International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.36, September 2025

A Survey of Query Refinement Techniques: From Neural Architectures to Practical Applications

Zahra Taheri
School of Computer Science, University of Windsor
Windsor, ON, Canada

Mahdis Saeedi
School of Computer Science, University of Windsor
Windsor, ON, Canada

Ziad Kobti
School of Computer Science, University of Windsor
Windsor, ON, Canada

Hossein Fani
School of Computer Science, University of Windsor
Windsor, ON, Canada

ABSTRACT

Query refinement plays a central role in modern information retrieval (IR) systems by improving query clarity, resolving ambiguity, and enhancing result relevance. This survey provides a comprehensive overview of the model architectures and application domains associated with query refinement techniques. The paper first examines classical non-neural models and then explores a range of neural architectures, including embedding-based methods, recurrent neural networks (RNNs), sequence-to-sequence (seq2seq) frameworks, and transformer-based models. Special attention is given to the progression from static representations to context-aware and generative approaches, with an emphasis on how these models capture user intent and session context. The study then reviews the deployment of query refinement methods across practical domains such as product search, music retrieval, job search, and personalized information access. These applications demonstrate the real-world impact of query refinement in handling ambiguous queries, adapting to user preferences, and improving overall retrieval performance. By highlighting key advancements and challenges, this survey offers insight into the current state and future direction of query refinement research.

Keywords

Query Reformulation, Query Suggestion, Information retrieval, Web Search

1. INTRODUCTION

In modern information retrieval (IR) systems, users often express their information needs through short and ambiguous queries that lack sufficient context or specificity. As a result, these systems frequently struggle to interpret intent accurately and return relevant results. Query refinement—a process that involves reformulating, expanding, or clarifying user queries—has emerged as a fundamental solution to this problem. By narrowing the gap between user intent and query expression, refinement techniques enable IR systems to deliver more precise and contextually appropriate results, thereby enhancing user satisfaction and retrieval effectiveness. The field of query refinement has evolved considerably over the past decades. Early efforts were dominated by symbolic and statistical models, including relevance feedback, pseudo-relevance feedback, and thesaurus-based expansion. While these methods offered foundational solutions for improving query quality, they were often limited by their reliance on lexical similarity and handcrafted features. The advent of machine learning, particularly deep learning, introduced new opportunities for modeling semantic relationships, user behavior, and contextual dependencies in query formulation.

This paper presents a comprehensive survey of query refinement with a specific focus on neural model architectures and their applications in real-world retrieval systems. The authors categorize the architectural landscape into four major classes: embedding-based models, recurrent neural networks (RNNs), sequence-to-sequence (seq2seq) models, and transformer-based architectures. These models vary in their ability to represent context, adapt to user intent, and generate coherent reformulations, and the paper analyzes their mechanisms and comparative advantages in detail. In addition to model architectures, this survey explores how these techniques have been applied across diverse retrieval environments, including product search, music discovery, job search, and personalized information access. These applications demonstrate the growing need for query refinement in domains where search intent is dynamic, user-specific, and sensitive to subtle contextual cues.

By synthesizing developments in both model design and applied settings, this survey aims to provide a clear view of the current capabilities and limitations of query refinement systems. The paper also highlights open challenges and future directions for research in user intent modeling, personalization, and scalable deployment of neural refinement techniques.

2. MODEL ARCHITECTURE

Query refinement plays a crucial role in assisting users during information-seeking activities by enabling more accurate and relevant retrieval outcomes. Extensive research has focused on developing techniques to make user-submitted queries more effective and comprehensible [12]. These techniques—encompassing query refinement, reformulation, and expansion—seek to bridge the gap between a user's initial query and their underlying information need. In recent years, the introduction of deep learning into information retrieval has further advanced this objective by enabling a more nuanced understanding of user behavior and intent [11]. Leveraging large-scale datasets and sophisticated neural architectures, these models can capture complex patterns in query formulation and search interactions. Approaches based on

word embeddings, sequence-to-sequence learning, and transformers have shown substantial improvements in both the precision and relevance of search results. This section provides an overview of traditional, non-neural query refinement models in Section 2.1, followed by neural models in Section 2.2.

2.1 Non-Neural Models

With the emergence of search engines in the 1990s and the rise of social media in the early 2000s, the volume of digital content available on the web grew rapidly—a trend that continues to accelerate. This explosion of information has driven the development of more sophisticated information retrieval (IR) techniques to help users find relevant content efficiently. Even prior to the advent of modern search engines, early work on query refinement began in the 1960s. Notably, Maron and Kuhns [38] introduced the concept of probabilistic indexing to improve literature retrieval in mechanized library systems. Their method assigned a relevance number—a statistical estimate of relevance—to each document, ranking results by their likelihood of satisfying the user's query. In parallel, Boolean retrieval models allowed users to manually refine queries using logical operators such as AND, OR, and NOT, laying the groundwork for structured query construction. These foundational techniques provided the basis for more advanced algorithms developed in subsequent decades. The concept of relevance feedback—introduced in 1971 [19]—marked a major milestone. It enabled systems to incorporate user judgments about previously retrieved documents to iteratively improve search performance. Rocchio's formulation of relevance feedback within the vector space model became a cornerstone for later research, representing both documents and queries as vectors in a multidimensional space [47]. Several families of non-neural refinement models evolved from this foundation: **Collection-based term co-occurrence:** methods leverage the frequent co-appearance of terms in documents to identify semantically related expansion terms [29, 51]. This helps recover relevant documents that might not contain the exact query terms but share a topical association. **Cluster-based retrieval:** organizes documents into clusters based on content similarity. Rather than matching individual documents to the query, the retrieval model ranks and returns the most relevant clusters [42, 27]. This helps contextualize results and reduce redundancy. **Comparative term distribution analysis:** compares how terms are distributed across relevant and non-relevant documents. This strategy reveals patterns useful for reweighting or augmenting query terms [Porter; Salton].
The 1970s and 1980s saw the development of the vector space model [Salton] and the introduction of TF-IDF weighting [Jones], which improved document-term relevance estimation by penalizing common terms and emphasizing rare, informative ones. These classical models were well-suited to the era's relatively short queries and smaller corpora, especially in academic settings. Despite modern advances, users still tend to issue short, under-specified queries, leading to ambiguity and vocabulary mismatch—the disconnect between user terminology and relevant document terms. This motivates continued use of query refinement strategies, including: **Pseudo-relevance feedback (PRF)** methods assume that the top-ranked documents from an initial query are relevant and extract expansion terms from them [17]. For instance, Lv and Zhai [37] proposed a positional relevance model that assigns higher weights to terms appearing near query terms. Karisani et al. [30] introduced a weighted feedback approach that prioritizes terms based on document relevance scores. Keikha et al. [31] further expanded this paradigm by linking queries to Wikipedia articles, extracting relevant terms that reflect the core meaning of the query. **Thesaurus-based expansion** leverages external lexical resources—such as WordNet—to expand queries using semantically related terms, including synonyms, hypernyms, and hyponyms [41]. For example, a query for "car" may be expanded with "vehicle," "automobile," or "sedan." This semantic broadening helps match conceptually similar content. WordNet's glosses and synsets are especially useful in capturing nuanced meaning [35, 57], and domain-specific thesauri can further enhance relevance in technical contexts. Azad et al. [5] proposed the **Wikipedia-WordNet-based Query Expansion (WWQE)** method, which combines WordNet's semantic structure with Wikipedia's rich context. Their two-level synset extraction process increases expansion depth, and a novel in-link score ranks terms based on their prominence in Wikipedia. This hybrid approach balances semantic richness with real-world relevance. **Term co-occurrence analysis** is another widely used strategy. It assumes that frequently co-occurring terms are likely to be semantically related. Bai et al. [8] and Carpineto et al. [16] utilized top-ranked documents for co-occurrence-based expansion. Bhatia et al. [10] took this further by developing a probabilistic model that derives suggestions based solely on intra-corpus co-occurrence, avoiding reliance on external logs. Anand and Kotov et al. [4] extended co-occurrence models using external knowledge sources such as DBpedia and ConceptNet. Their graph-based approach calculates association strength using pointwise mutual information, enhancing semantic precision. Coa et al. [13] applied machine learning classifiers to filter out irrelevant expansion terms, while Xu et al. proposed a ranking scheme based on the strength of term co-occurrence within top documents.
These non-neural methods remain effective in scenarios where interpretability and resource-efficiency are priorities. However, they suffer from several limitations: Many rely solely on statistical co-occurrence, which may overlook semantic meaning or user context. 2-They may struggle with data sparsity, especially in specialized domains or low-resource settings.3- These models often fail to adapt to dynamic or evolving information needs, making them less responsive to modern interactive search behaviors. As a result, they frequently produce expansions that are too general, too narrow, or irrelevant due to lack of contextual understanding. To overcome these challenges, researchers have increasingly turned to neural network models. These offer deeper semantic representation capabilities, are more adaptable to varied data types, and learn directly from user interactions. The following section (2.2) explores neural query refinement methods, detailing their architectures and contributions to addressing the limitations of traditional models.

2.2 Neural Models

Neural network models have become increasingly prominent in query refinement due to their ability to capture complex semantic relationships and contextual dependencies. Unlike traditional statistical approaches, which often rely on surface-level term co-occurrence or manually engineered features, neural models learn hierarchical patterns from large-scale datasets, enabling more accurate and flexible refinement strategies. These models excel at identifying latent relationships between users and queries, modeling subtle interactions that are critical for interpreting ambiguous or underspecified queries. For instance, embedding-based models such as Word2Vec and GloVe produce dense vector representations of words that reflect semantic similarity, allowing for more meaningful query expansion. Beyond static embeddings, recurrent neural networks (RNNs) have been applied to model the temporal structure and sequential flow of queries in a session, capturing user intent over time. Building on RNNs, sequence-to-sequence (seq2seq) models have been employed to generate refined versions of queries by treating the problem as a translation task—from an initial query to an improved one. More recently, transformer-based architectures, including BERT and GPT, have advanced the state of the art through attention mechanisms that model word dependencies across the entire input. These models enable deep contextual understanding and exhibit strong performance in both query reformulation and expansion. In the following subsections, we examine

four major categories of neural models used in query refinement: embedding-based models, RNNs, sequence-to-sequence architectures, and transformers. Each category is discussed in terms of its underlying mechanisms, representative methods, and their contributions to improving search effectiveness.

2.2.1 Embedding Models. Embedding-based models learn distributed representations of words in a continuous vector space, where the distance and orientation between vectors reflect linguistic relationships such as semantic similarity or syntactic roles [49]. These dense vector representations enable more nuanced comparisons between terms than traditional bag-of-words models and have proven effective in various NLP tasks, including query expansion, document retrieval, and sentiment analysis. By embedding words into a shared space, these models allow for more contextually aware query refinement, improving retrieval relevance and reducing vocabulary mismatch.

Mikolov et al. [40] introduced Word2Vec, a seminal approach that revolutionized the use of word embeddings in natural language processing. Word2Vec demonstrated that the geometric properties of word vectors could capture analogical relationships (e.g., $king–man + woman \approx queen$). The model offers two architectures: Continuous Bag of Words (CBOW), which predicts a word based on its context, and Skip-gram, which predicts surrounding words from a target word. These models, trained on large corpora, laid the foundation for embedding-based query refinement. Building on Word2Vec, Roy et al. [48] proposed a framework for query expansion that uses the K-nearest neighbors of a query vector to identify semantically related terms. This embedding-driven method showed substantial improvements over traditional expansion techniques, especially in addressing vocabulary mismatch. Similarly, Mitra et al. [43] utilized vector representations of queries and reformulations within a session to explore user intent transitions. Their method, based on convolutional neural networks (CNNs), analyzed differences between embedding vectors to model semantic relationships between reformulated queries. However, it did not integrate broader session data such as click-through behavior, potentially limiting its contextual depth. To enhance context-aware query suggestion, Jiang et al. [28] introduced the Reformulation Inference Network (RIN), which learns homomorphic embeddings that retain both syntactic and semantic properties across sessions. By modeling the relationships between queries and their reformulations, RIN improves the quality of suggestions and captures evolving user intent. Along similar lines, Ahmad et al. [2] proposed a model that represents both queries and clicked documents using context-aware neural embeddings, incorporating attention mechanisms to better align query and document representations within a session. This alignment enhances both ranking and reformulation quality by focusing on session-specific relevance signals. The use of embeddings is also prominent in large-scale recommendation systems. Okura et al. [45] developed a news recommendation system using embeddings for articles and user behaviors. Trained on massive datasets, the model achieved notable improvements in scalability and accuracy by aligning articles with user preferences through Word2Vec and GloVe embeddings. However, these static embeddings are limited by their inability to reflect word meaning variation across contexts.

To address this, contextual embeddings such as those produced by BERT and GPT have emerged. These models provide dynamic, context-sensitive representations that adjust word meanings based on surrounding text. Pre-trained on large-scale corpora, models like BERT offer deep semantic understanding and are more effective at modeling complex dependencies and long-range context [46]. Embedding models have also been applied in personalized search. Zhou et al. [60] introduced a cross-lingual query refinement model that enhances user profiles by combining word embeddings with topic models. Applied to pseudo-relevant documents, this hybrid approach improves relevance by incorporating both term-level and thematic context. Yao et al. [56] extended this direction by developing personalized word embeddings for each user, capturing individual search behavior and intent. Their multi-task learning framework trains ranking and reformulation modules simultaneously, improving both personalization and retrieval effectiveness.

Embedding-based models provide a foundational layer for neural query refinement by enabling efficient, semantically rich comparisons between terms, queries, and documents. Their ability to represent lexical semantics in a continuous space allows for flexible and effective query expansion and reformulation. Static embeddings such as Word2Vec and GloVe have shown strong performance in various tasks; however, they struggle with polysemy and contextual ambiguity. Contextual embeddings, such as those from BERT, address these limitations by adapting representations to surrounding text, offering improved disambiguation and semantic fidelity. Despite their advantages, embedding-based models often require large training corpora and may be less interpretable than symbolic methods. Furthermore, without additional architectural support (e.g., attention or recurrence), they may fail to fully capture user intent evolution across sessions. Overall, embeddings are essential to modern refinement pipelines and serve as the building blocks for more complex neural architectures discussed in the following subsections.

2.2.2 Recurrent Neural Networks Model. Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining a dynamic internal state that captures information from previous inputs. This memory-like capability makes RNNs particularly suitable for natural language processing tasks, including query reformulation, where understanding the order and dependencies between words is essential. In the context of query refinement, an RNN can encode an input sequence—such as an initial user query—and generate a refined version that is more contextually appropriate. This allows the system to track evolving user intent and produce more relevant query suggestions. However, traditional RNNs often struggle with modeling long-range dependencies due to the vanishing or exploding gradient problem, which can impede performance in complex or lengthy sequences.

To address these limitations, several works have built upon RNNs using architectural innovations such as hierarchical encoders and attention mechanisms. For example, Chen et al. [19] proposed a hierarchical recurrent neural query suggestion model that captures both short-term and long-term user interests. The model comprises a session-level encoder, which represents the user's current session context, and a user-level encoder, which integrates information from previous sessions to capture long-term preferences. An attention mechanism dynamically weights the relevance of past queries, allowing the model to prioritize queries that are more pertinent to the current context. Experiments demonstrated significant improvements in Mean Reciprocal Rank (MRR) and Recall, particularly in short sessions and for users with sparse histories, validating the model's effectiveness in generating personalized, context-aware query suggestions. Beyond session modeling, RNNs have also been applied in interactive query refinement. Erbacher et al. [23] introduced a framework that employs a user simulation model to iteratively improve query suggestions through clarification dialogues. The system uses a hierarchical RNN to encode both individual interactions and sequences of interactions, simulating user feedback during refinement. Reinforcement learning is used to optimize the system based on simulated user behavior, resulting in a model that dynamically adapts its suggestions. Empirical results show notable gains in MRR and Precision@k, emphasizing the benefit of simulated user feedback in enhancing refinement accuracy. The Reformulation Inference Network (RIN) by Jiang et al. [28] applies an RNN-based architecture to model reformulation across sessions. RIN uses homomorphic embeddings to capture both syntactic and semantic properties of queries, which are then used in an inference

network to generate improved formulations. This method allows for the representation of nuanced query transitions within sessions and was shown to outperform baseline models in query suggestion accuracy on real-world datasets. In a similar vein, Ahmad et al. [2] proposed a context-aware retrieval model that integrates sequential dependencies among queries and click data. Unlike prior multi-task models that isolate document matching or reformulation, this approach unifies these components using RNNs with attention mechanisms. The model dynamically attends to previous queries and interactions to reflect user intent more accurately, leading to improved query suggestions and document ranking.

To better handle long sequences, Gated Recurrent Units (GRUs) were introduced as a refinement of standard RNNs. GRUs incorporate gating mechanisms—specifically the update and reset gates—to manage the flow of information. These gates enable the model to preserve or discard information selectively, mitigating the vanishing gradient issue and improving learning across long sequences. Several recent models in query refinement and click prediction have leveraged GRUs to improve performance and stability. For example, Chen et al. [18] developed a context-aware click model that uses GRUs to model session-level query and click sequences. The architecture includes a relevance estimator and an examination predictor. The former encodes session context using pre-trained embeddings and GRUs, while the latter estimates the probability of click-through events. Attention mechanisms are employed to highlight the most relevant past interactions. Experiments showed significant performance gains in click prediction accuracy compared to traditional models. Additionally, Li et al. [33] proposed a query suggestion model that combines GRUs with adversarial learning. The GRU component encodes user query and click behavior, while adversarial examples generated by a classifier improve the model's robustness. The two components are jointly trained, enhancing the model's ability to handle noisy or challenging inputs. Results demonstrated consistent improvements across various context lengths, highlighting the benefit of incorporating adversarial learning into GRU-based architectures.

RNNs and GRUs offer powerful mechanisms for modeling sequential user behavior in query refinement. RNNs are capable of capturing the temporal structure of search sessions and are particularly effective in tasks where query context evolves over time. However, their performance degrades with long sequences due to gradient instability. GRUs address these issues by regulating information flow through gating mechanisms, leading to better retention of long-term dependencies. While GRUs improve on traditional RNNs in terms of memory and learning efficiency, they also introduce added complexity in model tuning and training. Despite these trade-offs, both RNNs and GRUs have demonstrated strong performance in refining queries, modeling user intent, and generating personalized suggestions—especially when combined with attention mechanisms or reinforcement learning. These models have laid the groundwork for more advanced architectures, such as transformers, which further enhance contextual understanding and scalability in modern query refinement systems.

2.2.3 Sequence-to-Sequence Models. Sequence-to-sequence (seq2seq) models represent a significant advancement over traditional RNN and GRU architectures for tasks involving the transformation of one sequence into another. These models adopt an encoder-decoder structure, where the encoder processes the input sequence and encodes it into a fixed-length context vector, which is then used by the decoder to generate an output sequence. This design supports flexible handling of variable-length inputs and outputs, making seq2seq models particularly well-suited for tasks such as machine translation, text summarization, and query reformulation.

An important innovation in seq2seq models is the integration of attention mechanisms, which allow the decoder to dynamically attend to specific elements of the input sequence during decoding. This mitigates the limitations of fixed-length context vectors and improves the model's ability to handle long-range dependencies and complex query structures. As a result, seq2seq models have been increasingly adopted in query suggestion and reformulation, where capturing nuanced context is critical for generating relevant outputs. Sordoni et al. [50] introduced a hierarchical seq2seq model that improves query suggestion by encoding the sequence of previous queries in a session. The model uses a two-level architecture: a session-level RNN that generates a context vector summarizing past queries, and a query-level decoder that produces the next query. Attention mechanisms further enhance this model by allowing it to focus on relevant parts of the session history. This hierarchical structure addresses the shortcomings of traditional seq2seq models, which typically neglect broader session context, and enables the generation of queries that better reflect evolving user intent. A common challenge in query reformulation is the inability of seq2seq models to handle out-of-vocabulary (OOV) or low-frequency terms. To address this, Dehghani et al. [20] proposed a seq2seq framework that incorporates both a query-aware attention mechanism and a copy mechanism. The attention mechanism weighs the importance of individual queries in the session, while the copy mechanism enables the decoder to reuse specific terms from earlier queries—even if they are rare or unseen in training data. The model performs well as both a discriminative ranker and a generative model, demonstrating its robustness in handling complex vocabulary and session dependencies.

To further capture long- and short-term dependencies, Wu et al. [55] introduced the Feedback Memory Network (FMN), which integrates session context and user feedback via an external memory component. The model stores previously issued queries and click-through data, using an attention mechanism to retrieve relevant past interactions when generating new suggestions. FMN shows strong empirical performance, outperforming prior methods in both precision and recall, particularly in commercial search settings. This architecture demonstrates the benefit of maintaining an explicit memory of user behavior for adaptive and accurate query suggestion. Zhong et al. [59] proposed the Personalized Query Suggestion (PQS) model, which builds on the seq2seq framework with stacked LSTM layers in both encoder and decoder components. To personalize query suggestions, the model incorporates an attention mechanism and a user-specific embedding layer trained on historical search data. This enables the system to generate recommendations that align closely with individual users' preferences and behavior. Experiments confirm that PQS substantially improves both relevance and user satisfaction over baseline methods, highlighting the value of user-aware refinement strategies. In exploratory search scenarios, users may lack domain expertise to evaluate the usefulness of query suggestions. Medlar et al. [39] addressed this by generating alternative queries that yield results similar to those currently retrieved. Their model employs a seq2seq architecture that combines summarization techniques with query reformulation. The encoder captures session context, while the decoder produces concise, contextually relevant alternative queries. Results show that this approach improves user satisfaction and relevance, especially in scientific literature search tasks. Additionally, seq2seq models have been applied in query auto-completion. Wang et al. [54] proposed an LSTM-based encoder-decoder model for predicting query completions in real time. The model incorporates beam search to generate multiple completion candidates and uses sequential patterns within the input to produce fluent, context-aware completions. Their approach demonstrates superior performance in both speed and accuracy compared to traditional auto-completion systems.

Sequence-to-sequence models offer a flexible and powerful framework for query reformulation, suggestion, and completion by enabling contextualized generation of variable-length sequences. Their encoder-decoder structure and attention mechanisms allow

for nuanced modeling of session context and user behavior, outperforming traditional RNNs in many query refinement tasks. However, these models come with limitations. They require large volumes of training data and computational resources, particularly when extended with deep LSTM stacks or attention layers. Moreover, performance can degrade if the model is poorly trained or if training data is domain-mismatched, leading to outputs that are syntactically fluent but semantically irrelevant. The integration of copy mechanisms and personalized embeddings partially addresses these challenges, enhancing robustness in the presence of rare terms or diverse user preferences. Despite their complexity, seq2seq models continue to serve as a core architecture in many modern query refinement systems, bridging earlier sequential models and the more recent transformer-based approaches discussed in the following section.

2.2.4 Transformer-Based Models. Transformer-based models, first introduced by Vaswani et al. [52], have significantly advanced the field of sequence modeling by addressing the limitations of recurrent architectures such as RNNs and seq2seq models. Unlike these earlier models, which process input sequences sequentially and are prone to inefficiencies in modeling long-range dependencies, transformers rely entirely on self-attention mechanisms to model relationships between all positions in a sequence simultaneously. This parallelism not only enhances training efficiency but also improves the model's ability to capture global contextual information, making transformers particularly effective in tasks such as query refinement and expansion. The transformer architecture uses multi-head attention to capture diverse relationships within input sequences and employs positional encodings to preserve word order. These design features allow transformers to process long and complex sequences with greater precision and efficiency. Several pre-trained transformer models—such as BERT [22], BART [32], and T5—have been fine-tuned for downstream tasks including query reformulation, suggestion, and expansion. Their contextual understanding and generalization capabilities have led to widespread adoption in both academic and industrial IR systems. Building on this foundation, Muster et al. [44] explored the integration of BERT and BART into query suggestion systems. BERT is used to understand user query semantics through bidirectional encoding, while BART, as a sequence-to-sequence model, generates candidate reformulations. The combined framework benefits from BERT's deep contextual representations and BART's generative capabilities, resulting in personalized and context-aware suggestions. Their results demonstrated that transformer-based methods outperform traditional RNN-based systems in generating relevant and diverse suggestions.

Despite the success of transformer models in encoding query semantics, many traditional implementations produce a single representation per query, which may be insufficient for handling ambiguous or underspecified queries. To address this limitation, Hashemi et al. [25] proposed a framework that generates multiple representations per query to capture different user intents or subtopics. Using BART to initialize the encoder-decoder architecture, the model also incorporates a guided transformer mechanism that integrates external knowledge to enhance diversity in representation. This approach provides more nuanced and flexible interpretations of a user's information need, improving both precision and relevance in retrieval. Context-aware transformer architectures have also been proposed to personalize query suggestions. Zhou et al. [62] introduced a model that combines RNNs, attention mechanisms, and transformer components to capture and encode user search history. The hierarchical encoder processes interaction data at multiple levels, and the attention mechanism adjusts its focus based on the current query context. In a follow-up work, Zhou et al. [63] (PSSL) extended this direction by applying self-supervised learning and contrastive sampling to train a personalized retrieval model. This method learns high-quality representations from unlabeled user interactions, improving scalability and performance across diverse search contexts.

A growing body of work also applies transformer-based models to query expansion. Zheng et al. [58] introduced BERT-QE, a method that selects semantically relevant text chunks from documents to expand queries. The model encodes both the query and candidate chunks using BERT, ranks their relevance via cosine similarity, and then constructs expanded queries that better reflect user intent. This chunk-based expansion technique has shown improvements in text retrieval effectiveness by providing richer context for ambiguous queries. Transformer models have also been used for query reformulation in community-based platforms. Cao et al. [14] developed a BERT-based reformulation pipeline for Stack Overflow, using historical query logs to retrieve and rank reformulation candidates. Their model identifies semantically related queries and rewrites the original query to better align with typical user phrasing in the domain. Similarly, Li et al. introduced Cooperative Neural Information Retrieval (CNIR), where queries are first enriched with relevant entities from external knowledge bases and then passed through a BERT-based retrieval pipeline. These methods illustrate how knowledge-aware and task-specific enhancements can augment transformer-based reformulation. In e-commerce settings, Agrawal et al. applied the T5 model for product search by treating query reformulation as a generative task. Their approach integrates reinforcement learning to optimize query generation based on user feedback. The generative capacity of T5 enables the model to propose diverse query variants, while reinforcement learning adapts these suggestions to user behavior, improving the alignment of search results with evolving intent. Kaist et al. explored the integration of GPT-4 [14] with structured knowledge sources to enhance query suggestions. By fine-tuning GPT-4 with user interaction data and external ontologies, the system generates queries that are both contextually and semantically enriched. This framework addresses personalization and context sensitivity without relying solely on explicit user profiling. A related approach, Generative Query Recommendation (GQR) by Cai et al. [6], leverages large language models (LLMs) such as GPT to generate query recommendations from prompt examples. This system bypasses the need for large labeled datasets or complex indexing structures, making it particularly suitable for cold-start scenarios. GQR demonstrates that competitive performance can be achieved using pre-trained generative models with minimal fine-tuning. Transformer models have also been used in conversational search systems. Aliannejadi et al. [3] developed a BERT-based clarifying question generation system for open-domain information-seeking conversations. By fine-tuning BERT on dialogue logs, the model generates context-specific clarifications for ambiguous queries. This enhances the user interaction experience and improves retrieval precision by narrowing down user intent through natural question generation.

Transformer-based models have redefined the landscape of query refinement and suggestion by introducing mechanisms that capture both local and global context with high fidelity. Their self-attention architecture enables modeling of complex dependencies across sequences without the bottlenecks of recurrence, allowing for faster training and better generalization. Pre-trained transformers such as BERT, BART, T5, and GPT-4 provide strong out-of-the-box performance, while task-specific fine-tuning or integration with external knowledge further enhances their utility. In contrast to earlier models that offered single-vector query representations, transformer models can represent multiple intents and subtopics within a query and generate personalized outputs tailored to user profiles. Their applications span query expansion, suggestion, clarification, and reformulation across domains ranging from web search and e-commerce to scientific literature and dialogue systems. However, these models are computationally expensive and may require large-scale datasets and infrastructure to fine-tune effectively. Despite

these challenges, transformers remain the current state-of-the-art in neural query refinement and are poised to evolve further with continued advancements in large language models and efficient transformer variants. In the following section, we turn our attention to applications of query refinement models, exploring how they are deployed in real-world systems across different search environments.

3. APPLICATIONS

As discussed in the preceding sections, query refinement plays a central role in enhancing the effectiveness of information retrieval systems across a wide range of domains. By improving the quality and clarity of user queries, these methods help bridge the gap between user intent and system understanding, resulting in more relevant search results, greater user satisfaction, and improved engagement. This section explores specific applications of query refinement in various domains, including information and product search, music and job search, and personalized retrieval.

3.1 Information and Product Search

Query refinement significantly enhances the relevance and precision of search results in domain-specific retrieval environments, such as product search, music discovery, and job search. These applications often involve large, dynamic content collections and highly varied user intents, making effective query processing essential. In job search, the challenge lies in mapping underspecified user queries to job opportunities that align with career profiles. Zhong et al. [59] proposed a personalized query suggestion model for LinkedIn, incorporating structured data from user profiles and unstructured query text. The integration of professional attributes enables more accurate matching and personalization. Similarly, Zhou et al. [64] developed a method that models both short-term (local flow) and long-term (global flow) user intent by incorporating user feedback and semantic data from resumes and job postings, enabling adaptive and user-centric job search refinement. In e-commerce, predicting user intent is complicated by ambiguous queries, sparse feedback, and rapidly changing product catalogs. Hirsch et al. [26] analyzed query reformulations in eBay search logs and identified three main strategies—add, remove, and replace—with 'replace' being the most prevalent. Their model predicts whether a query will be reformulated before the results are retrieved, offering a foundation for proactive refinement. To further improve product coverage, Agrawal et al. [1] proposed a hybrid approach combining large language model (LLM) generation with reinforcement learning. This model generates diverse reformulations that are semantically consistent with the original query while covering a broader product range, outperforming traditional supervised and reinforcement-only baselines. Music search, particularly in platforms like Spotify, faces unique challenges due to incomplete queries and the live, keystroke-level nature of user interaction. As Spotify's catalog has expanded to include podcasts and audiobooks, user queries have become increasingly ambiguous. To address this, Liu et al. [34] proposed three approaches: co-occurrence mining from logs, classification of query completeness, and graph learning on item metadata. The final ranking of suggestions is conducted using a pointwise ranking model. Their findings show that co-occurrence-based suggestions are most frequently shown and clicked, validating the utility of lightweight, context-sensitive refinement strategies in real-time environments.

Across these domains, query refinement enhances discovery and retrieval by accommodating domain-specific query behavior, leveraging implicit feedback, and managing ambiguity. The success of these models often depends on the integration of session history, semantic metadata, and personalization signals. While structured data (e.g., user profiles, item metadata) provides valuable features, adaptive learning from session logs and reinforcement-based generation improves diversity and personalization, especially in dynamic and intent-sensitive environments.

3.2 Personalization

In many retrieval systems, the same query can reflect drastically different intentions depending on the user's background, preferences, or current needs. For instance, the query "Apple" may refer to a tech company for one user and a fruit for another. This variability necessitates personalized search models that infer user intent by analyzing search history, behavior, or context. Early personalized search approaches [15, 9, 53] relied primarily on click-based or topic-based profiles derived from historical queries. More recent neural approaches [24, 36] build dynamic user profiles using deep learning, allowing models to adapt to short- and long-term interest patterns. However, query ambiguity and noisy input remain obstacles. Zhou et al. [62] addressed this by combining query disambiguation and personalized language modeling. Their framework incorporates both short-term session features and long-term user behavior to predict and refine user intent. Instead of relying on explicit user profiles, Yao et al. [56] proposed training personalized word embeddings, which enable different semantic representations of the same word based on individual users. This approach addresses semantic ambiguity directly within the representation space, improving personalization without requiring structured profiles. Deng et al. [21] introduced a dual-feedback model, which incorporates both positive and negative behaviors to construct a more nuanced intent representation. Feedback is extracted from current sessions and long-term behavior, resulting in more precise personalization during ranking. Further extending this direction, Baek et al. [7] used interaction histories to augment prompts for large language models (LLMs), constructing entity-centric knowledge repositories per user. These repositories are then used to personalize search responses by injecting user-specific context into the prompt. Zhou et al. [63] addressed data sparsity in personalization by introducing a contrastive sampling framework. The model performs sentence-level and sequence-level encoding, learning query and document representations as well as behavior sequences from query logs. For users with insufficient data, Zhou et al. [61] proposed incorporating friend networks into the model, assuming that users with social ties often share interests. This group-based personalization method effectively improves ranking in cold-start scenarios. In platform-specific applications such as LinkedIn, Zhong et al. [59] proposed a query suggestion model that integrates user profiles with initial queries to provide more relevant recommendations. Zhou et al. [64] extended this idea by leveraging semantic content from resumes and job postings, improving both personalization and retrieval coverage within the professional search domain.

Personalization in query refinement has advanced from static, profile-based models to dynamic, learning-based systems capable of modeling ambiguity and evolving interests. Neural methods allow for adaptive modeling of user behavior, while embedding-based approaches introduce fine-grained semantic variation at the lexical level. However, effective personalization remains dependent on high-quality behavioral data. In data-sparse environments, models incorporating social networks or contrastive sampling provide promising solutions. Collectively, these techniques improve user satisfaction by aligning search results more closely with individualized intent.

4. COMPARATIVE ANALYSIS

This section provides a detailed comparative analysis of the neural architectures discussed, addressing their strengths, weaknesses, and performance characteristics.

Table 1. : Comparison of Model Types for Query Refinement

Model Type	Semantic Understanding	Context Modeling	Computational Efficiency	Personalization	Scalability
Embedding Models	High	Low	High	Medium	High
RNNs/GRUs	Medium	High	Medium	High	Medium
Seq2Seq	High	High	Low	High	Low
Transformers	Very High	Very High	Low	Very High	Low

4.1 Performance Characteristics Summary

The comparative table 1 indicates the core strengths and weaknesses of four major neural architectures—Embedding Models, RNNs/GRUs, Seq2Seq, and Transformers—across five critical dimensions: semantic understanding, context modeling, computational efficiency, personalization, and scalability.

Semantic Understanding: Transformers rank highest due to their contextualized representations and multi-head attention, enabling nuanced meaning capture across diverse queries. Embedding models and Seq2Seq architectures achieve strong results in static or moderately dynamic contexts, while RNNs/GRUs show moderate capability due to their sequential processing constraints. **Context Modeling:** Context modeling ability increases with architectural complexity. Transformers lead with global self-attention, while RNNs/GRUs and Seq2Seq leverage temporal or encoder-decoder dependencies effectively. Embedding models lag here, as they lack mechanisms to capture query sequence dependencies. **Computational Efficiency:** Efficiency is inversely related to modeling sophistication. Embedding models are the most efficient due to simple vector lookups, followed by RNNs/GRUs, which are moderately efficient but limited by sequential inference. Seq2Seq and Transformer models incur significant latency and memory costs, especially in long-query scenarios. **Personalization:** Transformers and RNNs/GRUs are most adaptable to personalization, integrating user history and behavioral signals effectively. Seq2Seq models also adapt well in personalized generation contexts. Embedding models can incorporate user-specific vectors but lack deep contextual integration. **Scalability:** Embedding models excel in large-scale deployments due to minimal computational overhead, while RNNs/GRUs scale moderately well with careful optimization. Seq2Seq and Transformers face scalability challenges tied to their computational complexity, though optimizations (e.g., distillation, sparse attention) can mitigate these limitations. The table highlights a core trade-off—architectures that excel in semantic depth and personalization (e.g., Transformers) tend to incur high computational and scalability costs, while architectures optimized for speed and scale (e.g., Embedding Models) sacrifice context and adaptability. For real-world query refinement, hybrid architectures often combine these approaches, using efficient models for retrieval and more sophisticated models for re-ranking or reformulation.

4.2 Empirical Performance Analysis

Query Expansion and Reformulation Accuracy: Comparative studies reveal distinct performance patterns across architectures. Roy et al. [48] demonstrated that embedding-based query expansion using Word2Vec achieves 15-18% improvement in MAP (Mean Average Precision) over traditional pseudo-relevance feedback methods. However, the static nature of these embeddings limits their effectiveness in ambiguous contexts. RNN-based approaches show consistent improvements in session-based scenarios. Ahmad et al. [2] reported that context-aware RNN models with attention mechanisms achieve 22-28Sequence-to-sequence models excel in generative reformulation tasks. Dehghani et al. [20] showed that seq2seq models with copy mechanisms achieve 31Transformer-based models consistently achieve the highest accuracy across diverse tasks. Muster et al. [44] demonstrated that BERT-BART combined frameworks outperform RNN-based systems by 35-42

Table 2. : Domain-Specific Architectural Suitability for Query Refinement

Domain	Key Requirements	Most Suitable Architectures	Representative References
Web Search	Handling heterogeneous queries, semantic disambiguation, session and long-term context modeling	Transformer-based models (e.g., BERT, hierarchical transformers) with hybrid retrieval+re-ranking pipelines	Zhou et al. [?], Nogueira et al. [?]
E-commerce Search	Product catalog structure, query expansion for discovery, handling brand-specific terms	Seq2Seq with copy mechanisms, transformer-based generative reformulation models	Agrawal et al. [?], Dehghani et al. [20]
Music & Media Retrieval	Low latency, real-time interaction, balance between accuracy and speed	Lightweight co-occurrence models, RNN-based models with attention for moderate context modeling	Liu et al. [?], Chen et al. [?]
Professional & Job Search	High personalization, semantic matching between job requirements and profiles, long-form query handling	Transformer-based models incorporating structured profile data	Zhong et al. [59], Sun et al. [?]

4.3 Domain-Specific Architectural Considerations

Analysis from table 2 shows that: **Web Search Context:** The heterogeneous nature of web search queries presents unique challenges that favor different architectural approaches. Transformer-based models like BERT demonstrate strong performance in this domain due to their ability to handle diverse query types and semantic ambiguity. Zhou et al. [62] showed that hierarchical transformer architectures effectively capture both short-term session context and long-term user preferences in web search scenarios. However, the computational requirements of transformers necessitate careful deployment strategies, with many systems using hybrid approaches that combine fast embedding-based retrieval with selective transformer re-ranking. **E-commerce Search Characteristics:** Product search environments exhibit distinct patterns that influence architectural effectiveness. The structured nature of product catalogs and the importance of query expansion for product discovery make sequence-to-sequence models particularly valuable. Agrawal et al. [1] demonstrated that generative reformulation approaches achieve superior product coverage compared to traditional expansion methods. The challenge of handling product-specific terminology and brand names also favors models with copy mechanisms, as implemented in the seq2seq frameworks discussed by Dehghani et al. [20]. **Music and Media Retrieval:** Interactive media platforms face unique constraints that prioritize response time alongside accuracy. Liu et al. [34] found that lightweight co-occurrence models often match complex neural approaches in user satisfaction metrics while providing significantly faster response times. This domain exemplifies the importance of efficiency-accuracy tradeoffs, where RNN-based models with attention mechanisms provide an optimal balance between context awareness and computational efficiency for real-time recommendation scenarios. **Professional and Job Search:** The professional search domain benefits significantly from personalization capabilities, making it well-suited for transformer-based approaches when computational resources allow. Zhong et al. [59] demonstrated that incorporating structured profile data into personalized query suggestion models substantially improves relevance. The longer-form nature of professional queries and the importance of semantic matching between job requirements and candidate profiles favor architectures with strong contextual understanding capabilities.

5. CONCLUSION

Query refinement continues to be a central component in improving the effectiveness, personalization, and usability of information retrieval systems. As user queries remain short, ambiguous, and often underspecified, the need for robust refinement mechanisms that can interpret, reformulate, and personalize search inputs has become increasingly critical.

This survey provided an in-depth examination of the model architectures that underpin modern query refinement techniques, highlighting the transition from traditional non-neural methods to advanced neural approaches. The analysis revealed several key findings:

Architectural Evolution: The progression from embedding-based models through RNNs to transformer architectures demonstrates increasing sophistication in semantic understanding and context modeling. Transformer models achieve 30-50% improvement in

MRR over baseline methods, representing the current state-of-the-art, though at significant computational cost.

Performance Trade-offs: The comparative analysis reveals fundamental trade-offs between accuracy, efficiency, and scalability. While transformer models excel in semantic understanding, embedding-based approaches remain superior for large-scale, latency-sensitive applications. RNN-based models offer the best balance for session-aware applications with moderate computational requirements.

Domain-Specific Adaptations: Applications across product search, job search, music retrieval, and personalized web search demonstrate that effective refinement requires domain-specific adaptations. Performance improvements range from 8-15% when models are fine-tuned for specific domains compared to generic approaches.

Personalization Impact: The integration of user behavior and preferences consistently improves refinement quality by 12-18% across all architectures, with transformer-based personalized models achieving the highest performance on benchmark datasets.

Despite significant progress, several challenges remain. These include handling sparse or noisy user behavior data, scaling personalized models to large populations, managing computational costs associated with large transformer models, and addressing issues of interpretability. Moreover, emerging applications—such as conversational search, multimodal retrieval, and interactive recommendation—demand new refinement models that can operate across modalities, time, and intent shifts.

Future research should explore more adaptive, user-aware, and explainable refinement strategies, particularly those that can generalize across domains with minimal supervision. The integration of large language models with structured knowledge, user interaction data, and reinforcement learning presents a promising direction for building more intelligent and responsive refinement systems.

Through this comprehensive survey, the authors aimed to provide both architectural foundations and practical deployment insights for query refinement techniques. By bridging the gap between theoretical advances and real-world applications, this work supports the development of more effective, context-sensitive, and human-centered retrieval systems.

6. REFERENCES

[1] Sanjay Agrawal, Srujana Merugu, and Vivek Sembium. Enhancing e-commerce product search through reinforcement learning-powered query reformulation. In *Proceedings of the 32nd ACM International CIKM*, 2023.

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd international ACM SIGIR*, 2019.

[3] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir*, 2019.

[4] Rajul Anand and Alexander Kotov. An empirical comparison of statistical term association graphs with dbpedia and conceptnet for query expansion. In *Proceedings of the 7th Annual Meeting of the Forum for Information Retrieval Evaluation*, 2015.

[5] Hiteshwar Kumar Azad and Akshay Deepak. A new approach for query expansion using wikipedia and wordnet. *Information sciences*, 2019.

[6] Andrea Bacciu, Enrico Palumbo, Andreas Damianou, Nicola Tonellotto, and Fabrizio Silvestri. Generating query recommendations via llms. *arXiv preprint arXiv:2405.19749*, 2024.

[7] Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM Web Conference 2024*, 2024.

[8] Jing Bai, Dawei Song, Peter Bruza, Jian-Yun Nie, and Guihong Cao. Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM international CIKM*, 2005.

[9] Paul N Bennett, Filip Radlinski, Ryen W White, and Emine Yilmaz. Inferring and using location metadata to personalize web search. In *Proceedings of the 34th international ACM SIGIR*, 2011.

[10] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. Query suggestions in the absence of query logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[11] Alexey Borisov, Martijn Wardenaar, Ilya Markov, and Maarten de Rijke. A click sequence model for web search. SIGIR '18. Association for Computing Machinery.

[12] Fei Cai, Maarten De Rijke, et al. A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval*, 2016.

[13] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[14] Kaibo Cao, Chunyang Chen, Sebastian Baltes, Christoph Treude, and Xiang Chen. Automated query reformulation for efficient search based on query logs from stack overflow. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE.

[15] Mark J Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international CIKM*, 2010.

[16] Claudio Carpineto, Renato De Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, 2001.

[17] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 2012.

[18] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. A context-aware click model for web search. In *Proceedings of the 13th international conference on web search and data mining*, 2020.

[19] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. Hierarchical neural query suggestion with an attention mechanism. *Information Processing & Management*, 2020.

[20] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.

[21] Chenlong Deng, Yujia Zhou, and Zhicheng Dou. Improving personalized search with dual-feedback network. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics*, 2019.

[23] Pierre Erbacher, Ludovic Denoyer, and Laure Soulier. Interactive query clarification and refinement via user simulation. In *Proceedings of the 45th International ACM SIGIR*, 2022.

[24] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. Personalizing search results using hierarchical rnn with query-aware attention. In *Proceedings of the 27th ACM international CIKM*, 2018.

[25] Helia Hashemi, Hamed Zamani, and W Bruce Croft. Learning multiple intent representations for search queries. In *Proceedings of the 30th ACM International CIKM*, 2021.

[26] Sharon Hirsch, Ido Guy, Alexander Nus, Arnon Dagan, and Oren Kurland. Query reformulation in e-commerce search. In *Proceedings of the 43rd International ACM SIGIR*, pages 1319–1328, 2020.

[27] Nick Jardine and Cornelis Joost van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information storage and retrieval*, 1971.

[28] Jyun-Yu Jiang and Wei Wang. Rin: Reformulation inference network for context-aware query suggestion. In *Proceedings of the 27th ACM international CIKM*, 2018.

[29] Karen Sparck Jones. *Automatic keyword classification for information retrieval*. Archon Books, 1971.

[30] Payam Karisani, Maseud Rahgozar, and Farhad Oroumchian. A query term re-weighting approach using document similarity. *Information Processing & Management*, 2016.

[31] Andisheh Keikha, Faezeh Ensan, and Ebrahim Bagheri. Query expansion using pseudo relevance feedback on wikipedia. *Journal of Intelligent Information Systems*, 2018.

[32] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[33] Ruirui Li, Liangda Li, Xian Wu, Yunhong Zhou, and Wei Wang. Click feedback-aware query recommendation using adversarial examples. In *The World Wide Web Conference*, 2019.

[34] Alva Liu, Humberto Jesús Corona Pampin, and Enrico Palumbo. Bootstrapping query suggestions in spotify's instant search system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[35] Shuang Liu, Fang Liu, Clement Yu, and Weiyi Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proceedings of the 27th annual international ACM SIGIR*, 2004.

[36] Shuqi Lu, Zhicheng Dou, Xu Jun, Jian-Yun Nie, and Ji-Rong Wen. Psgan: A minimax game for personalized search with limited and noisy click data. In *Proceedings of the 42nd international ACM SIGIR*, 2019.

[37] Yuanhua Lv and ChengXiang Zhai. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[38] Melvin Earl Maron and John L Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 1960.

[39] Alan Medlar, Jing Li, and Dorota Głowacka. Query suggestions as summarization in exploratory search. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, 2021.

[40] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[41] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[42] Jack Minker, Gerald A Wilson, and Barbara H Zimmerman. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, 1972.

[43] Bhaskar Mitra. Exploring session context using distributed representations of queries and reformulations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015.

[44] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. Using bert and bart for query suggestion. In *Joint Conference of the Information Retrieval Communities in Europe*, 2020.

[45] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD*, 2017.

[46] Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. A survey of text representation and embedding techniques in nlp. 2023.

[47] Joseph John Rocchio Jr. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, 1971.

[48] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, 2016.

[49] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015.

[50] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international CIKM*, 2015.

[51] Cornelis Joost Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of documentation*, 1977.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

[53] Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. Search personalization with embeddings. In *European Conference on Information Retrieval*. Springer, 2017.

[54] Sida Wang, Weiwei Guo, Huiji Gao, and Bo Long. Efficient neural query auto completion. In *Proceedings of the 29th ACM International CIKM*, 2020.

[55] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Query suggestion with feedback memory network. In *Proceedings of the 2018 World Wide Web Conference*, 2018.

[56] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. Employing personal word embeddings for personalized search. In *Proceedings of the 43rd international ACM SIGIR*, 2020.

[57] Jiuling Zhang, Beixing Deng, and Xing Li. Concept based query expansion using wordnet. In *2009 international e-conference on advanced science and technology*. IEEE, 2009.

[58] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. Contextualized query expansion via unsupervised chunk selection for text retrieval. *Information Processing & Management*, 2021.

[59] Jianling Zhong, Weiwei Guo, Huiji Gao, and Bo Long. Personalized query suggestions. In *Proceedings of the 43rd International ACM SIGIR*, 2020.

[60] Dong Zhou, Séamus Lawless, Xuan Wu, Wenyu Zhao, and Jianxun Liu. A study of user profile representation for personalized cross-language information retrieval. *Aslib Journal of Information Management*, 2016.

[61] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. Group based personalized search by integrating search behaviour and friend network. In *Proceedings of the 44th international ACM SIGIR*, 2021.

[62] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. Encoding history with context-aware representation learning for personalized search. In *Proceedings of the 43rd international ACM SIGIR*, 2020.

[63] Yujia Zhou, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. Pssl: self-supervised learning for personalized search with contrastive sampling. In *Proceedings of the 30th ACM international CIKM*, 2021.

[64] Zile Zhou, Xiao Zhou, Mingzhe Li, Yang Song, Tao Zhang, and Rui Yan. Personalized query suggestion with searching dynamic flow for online recruitment. In *Proceedings of the 31st ACM International CIKM*, 2022.