Query Attribute Modeling: Improving search relevance with Semantic Search and Meta Data Filtering

Karthik Menon*, Batool Arhamna Haider, Muhammad Arham*,
Kanwal Mehreen, Ram Mohan Rao Kadiyala, Muhammad Ali Shafique, Hamza Farooq
traversaal.ai
USA
Email: {karthik, batool, muhammad.arham, ram, ali, hamza}@traversaal.ai, kanwalmehreen2000@gmail.com

***Abstract*—The exponential growth of e-commerce has created vast product catalogs where traditional keyword, semantic or hybrid search struggles to balance precision with relevance, frequently overlooking attribute constraints or misinterpreting user intent. This study introduces Query Attribute Modeling (QAM), a hybrid framework that enhances search precision and relevance through a two-step process. First, item descriptions and titles are decomposed into structured attributes and stored as metadata key-value pairs alongside item description embeddings. Second, QAM decomposes open text queries into structured metadata tags and semantic elements, enabling focused retrieval by automatically extracting metadata filters from free-form text queries and reducing noise. Experimental evaluation using the Amazon Toys Reviews dataset (10,000 unique items with 40,000+ reviews and detailed product attributes) demonstrated QAM's superior performance, achieving a mean average precision at 5 (mAP@5) of 52.99%. QAM showed substantial improvements: 28.67% over BM25 keyword-based search, 6.5% over semantic search, 8.58% over cross-encoder reranking, and 9.96% over hybrid search combining encoder embeddings and BM25 results using Reciprocal Rank Fusion. The results establish QAM as a robust solution for Enterprise Search applications, particularly in e-commerce systems.**

***Index Terms*—Information Retrieval, Large Language Models, Metadata Filtering**

I. Introduction

The evolution of search engines has progressed from basic retrieval systems to advanced models capable of understanding context and semantics. In its infancy, search engines were primarily concerned with the retrieval of information, employing crawling, indexing, and ranking mechanisms to facilitate access to indexed web pages. Although revolutionary, this initial paradigm lacked the ability to discern contextual relevance and user intent, leading to a search experience that often failed to meet user expectations [1].

During the mid-1990s [2], a paradigm shift occurred with the emergence of keyword-based search. This approach, epitomized by search engines such as Excite [3] and WebCrawler [4], allowed users to retrieve information based on specific keywords or phrases. However, they exhibit notable weaknesses, such as a lack of understanding of the semantic meaning of queries, which can result in irrelevant results when keywords have multiple meanings [5]. This shortcoming highlighted the need for more advanced search technologies capable of interpreting the intent and contextual meaning behind queries.

Later era witnessed the emergence of semantic search with methods like Latent Semantic Analysis Theory [6] and TexLexAn [7]. By incorporating natural language processing and machine learning techniques, semantic search systems aimed to provide more accurate and contextually relevant results, marking a departure from simplistic keyword matching paradigms and ushering in a new era of search sophistication and user-centricity. However, semantic search also encounters challenges, including managing language ambiguity, ensuring scalability, and addressing computational overhead, which can result in incomplete or inaccurate results, particularly in complex, real-world scenarios [8].

In recent years, hybrid search [9] has emerged as a synergistic fusion of keyword-based precision and semantic contextual understanding. This hybrid approach combines the strengths of both the keyword-based and semantic search approaches, thus enhancing the overall search experience for users. Despite its promise, challenges persist in integrating keyword and semantic results, particularly in scenarios involving complex queries and rich metadata like, *"I am looking for educational toys specifically from LEGO, designed to promote creativity, suitable for children aged 5-8"* and *"Locate a top-rated board game from Hasbro for kids aged 9-12 within a budget of $40"*.

Against this backdrop of evolving search methodologies, Query Attribute Modeling (QAM) emerges as a new paradigm designed to redefine enterprise search. QAM introduces a novel framework that harmonizes the semantic and keyword-based capabilities, addressing the inherent limitations of existing search systems. By systematically dissecting user queries into structured metadata tags and semantic components, QAM enables a more precise and contextually relevant interpretation of user queries.

The primary objective of this research is to demonstrate how Query Attribute Modeling enhances search precision and relevance based on user open text search. Through detailed

*Equal contribution

experimentation and analysis, we aim to showcase its potential to transform enterprise search by addressing the challenges of scalability, efficiency, and adaptability in handling complex real-world queries. The following sections outline the methodology (Section II), experimentation (Section III), and results (Section IV), highlighting QAM's effectiveness in meeting the growing demands of modern search technologies.

II. Methodology

The methodology employed in our research follows a systematic approach to enhance the precision and relevance of search results within the context of Query Attribute Modeling (QAM). It comprises of four distinct steps, each designed to address specific aspects of search refinement and optimization, as shown in Figure 1.

Algorithm 1 QAM Algorithm

```
Require: Query $Q$, Dataset $D$
Input: $Q =$ "A long black dress from Zara under \$100"
Output: Ranked search results $R$

Step 1: Query Decomposition
$Q_{\text{metadata}} \leftarrow$ Extract metadata tags (e.g., color, brand)
$Q_{\text{semantics}} \leftarrow$ Extract semantic elements

Step 2: Metadata Filtering
$D_{\text{filtered}} \leftarrow \{p \in D \mid p.\text{metadata matches } Q_{\text{metadata}}\}$

Step 3: Review Similarity
for each product $p \in D_{\text{filtered}}$ do
    $p$.score $\leftarrow$ CosSim(Enc($Q_{\text{semantics}}, p$))
end for

Step 4: Final Ranking
for each product $p \in D_{\text{filtered}}$ do
    $p$.final_score $\leftarrow$ CrossEncoder($Q, p$)
end for
$R \leftarrow$ Sort($D_{\text{filtered}}$, by $= p$.final_score)
return Top-$N$ results from $R$
```

A. Query Decomposition

The first step focuses on dissecting user queries into two primary components: metadata tags and semantic elements. This decomposition enables the search system to separate explicit user requirements (e.g., "color" or "brand")) from the deeper contextual meaning of the query. To achieve this, we employ a language model (e.g., GPT-4o) [10], which excels in parsing complex queries and extracting structured information.

- **Metadata Tags:** These include structured attributes such as product brand, material, price constraints, and preferred user demographics (e.g., age groups). These tags provide a structured way for filtering datasets effectively.
- **Semantic Elements:** These capture the contextual intent of the query, allowing the system to understand implicit preferences and refine results accordingly.

B. Metadata Filtering for Enhanced Search Precision

Building upon the extracted metadata tags, the subsequent step focuses on enhancing search precision by using these tags to filter the dataset and retain only the most relevant items. Metadata attributes such as material, brand, and color play a crucial role in this filtering process. For instance, in a query like "a little black dress," the system utilizes the extracted metadata tag "black" & "Zara" to exclude irrelevant results, such as dresses of other colors or brands. Similarly, filtering by material and brand ensures that user preferences are prioritized early in the pipeline, reducing computational overhead for subsequent steps. This method enhances both efficiency and precision by eliminating noise from the dataset. Metadata filtering has been shown to be a lightweight yet impactful technique for aligning search results with user intent [11].

C. Query and Product Description Similarity Search

This step employs semantic embeddings and cosine similarity to connect user queries with relevant qualitative information in product reviews. Semantic embeddings, generated using advanced models like nomic-embed-text-v1 [12], encode the contextual meaning of the query and reviews into vector representations. Cosine similarity is then calculated to measure how well a product aligns with the user's intent. For example, if a query specifies *"suitable for formal events,"* this step prioritizes products with reviews mentioning *"formal occasions"*. By linking the subjective components of the query with qualitative descriptions in the reviews, this step deepens the system's understanding of user requirements and enhances result relevance. This builds on existing methodologies using contextual review analysis to improve search outcomes [13].

D. Final Ranking

The final step integrates the outputs of the previous phases to deliver the most relevant results. A cross-encoder model, such as msmarco-MiniLM-L12-en-de-v1 [14], is employed to compute the final relevance score for each product. Unlike bi-encoders, which generate separate embeddings for queries and products and compute relevance scores based on their similarity, cross-encoders process the query and product together, directly modeling their interaction. This approach allows cross-encoders to capture finer-grained relationships between the query and product, leading to more accurate rankings [15]. For each product in the filtered dataset, the cross-encoder computes a final score based on the semantic similarity between the query and product attributes. The results are then sorted by these scores to produce a ranked list of items, ensuring that the most relevant results are prioritized. This step ensures the delivery of highly personalized and contextually relevant search results.

III. Experimentation

A. Data

The experimentation phase utilized the *Amazon Toys Reviews* dataset, which consists of 10,000 unique items with

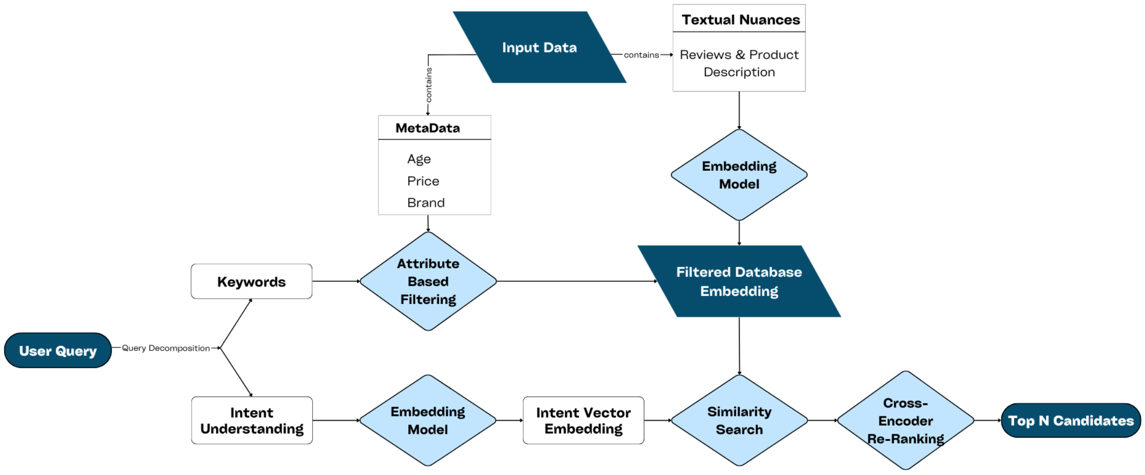

Fig. 1. Query Attribute Modeling (QAM) methodology, illustrating the four-stage process of query understanding, metadata filtering, semantic search, and re-ranking of results from the filtered dataset.

product descriptions and 40,000+ reviews including 15 raw and engineered features. This data set was chosen for its extensive coverage of product reviews, which facilitates a detailed analysis at the review level for each product.

In addition to reviews, a significant focus was placed on feature extraction from product descriptions. This involved extracting essential attributes such as brand and required minimum age. To achieve this, advanced text preprocessing techniques were applied, using natural language processing (NLP) libraries such as NLTK and spaCy. These techniques enabled the extraction of pertinent information from the textual descriptions, enriching the dataset with valuable metadata.

To evaluate QAM and its competing methods, a diverse set of 1,000 queries was generated using GPT-4o. These queries were designed to simulate realistic user searches, capturing both explicit requirements (e.g., brand, price, age) and subjective intent (e.g., suitability for specific occasions). Out of the generated queries, 200 high-quality queries were selected for the evaluation dataset to ensure alignment of brand names and attributes with the entries in the original Amazon dataset. Examples include: *"Can I find Playteachers toys for kids aged 6 to 15?"* and *"Looking for a Kaleidoscope toy for my 3-year-old, priced around $12."* This carefully curated query set provided a robust basis for evaluating QAM's hybrid approach to address both explicit preferences and contextual query intent.

B. Evaluation Setup

The evaluation involved running each query against five search methods: BM25 keyword-based search [16], semantic search [8], cross-encoder re-ranking, hybrid search, and QAM. Each method returned the top 10 results, which were annotated for relevance using an LLM (GPT-4o).

Annotation Process: The LLM was given both the query and the returned results and was tasked with determining whether each result was relevant. The relevance was based on the following:

- Exact match for metadata (e.g., price, brand). For quantitative values including rating and price, we allowed for a 20% percent complacency between the returned value and the required value to allow flexibility in responses.
- Semantic alignment for contextual preferences.

Scoring Metrics: The annotated results were evaluated using precision@k (P@k) and mean average precision@k (mAP@k). These metrics captured the accuracy and ranking quality of each method [17].

Precision at k (P@k) measures the ratio of relevant items among the top K results, as shown in (1).

$$Precision@k = \frac{\text{Relevant Results @k}}{k} \tag{1}$$

Average Precision@K (AP@K) calculates the precision at each rank where a relevant item appears, averaged over all relevant items, as defined in (2).

$$AP@K = \frac{1}{\min(K, \text{Total Relevant Items})} \sum_{i=1}^{K} P(i) \cdot rel(i) \tag{2}$$

AP@k score values the ranking of retrieved results, returning a higher score if relevant data points are ranked higher than non-relevant results. Mean Average Precision (mAP@K) computes the mean of AP@K scores across all data samples, providing an aggregate score for all queries, as given in (3).

$$mAP@K = \frac{1}{N} \sum_{q=1}^{N} AP@K_q \tag{3}$$

The use of an LLM as a judge automated the annotation process, reducing human bias and ensuring consistent evaluation standards [18]. Additionally, for complex queries and certain metadata combinations, the QAM search method may

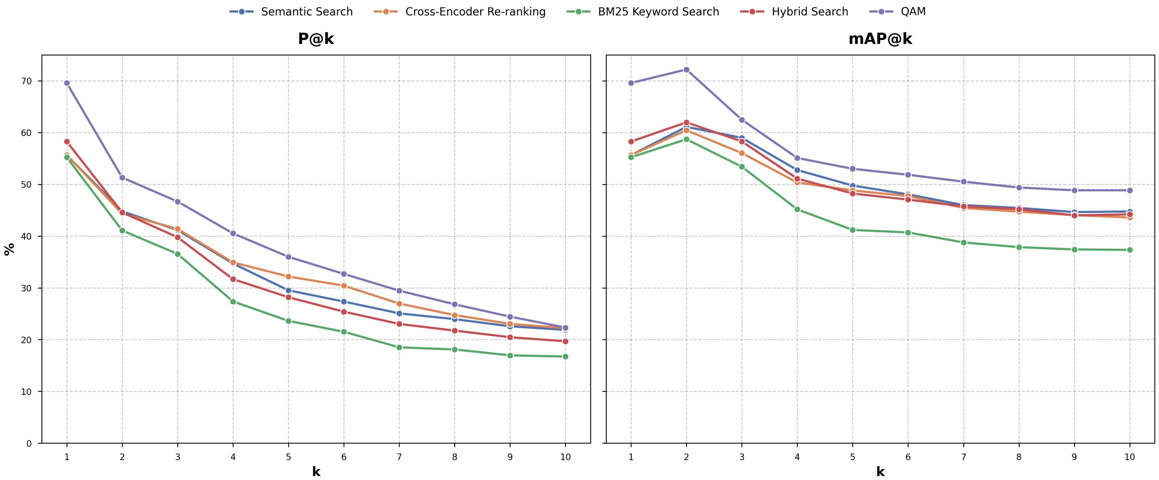

Fig. 2. Comparison of Precision and mAP scores for five retrieval methods across varying k values. The y-axis represents the percentage of relevant documents among the top-k results, with QAM outperforming all other methods across all k values.

significantly reduce the candidate set, yielding fewer than k retrieved results. To ensure an unbiased evaluation, we restricted our analysis to search instances where at least k relevant documents were available. In cases where fewer than k relevant results existed, the missing results were treated as non-relevant, thereby penalizing our approach for failing to retrieve the desired number of relevant documents.

TABLE I
PRECISION@K SCORES ACROSS METHODS

Method	P@3	P@5	P@10
Keyword Search	36.55%	23.62%	16.74%
Semantic Search	41.15%	29.52%	21.89%
Re-Ranking	41.38%	32.19%	22.21%
Hybrid Search	39.77%	28.19%	19.68%
QAM	**46.67%**	**36.00%**	**22.32%**

[a] **Bold** indicates highest precision for each metric.

TABLE II
MEAN AVERAGE PRECISION (MAP@K) SCORES

Method	mAP@3	mAP@5	mAP@10
Keyword Search	53.39%	41.19%	37.33%
Semantic Search	58.97%	49.75%	44.75%
Re-Ranking	56.03%	48.81%	43.59%
Hybrid Search	58.28%	48.22%	44.20%
QAM	**62.47%**	**52.99%**	**48.84%**

[a] **Bold** indicates highest mAP for each column.

IV. RESULTS AND ANALYSIS

The results demonstrate that QAM significantly outperforms traditional search methods.

In terms of Mean Average Precision at 5 (mAP@5), QAM achieved a score of 52.99%, which is consistently higher than the scores of the other methods. Specifically, QAM showed a 28.67% improvement over BM25 keyword-based search (41.19%), 6.5% over semantic search (49.75%), and 8.58% over cross-encoder reranking (48.81%). QAM achieved a 9.96% improvement compared to hybrid search, which combined encoder embeddings and BM25 search results using Reciprocal Rank Fusion (RRF) and scored 48.22%. Table II summarizes the mAP@K scores for all methods.

Furthermore, the comparison of Precision@K, summarized in Table I, across all methods demonstrates the consistent superiority of QAM. Across all values of k (1 to 10), QAM consistently retrieves a higher percentage of relevant results compared to other methods. Figure 2 summarizes these findings, illustrating how P@K and mAP@K vary with k. The results indicate that QAM outperforms all other approaches in the Amazon toy data set retrieval task by effectively filtering out irrelevant results prior to searching, thus improving the overall relevance of the retrieved documents.

Thus, QAM outperforms hybrid search and other retrieval methods in scenarios requiring both specificity and contextual understanding. Unlike hybrid search, which combines multiple ranking signals, QAM's structured approach enhances relevance and retrieval accuracy, making it a more effective solution for modern search challenges.

V. CONCLUSION

In conclusion, this research introduces Query Attribute Modeling (QAM), an innovative framework for enhancing precision and relevance in search systems. By systematically integrating query decomposition, metadata filtering, and contextual analysis, QAM consistently outperforms traditional keyword-based and semantic search methods. For the next phase, we aim to enable the Language Model (LLM) API to autonomously identify relevant keyword tags from user queries, eliminating the need for explicit guidance and enhancing the dynamism of our query deconstruction process. Additionally, the integration of powerful vector databases like Qdrant [19] will streamline information retrieval, contributing to a more sophisticated search experience. We intend to address scalability limitations inherent in manual data labeling by scaling our model to standard databases and a wider array of queries, ensuring stability and robustness across diverse datasets.

References

[1] A. Kumari and J. Thakur, "Semantic web search engines : a comparative survey," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 107–115, 2019.

[2] M. Abu Kausar, V. Dhaka, and S. Singh, "Web crawler: A review," *International Journal of Computer Applications*, vol. 63, pp. 31–36, 02 2013.

[3] "Excite - web portal and search engine," https://www.excite.com/.

[4] "Webcrawler - search engine," https://www.webcrawler.com/.

[5] W. Tannebaum and A. Rauber, "Learning keyword phrases from query logs of uspto patent examiners for automatic query scope limitation in patent searching," *World Patent Information*, vol. 41, pp. 15–22, 2015.

[6] T. K. Landauer, "Learning and representing verbal meaning: The latent semantic analysis theory," *Current Directions in Psychological Science*, vol. 7, no. 5, pp. 161–164, 1998. [Online]. Available: https://doi.org/10.1111/1467-8721.ep10836862

[7] J.-P. Redonnet, "TexLexAn: An open source automatic text summarizer," http://texlexan.sourceforge.net/, 2024, accessed: 26.04.2024.

[8] A. Srivastava, M. Nalluri, T. Lata, G. Ramadas, N. Sreekanth, and H. Vanjari, "Scaling ai-driven solutions for semantic search," 12 2023, pp. 1581–1586.

[9] L. Gao, Z. Dai, and J. Callan, "Complement lexical retrieval model with semantic residual embeddings," in *European Conference on Information Retrieval*. Springer, 2021, pp. 146–160.

[10] OpenAI, "Gpt-4o system card," 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[11] e. a. Ulrich, H., "Understanding the nature of metadata: systematic review," *Journal of Medical Internet Research*, vol. 24, p. e25440, 2022.

[12] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, "Nomic embed: Training a reproducible long context text embedder," 2024. [Online]. Available: https://arxiv.org/abs/2402.01613

[13] D. Limbu, A. Connor, R. Pears, and S. MacDonell, "improving web search using contextual retrieval," pp. 1329–1334, 2009.

[14] H. Face, "Cross-encoder/msmarco-minilm-l12-en-de-v1," https://huggingface.co/cross-encoder/msmarco-MiniLM-L12-en-de-v1, n.d.

[15] X. Qin, X. Liu, X. Zheng, J. Liu, and Y. Zhu, "An empirical study of uniform-architecture knowledge distillation in document ranking," 2023.

[16] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, pp. 333–389, 01 2009.

[17] C. D. Manning, P. Raghavan, and H. Schütze, "Chapter 8: Evaluation in information retrieval," in *Introduction to Information Retrieval*. Cambridge University Press, 2009, p. 161, retrieved 2015-06-14.

[18] C. Head, "Large language model applications for evaluation: opportunities and ethical implications," *New Directions for evaluation*, vol. 2023, pp. 33–46, 2023.

[19] Qdrant, "Qdrant documentation," 2024, accessed: 2024-06-26. [Online]. Available: https://qdrant.tech/documentation/