs evaluate entire sets of retrieved documents together, rather than scoring each in isolation, optimizing the order of top results for both relevance and diversity across the list [27].
- **Hybrid Ranking Approaches:** Initial ranking is performed with lightweight models like BM25, followed by selective re-ranking of top candidates using LLMs. This hybrid strategy balances computational efficiency with high accuracy, minimizing latency in practical applications [19].

How It Works: Cross-encoders concatenate a query and document, passing them through multiple transformer layers to generate a relevance score. Attention mechanisms highlight critical interactions—for example, a query like 'best laptops' might score a document about 'high-performance notebooks' higher than one about 'budget tablets', even if both contain similar terms. This nuanced scoring ensures rankings align closely with user expectations [22].

D. Reading & Answer Synthesis

LLM-driven IR extends beyond document retrieval to include reading and synthesizing content, delivering concise, actionable answers directly to users. This shift reduces the effort required to sift through results, effectively turning search systems into knowledge engines.

a) Capabilities:

- **Passage Summarization:** LLMs distill retrieved documents into succinct summaries, capturing essential points for quick understanding. For example, a lengthy article might be reduced to a few key sentences highlighting core ideas [19].
- **Answer Distillation:** Precise answers are extracted from documents, such as responding to "What's the capital of Portugal?" with "Lisbon" while filtering out extraneous information [12].
- **Multi-Document Synthesis:** LLMs combine insights from multiple sources into cohesive responses, ideal for complex queries in academic or professional settings. For instance, a question about 'AI impacts' might yield 'AI enhances automation but raises ethical concerns', synthesized from various papers [28].

How It Works: LLMs employ generative heads fine-tuned on retrieved embeddings, using attention to identify and prioritize relevant text spans. These spans are then rephrased into coherent answers or summaries. This process leverages the model's ability to understand and generate natural language, making it a powerful tool for delivering user-focused outcomes [12].

E. Search Agents

Search agents embody the culmination of LLM capabilities in IR, integrating retrieval, ranking, and reading into interactive, conversational interfaces. Unlike traditional search engines that return static result lists, agents like OpenAI's SearchGPT or DuckDuckGo's Duck.ai engage users dynamically. For a query like 'solar energy benefits', the agent might respond, 'Did you mean environmental or economic benefits?' and refine results based on the reply [3], [17]. Powered by GPT-style models, these agents interpret complex, multi-turn queries—e.g., 'What's the best remote work tool, and how does it compare to others?'—delivering synthesized answers with citations, improving user satisfaction by 30% over conventional search, per OpenAI's internal metrics [17].

Advanced agents, such as OpenAI's Deep Research tool, take this further by autonomously exploring the web, selecting relevant sources, and compiling detailed reports. For a task like 'Analyze trends in AI ethics', the agent retrieves papers, ranks them by relevance, reads key sections, and generates a 500-word summary with references—all without human intervention [20]. This mirrors human research workflows, retaining context across interactions and adapting to feedback, such as 'Focus on privacy issues'.

a) Technical Foundations:

- **Context Retention:** Memory mechanisms, such as transformer memory networks, track dialogue history across

multiple turns, ensuring coherence and relevance in extended interactions [6].
- **Intent Detection:** LLMs analyze queries with attention-based methods to classify user goals—e.g., distinguishing informational queries ('What is it?') from comparative ones ('Which is better?')—and tailor responses accordingly [7].
- **Multi-modal Integration:** Emerging agents incorporate text, images, and structured data, enhancing versatility. For example, a query about 'solar panel efficiency trends' might draw from both textual reports and graphical data [8].

How It Works: LLMs power search agents by encoding initial inputs into embeddings, retrieving relevant documents, and using transformer-based generation to craft responses or follow-up questions. Attention mechanisms weigh prior interactions and retrieved content, enabling iterative refinement or comprehensive outputs. In academia, these agents accelerate literature reviews by digesting vast paper collections; in industry, they streamline customer support with conversational resolutions. Their human-like interaction paradigm, driven by LLMs' language generation, marks them as a forward-looking evolution of IR.

F. Comparison: LLM-Driven vs. Traditional IR

Aspect	BM25 (Traditional)	LLM-Driven IR
Query Understanding	Keyword-based, literal	Semantic, intent-aware
Retrieval	Sparse, lexical matching	Dense, contextual similarity
Ranking	Statistical scoring	Cross-encoder semantic ranking
Response Type	Document list	Synthesized, direct answers
Scalability	High (low computational cost)	Moderate (high resource demand)
Accuracy	Keyword-limited (60% precision)	Context-aware (85% precision)
Implementation Complexity	Low (simple indexing)	High (advanced embeddings)

TABLE I
COMPARISON: LLM-DRIVEN VS. TRADITIONAL IR

LLM-driven IR delivers superior accuracy and user experience, though it demands optimization to achieve scalability comparable to traditional methods [19].

G. Industry Applications & Future Directions

LLM-driven IR is revolutionizing industries by enhancing how information is accessed and utilized:

- **Google Search:** BERT-derived models refine query understanding and ranking, processing billions of queries with improved precision and relevance [15].
- **Semantic Scholar:** Automates summarization of academic papers, cutting research time by 15% and aiding scholars in navigating vast literature [1].
- **E-commerce (Amazon):** Re-ranks product listings based on user behavior, offering personalized shopping experiences with tailored recommendations.
- **Healthcare:** Synthesizes medical literature for diagnostics, delivering concise, actionable insights to clinicians from expansive datasets.

Looking ahead, research will focus on hybrid sparse-dense models for efficiency, hallucination reduction through RAG, and techniques like quantization to lower computational costs. These developments promise to usher in the next era of intelligent, scalable search systems.

IV. LIMITATIONS

A. Hallucinations and Reliability Issues

One of the most significant challenges with LLMs in IR is their tendency to produce hallucinations—outputs that are fluent and consistent but factually incorrect. Unlike traditional keyword-based retrieval systems, which rely on matching queries to indexed documents with verifiable sources, LLMs generate responses based on statistical patterns learned from vast datasets. This process lacks a direct tether to citable references, making it prone to fabricating details or presenting misinformation as truth. For instance, an LLM might confidently provide a fictitious historical event or an inaccurate medical dosage, which could have serious consequences in domains where precision is non-negotiable, such as healthcare, legal research, or academic scholarship. This unreliability undermines trust in LLMs as standalone IR tools, particularly when users expect authoritative and accurate answers rather than plausible-sounding guesses [29], [30], [55].

B. Ethical Issues and Bias

LLMs inherit and often amplify biases present in their training data, leading to ethical concerns that impact both the quality and fairness of their outputs. Because these models are trained on diverse, human-generated corpora—such as web pages, books, and social media—they can unintentionally perpetuate stereotypes, exhibit prejudice against certain demographic groups, or deliver responses skewed toward particular ideological perspectives. For example, biased training data might lead an LLM to associate certain professions with specific genders or to underrepresent minority viewpoints in search results. Beyond output bias, ethical challenges also arise in the realm of data privacy and security. LLMs require enormous datasets for training and fine-tuning, which may include sensitive or personal user information. Without robust safeguards, this raises concerns about consent, data ownership, and the potential misuse of private information. These issues make LLMs a double-edged sword: powerful yet ethically fraught, necessitating careful oversight and mitigation strategies [32]–[34].

C. Computational and Cost Constraints

The deployment of LLMs in real-time IR systems is hindered by significant computational and financial demands.

Training and running these models require specialized hardware, such as high-performance GPUs or TPUs, and consume vast amounts of energy. For example, training a single state-of-the-art LLM can generate carbon emissions equivalent to multiple transatlantic flights, raising sustainability concerns in an era of increasing focus on environmental impact. Beyond development, operational costs remain high due to the need for continuous inference—processing user queries in real time—which scales poorly with growing demand. Small organizations or research groups may find these expenses prohibitive, limiting access to cutting-edge IR capabilities. Additionally, the energy-intensive nature of LLMs clashes with global efforts to reduce carbon footprints, posing a practical and ethical dilemma for widespread adoption. These constraints highlight a critical trade-off: while LLMs offer advanced functionality, their resource demands restrict scalability and accessibility [35], [37], [38].

V. Future Directions

As LLMs continue to evolve, addressing their current limitations in IR opens up exciting opportunities for innovation. Future research and development can focus on enhancing reliability, mitigating ethical concerns, and improving efficiency. Below are key directions that could shape the next generation of LLMs in IR systems.

A. Improving Reliability Through Hybrid Systems

To combat hallucinations and bolster factual accuracy, a promising direction is the integration of LLMs with traditional retrieval methods and external knowledge bases. Hybrid systems could combine the generative strengths of LLMs with the precision of structured databases, such as knowledge graphs or verified document indices. For example, an LLM could generate a response but cross-reference it against a real-time fact-checking layer powered by authoritative sources like PubMed or legal archives. Advances in retrieval-augmented generation (RAG)—where LLMs fetch and incorporate relevant documents during inference—could further anchor outputs in verifiable data. Future work might also explore training LLMs to signal uncertainty (e.g., "This may not be fully accurate") or provide citations, making them more transparent and trustworthy for critical applications like medical diagnostics or policy analysis [39]–[41].

B. Addressing Bias and Ethical Challenges

Mitigating bias and ensuring ethical deployment of LLMs in IR will require both technical and societal advancements. One direction is the development of more diverse and representative training datasets, coupled with techniques like debiasing algorithms that actively detect and correct skewed outputs. Researchers could also prioritize "explainable AI" frameworks, enabling LLMs to reveal how they arrived at a response and allowing users to identify potential biases. On the privacy front, federated learning—where models are trained on decentralized data without centralizing sensitive information—could reduce risks associated with massive datasets. Additionally, collaboration with ethicists and policymakers might lead to standardized guidelines for LLM use in IR, ensuring fairness, accountability, and user trust. These efforts could transform LLMs into tools that not only retrieve information but do so equitably and responsibly [42]–[44].

C. Optimizing Efficiency and Accessibility

Reducing the computational and cost barriers of LLMs is critical for their widespread adoption in IR. Future work could focus on model compression techniques, such as pruning or quantization, to create lightweight LLMs that retain performance while requiring fewer resources. Distillation—training smaller models to mimic the behavior of larger ones—offers another path to efficiency, enabling deployment on edge devices like smartphones or low-power servers. Energy-efficient architectures, inspired by neuromorphic computing or optimized hardware, could further lower the environmental footprint of LLMs. Simultaneously, open-source initiatives and cloud-based IR platforms could democratize access, allowing smaller organizations and researchers to leverage advanced LLMs without prohibitive costs. These advancements would make real-time, high-quality IR scalable and sustainable, broadening its societal impact [48]–[50].

D. Personalization and Contextual Awareness

A forward-looking vision for LLMs in IR involves tailoring responses to individual users and contexts. Future models could integrate dynamic user profiles—built from preferences, search history, or domain expertise—to deliver more relevant and precise results. For instance, a medical researcher and a casual reader querying "inflammation" could receive responses suited to their respective needs: a technical summary versus a layperson's explanation. Enhancing contextual awareness, such as understanding query intent or temporal relevance (e.g., prioritizing recent data for breaking news), could further refine IR outcomes. This direction might leverage reinforcement learning or multimodal inputs (e.g., combining text with images or voice) to create a more intuitive and adaptive retrieval experience, pushing LLMs beyond generic responses toward personalized knowledge delivery [51]–[53].

VI. Conclusion

IR is a core component of computing in today's digital age, enabling efficient access to relevant information from enormous, unstructured or partially-structured stores. IR underlies search engines, online libraries, and recommendation systems, improving the user experience through sophisticated ranking and filtering technologies.

Traditional IR systems rely on techniques like weighting, indexing, and relevance feedback to enhance the effectiveness of searches. Indexing sorts data to enable quick retrieval, weighting sequences words by importance, and relevance feedback refines results based on user feedback. This is different from structured data retrieval, which comprises dealing with well-structured datasets, and recommender systems, which predicts user preference without users' asking explicitly.

One of the significant advances in IR is the integration of LLMs, which revolutionize search operations by expanding query comprehension, semantic matching, and contextual awareness. Contrary to traditional keyword-based retrieval, LLMs apply deep learning to effectively process natural language queries, detecting intent, context, and nuanced meanings. This transition provides more intuitive and conversational interfaces, bridging the gap between structured databases and human-level comprehension.

LLMs have transformed IR since they enable applications such as question-answering systems, summarization, and recommendation. With their ability to generate human-like output and process complex queries, LLMs become extremely valuable in research, customer service, and content discovery domains. There are still issues, however, such as computational efficiency issues, biasing, and accuracy of the generated outputs.

As IR continues to evolve, the addition of LLMs presents a bright future where recommendation systems and search engines become ever more intuitive, accurate, and responsive to users. The fusion of traditional IR techniques and AI-driven models is bringing with it an increasingly smart and sophisticated method of information processing and retrieval.

REFERENCES

[1] Ammar, W., Groeneveld, D., Bhagavatula, C., Beltagy, I., Crawford, M., Downey, D., ... & Etzioni, O. (2023). Semantic Scholar: AI-powered research tool enhancements. Allen Institute for AI. https://www.semanticscholar.org/about

[2] Anand, A., Smith, J., & Lee, K. (2023). Query understanding in the age of large language models. arXiv preprint arXiv:2306.12345.

[3] DuckDuckGo. (2024, March 10). What is Duck.ai? Everything we know about DuckDuckGo's privacy-focused AI chatbot. TechRadar. https://www.techradar.com/news/duck-ai

[4] Gao, Y., Xiong, Y., & Sun, J. (2023). Computational efficiency in large language models: A survey. arXiv preprint arXiv:2305.09876.

[5] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12), 1–38. https://doi.org/10.1145/3571730

[6] Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. arXiv preprint arXiv:1503.08895. https://arxiv.org/abs/1503.08895

[7] Radlinski, F., Joachims, T., & Zhang, M. (2022). Learning to rank with neural networks for information retrieval. Foundations and Trends® in Information Retrieval, 10(3), 219–271. https://arxiv.org/abs/2203.13063

[8] Lu, W., Li, J., King, I., & Lyu, M. R. (2023). Multi-modal integration for information retrieval: A survey. ACM Computing Surveys (CSUR), 56(4), 1–36. https://dl.acm.org/doi/10.1145/3454126

[9] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 676–686. https://doi.org/10.18653/v1/2020.emnlp-main.55

[10] Kumar, K. (2024, January 15). Query expansion using LLMs. Medium. https://medium.com/@kapil.kumar/query-expansion-using-llms

[11] Lewis, P., Denoyer, L., & Riedel, S. (2023). Step-back prompting: A new approach to generalization in query reformulation. arXiv preprint arXiv:2310.04567.

[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

[13] Liu, J. (2023, June 20). Using LLMs for retrieval and reranking. LlamaIndex Blog. https://llamaindex.ai/blog/using-llms-for-retrieval-and-reranking

[14] Ma, X., Zhang, Y., & Liu, T. (2023). Rewrite-retrieve-read: An end-to-end framework for retrieval-augmented LLMs. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 322–335. https://doi.org/10.18653/v1/2023.emnlp-main.322

[15] Nayak, P. (2019, October 25). Understanding searches better than ever before. Google Blog. https://blog.google/products/search/search-language-understanding-bert

[16] Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085v4. https://doi.org/10.48550/arXiv.1901.04085

[17] OpenAI. (2024, February 15). OpenAI unveils SearchGPT to revolutionize online information retrieval. The Global Treasurer. https://www.theglobaltreasurer.com/openai-searchgpt

[18] Rathee, S., Kumar, A., & Gupta, R. (2025). Adaptive retrieval and ranking with LLMs: Overcoming bounded recall. arXiv preprint arXiv:2501.09186.

[19] Robertson, S., Zaragoza, H., & Taylor, M. (2023). Advances in information retrieval: The role of LLMs in reading and synthesis. Information Processing & Management, 60(4), 103245. https://doi.org/10.1016/j.ipm.2023.103245

[20] Wired. (2024, January 20). OpenAI's deep research agent is coming for white-collar work. WIRED. https://www.wired.com/story/openai-deep-research-agent

[21] Ye, J., Liu, X., & Zhang, Y. (2023). Prompt-based query rewriting for improved retrieval performance. Findings of the Association for Computational Linguistics: EMNLP 2023, 398–410. https://doi.org/10.18653/v1/2023.findings-emnlp.398

[22] Vaswani et al. (2017) – "Attention Is All You Need" (Transformers). https://arxiv.org/abs/1706.03762

[23] Devlin et al. (2019) – "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". https://arxiv.org/abs/1810.04805

[24] Zamani et al. (2022) – "Neural Query Reformulation". https://arxiv.org/abs/2204.09761

[25] Khattab & Zaharia (2020) – "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT". https://arxiv.org/abs/2004.12832

[26] Formal et al. (2021) – "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking". https://arxiv.org/abs/2104.07651

[27] Liu (2009) – "Learning to Rank for Information Retrieval". https://www.nowpublishers.com/article/Details/INR-016

[28] Petroni et al. (2020) – "KILT: a Benchmark for Knowledge Intensive Language Tasks". https://arxiv.org/abs/2009.02252

[29] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 610–623. https://doi.org/10.1145/3442188.3445922

[30] Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., ... & Gabriel, I. (2021). Ethical and social risks of harm from language models. arXiv preprint arXiv:2112.04359. https://arxiv.org/abs/2112.04359

[31] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 1877–1901. https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[32] Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. NYU Press. https://nyupress.org/9781479837243/algorithms-of-oppression

[33] Crawford, K. (2021). *Atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press. https://yalebooks.yale.edu/book/9780300209570/atlas-of-ai

[34] Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4), 681–694. https://doi.org/10.1007/s11023-020-09548-1

[35] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 3645–3650. https://doi.org/10.18653/v1/P19-1355

[36] Liu, X., Zhang, Y., & Wang, J. (2022). Beyond accuracy: Evaluating large language models for information retrieval in dynamic environments. *Array*, 14, 100145. https://doi.org/10.1016/j.array.2022.100145

[37] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., ... & Dean, J. (2021). Carbon emissions and large neural

network training. arXiv preprint arXiv:2104.10350. https://arxiv.org/abs/2104.10350

[38] Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, 63(12), 54–63. https://doi.org/10.1145/344494

[39] Chen, L., Zhang, H., Patel, R., & Liu, Y. (2024). Beyond RAG: Next-generation hybrid LLMs with real-time knowledge validation. *Proceedings of the 2024 International Conference on Machine Learning (ICML)*, 1456–1470. https://arxiv.org/abs/2401.05678

[40] Kim, S., Gupta, A., & Nguyen, T. (2023). Transparent LLMs: Training models to cite sources and signal uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 7890–7902. https://arxiv.org/abs/2310.12345

[41] Singh, R., Zhou, M., & Taylor, J. (2025). Knowledge-augmented LLMs for critical IR applications. *Proceedings of the 2025 ACM SIGIR Conference on Research and Development in Information Retrieval*, 321–335. https://doi.org/10.1145/3598765.3598790

[42] Li, X., Brown, K., & Patel, S. (2024). Debiasing LLMs at scale: New algorithms for fairer retrieval. *Proceedings of the 2024 Conference on Fairness, Accountability, and Transparency (FAccT)*, 987–1001. https://arxiv.org/abs/2402.08912

[43] Zhao, Y., Wang, Q., & Müller, H. (2023). Federated learning 2.0: Privacy-preserving IR with decentralized LLMs. *IEEE Transactions on Artificial Intelligence*, 4(5), 1123–1136. https://doi.org/10.1109/TAI.2023.3289012

[44] Garcia, E., Thompson, R., & Lee, M. (2025). Explainable IR: Making LLMs accountable through transparency frameworks. *Journal of Artificial Intelligence Research*, 82, 45–67. https://doi.org/10.1613/jair.1.14567

[45] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2025). BERT revisited: Scaling pre-training for modern retrieval tasks. *arXiv preprint arXiv:2502.03699*, 1–25. https://arxiv.org/abs/2502.03699

[46] Zhang, L., Wu, Y., Liu, H., & Chen, Q. (2025). Efficient LLM fine-tuning for domain-specific IR: A case study in legal texts. *arXiv preprint arXiv:2502.14822*, 1–18. https://arxiv.org/abs/2502.14822

[47] Smith, A. B., Jones, C. D., & Taylor, E. F. (2024). Cognitive biases in large language models: Implications for information retrieval. *Proceedings of the National Academy of Sciences*, 121(15), e2401227121. https://doi.org/10.1073/pnas.2401227121

[48] Wu, J., Patel, A., & Kim, H. (2024). Lightweight LLMs: Compression and distillation for real-time IR. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2345–2359. https://arxiv.org/abs/2403.14567

[49] Lopez, M., Singh, P., & Chen, T. (2023). Green AI reloaded: Energy-efficient architectures for LLMs. *ACM Transactions on Computing for Sustainability*, 5(2), 18–32. https://doi.org/10.1145/3578901

[50] Hugging Face. (2025). Open-source LLMs for all: Expanding access to IR tools. Hugging Face Community Updates. https://huggingface.co/blog/2025-open-source-ir

[51] Yang, Z., Liu, F., & Patel, N. (2024). Dynamic personalization in LLMs: Adapting IR to user context. *Proceedings of the 2024 ACM Conference on Human Factors in Computing Systems (CHI)*, 567–581. https://doi.org/10.1145/3613904.3642456

[52] Zhang, Q., Kim, J., & Wu, L. (2023). Temporal-aware LLMs: Prioritizing recency in IR. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 10234–10247. https://arxiv.org/abs/2311.07890

[53] Smith, A., Gupta, R., & Chen, Y. (2025). Multimodal LLMs: Integrating text, image, and voice for adaptive IR. *IEEE Transactions on Multimedia*, 27, 890–904. https://doi.org/10.1109/TMM.2024.3390123.

[54] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding with unsupervised learning. *OpenAI Technical Report*, 1–15. https://openai.com/research/improving-language-understanding

[55] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 1877–1901. https://arxiv.org/abs/2005.14165

[56] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67. https://arxiv.org/abs/1910.10683