From Unstructured to Structured: LLM-Guided Attribute Graphs for Entity Search and Ranking

Yilun Zhu
Amazon.com, Inc.
Seattle, USA
yilunzhu@amazon.com

Nikhita Vedula
Amazon.com, Inc.
Seattle, USA
veduln@amazon.com

Shervin Malmasi
Amazon.com, Inc.
Seattle, USA
malmasi@amazon.com

Abstract

Entity search, i.e., finding the most similar entities to a query entity, faces unique challenges in e-commerce, where product similarity varies across categories and contexts. Traditional embedding-based approaches often struggle to capture nuanced context-specific attribute relevance. In this paper, we present a two-stage approach combining Large Language Model (LLM)-driven attribute graph construction with graph-aware LLM ranking. In the offline stage, we extract structured product attributes from unstructured text, and construct a reusable attribute graph with category-aware schemas. In the online stage, we rank retrieved candidates by reasoning over this structured representation rather than raw text, reducing per-product token usage by 57% while improving ranking precision. Experiments show that our approach outperforms multiple baselines under zero-shot scenarios, achieving a over 5% improvement in average precision without requiring training data, generalizes robustly across diverse product categories, and shows immense potential for real-world deployment.

CCS Concepts

• **Information systems** → **Language models**; *Query representation*; • **Computing methodologies** → **Information extraction**.

Keywords

Entity Search, Structured Extraction, LLM Ranking

ACM Reference Format:
Yilun Zhu, Nikhita Vedula, and Shervin Malmasi. 2026. From Unstructured to Structured: LLM-Guided Attribute Graphs for Entity Search and Ranking. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3805712.3808501

1 Introduction and Background

Entity Search, the task of retrieving entities most similar to a given query entity, is fundamental in information retrieval [4], particularly in e-commerce where users seek substitutes or comparable products [19, 21]. Given a query product, a system must accurately retrieve and rank other entities that closely match the query's characteristics. Although product information typically exists as unstructured text (e.g., titles, descriptions), effective entity similarity search relies on structured attribute representations for systematic comparison [18, 28]. A key challenge is extracting structured attributes and values from unstructured data, where the attribute set is dynamic and must adapt as new products and categories emerge [10, 14]. Extracting structured product attributes from unstructured text (e.g., titles, descriptions) has been widely explored via sequence labeling, span-based extraction, and few-shot methods through graph-based inference and external knowledge augmentation [8, 11, 16, 17, 29]. Traditional approaches in industry typically follow two stages, retrieval and ranking, which face further limitations. Collaborative filtering struggles with cold-start problems [27], text-based similarity may miss nuanced attribute differences [25], and creating robust attribute-driven similarity measures remains challenging when item data is unstructured or inconsistent [5]. Dense semantic embedding models and graph neural networks (GNNs) have advanced product-focused entity retrieval systems by capturing semantic relationships in large catalogs and addressing cold-start problems [4, 6, 9, 12, 19, 31].

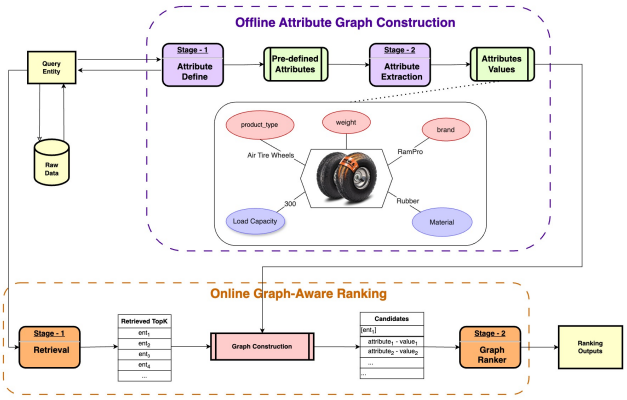

Figure 1: The framework of our Graph Ranker System.

Recent advances in Large Language Models (LLMs) offer a promising avenue. LLMs can serve as powerful zero-shot list-wise rankers of search results [1, 20, 24, 32], and recent work integrates them with structured or semi-structured data (e.g., knowledge graphs and tables) for enhanced reasoning [15, 23, 32]. However, these approaches feed unstructured product descriptions directly to the LLM. At an industrial scale, this poses challenges with respect to cost, latency, efficiency, and the ability to enforce hard constraints like matching on critical attributes and nuanced values when applied naively over raw text.

In this paper, we present an *Attribute-based Graph Ranker*, a hybrid system for enhanced similarity search that addresses the

SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3808501

above identified limitations through a two-stage design as shown in Figure 1: (i) an *offline* stage that uses LLM-based reasoning to extract structured attributes from unstructured product text and constructs a standardized attribute graph, and (ii) an *online* stage that integrates high-recall retrieval with graph-aware LLM ranking, where the LLM reasons over curated attribute sets rather than free-text descriptions, maintaining efficiency and interpretability. We differ from prior work by decoupling knowledge construction from ranking. By reasoning over structured attributes, our system yields two benefits: (1) a 57% reduction in per-product token usage, lowering inference cost and latency, and (2) explicit attribute-value comparisons enabling more precise similarity assessments than free-text reasoning alone. Our contributions are:

(i) **LLM-based Attribute Graph Construction**: An LLM-driven pipeline that defines category-aware attribute schemas, extracts structured attribute-value pairs from unstructured text, constructs a bipartite graph with product entities and attribute nodes linked by value relations, and enables precise attribute-based entity retrieval.

(ii) **Graph-aware LLM Ranking**: An LLM reasons over the structured attributes at inference time, improving ranking precision and interpretability.

(iii) Large scale evaluations show significant precision gains over strong baselines, and strong potential for real-world deployment.

2 Methodology

2.1 Stage 1: Attribute Graph Construction

The goal of this offline *LLM-based Attribute Graph Construction* stage is to enrich and index the massive entity dataset so that the online second stage (Section 2.2) can be fast and effective.[1] This stage includes: (i) Attribute Definition, and (ii) Attribute Extraction. All LLM-based steps utilize Claude 3.5 Sonnet v2 [3], selected based on a preliminary evaluation of 500 examples in which it achieved high precision, while meeting our latency requirements.

Attribute Definition. Different product categories have different characteristic attributes (e.g., "screen size" for TVs vs. "material" for clothing). To guide the extraction of attributes, we define an attribute schema for each product super-category and sub-category, and prompt an LLM to find key attributes relevant to products in that category. We do this at two levels: broad attributes that apply to the super-category (e.g., *Brand*, *Model*, *Dimensions* for "Electronics") and specific attributes for the sub-category (e.g., *Screen Size*, *Storage Capacity*, *Camera Resolution* for "Smartphones"). Hierarchical organization ensures super-category attributes (e.g., *Safety Standards*) remain consistent across sub-categories while sub-category attributes reflect domain-specific granularity. After standardization, we have 61 super-categories and 4,940 sub-categories, with approximately 8–10 attributes per super-category and 6–8 additional per sub-category. We curate these LLM-generated attribute lists through lightweight manual standardization via clustering and deduplication (e.g., merging "Screen Size" and "Display Size"), rather than defining attribute schemas from scratch. For unseen categories, the LLM generates a new schema on-the-fly using the same prompting approach – new sub-categories inherit super-category attributes while the LLM suggests category-specific ones. These predefined attribute sets ensure naming consistency and guide the extraction stage.

Attribute Value Extraction. We then extract attribute values for each entity in the catalog. For each entity, we input the product's textual data (title, description, bullet points) to the LLM with an instruction to output a JSON of attribute-value pairs, using that product category's attribute definition list as a guideline. The LLM extracts values for predefined attributes, and we also allow it to identify additional product-specific attributes from the text.

2.2 Stage 2: Graph-Aware LLM Ranking

During inference, our system receives a query entity ENT_q as input and outputs a ranked list of similar entities $L = [ent'_1, ent'_2, \ldots, ent'_k]$. The inference process leverages structured attributes previously extracted during the offline stage 1. If attributes for the query entity were not computed offline, the system dynamically invokes real-time attribute extraction to generate attribute-value pairs for ENT_q.[2] The online stage includes: (i) candidate generation and (ii) graph-aware LLM ranking.

Candidate Generation. Given the query entity ENT_q, we first retrieve an initial set of candidate entities from the complete catalog D. We employ dense retrieval using the embedding model `bge-m3` [7] to encode both query and candidate entities into dense embeddings based on concatenated product titles, descriptions, and bullet points. We use FAISS [13] for efficient approximate nearest-neighbor search, retrieving the top-K_d candidate entities with the highest cosine similarity scores, forming the initial candidate set C.

Graph-aware LLM Ranking. The final stage ranks candidate entities according to their similarity to the query entity, by reasoning over a bipartite graph $G = (V_P, V_A, E)$, where V_P represents product (entity) nodes, V_A represents attribute nodes, and each edge $e \in E$ connects a product node p to an attribute node a with edge label $val(p, a)$ denoting the attribute value for that product.[3] Given a query entity ENT_q, ranking the candidates in the candidate set C involves implicitly traversing and reasoning over the local subgraph $G_q = (V_{P_q}, V_{A_q}, E_q) \subseteq G$, where $V_{P_q} = \{ENT_q\} \cup C$, and V_{A_q} contains all attributes associated with the entities in V_{P_q}. We formulate the ranking task as a zero-shot, list-wise reasoning problem, prompting Claude 3.5 v2 to compare each candidate's attribute values against the query entity's attribute values. The similarity score $S(ENT_q, ENT_c)$ between query entity ENT_q and candidate entity $ENT_c \in C$ is computed implicitly by the LLM via assessing the edges connecting both entities to common attribute nodes:

$$S(ENT_q, ENT_c) = \mathcal{F}_{\text{LLM}}\big(\{(a,\ val(ENT_q, a),\ val(ENT_c, a)) \mid a \in V_{A_q}\}\big)$$

where $\mathcal{F}_{\text{LLM}}$ represents the LLM's reasoning function, which implicitly assigns importance to attributes and evaluates how well attribute values match or differ between the query and candidate entities. Candidates with stronger attribute alignment to the query entity receive higher similarity scores. We use a fine-grained 0–100

[1] Since category labels are missing from the original dataset, we use an LLM-based approach to predict and standardize these labels to ensure products are compared within the same category.

[2] Newly extracted attributes are cached for efficiency in future queries.

[3] We use attributes as nodes and values as edge labels for efficiency. With millions of products, representing each unique attribute value as a separate node would create an intractably large graph.

similarity scale[4] following established IR evaluation practices [26] and recent LLM evaluation frameworks [30], allowing for more nuanced differentiation of relevance levels than coarser scales. The final ranked list $L = \text{sorted}\{[ent'_1, \ldots, ent'_k]\}$ is produced by sorting candidates according to their similarity scores $S(ENT_q, ENT_c)$. By using structured reasoning over this local attribute graph, our system achieves high-precision, interpretable ranking results suitable for industry applications demanding accuracy and transparency.

All LLM interactions in our pipeline use structured JSON schemas with predefined attribute fields and deterministic settings (temperature=0) for reproducibility. The input to the LLM at each stage consists exclusively of curated product catalog data — sanitized attribute-value pairs during ranking, and product text with a constrained attribute list during extraction. No user-generated free text reaches the LLM, and output is validated against the expected schema. This controlled pipeline significantly mitigates prompt-based adversarial risks. Our prompts were iteratively refined and optimized for the Claude LLM during development, and the structured input format constrains the output space and reduces sensitivity to prompt phrasing compared to free-text prompting. We also note that our evaluation covers general consumer product categories. For highly specialized domains (e.g., medical supplies), some domain-specific fine-tuning or expert-curated attribute schemas may be necessary to achieve comparable extraction quality.

3 Experiments and Results

Method	P@1	P@3	P@5	MRR	mAP
Sparse Retrieval (SR)	40.48	37.30	31.90	50.87	50.43
Dense Retrieval (DR)	51.36	44.55	35.17	56.42	54.95
DR + raw-ranker	54.76	46.83	39.52	60.00	57.99
DR + graph-ranker	**57.14**	**50.00**	**47.14**	**63.97**	**60.42**

Table 1: Human evaluation on 200 query-candidate pairs. SR and DR indicate sparse and dense retrieval respectively.

Dataset and Experimental Setup: We evaluate our proposed approach on a large real-world dataset, Amazon Shopping Queries [21], prioritizing practical considerations like realistic data scale and manual evaluation where automated ground truth is lacking. This dataset contains challenging search queries paired with products and relevance labels (Exact, Substitute, Complement, Irrelevant). While the dataset is originally for query-to-product relevance, we repurpose it for our entity-to-entity search task by treating products as query entities, and considering all other products as candidates. There are 469,898 unique products under the US locale. For evaluation, we curate a diverse subset of 8.7K query entities sampled across various product super-categories. Each query entity is associated with its corresponding title, description, and other relevant information, from which we extracted pertinent attributes during a preprocessing stage. Certain product categories were excluded from consideration such as adult products (due to ethical concerns); books, media and video games (insufficient products and limited useful context to extract meaningful attributes); and categories with fewer than 500 products (to ensure sufficient data representation).

[4] 0 indicates complete dissimilarity and 100 represents perfect feature matching.

We compare our approach against the following baselines:

(i) **Retriever-only.** We implement two standard retrieval baselines: sparse retrieval (SR) using BM25, and dense retrieval (DR) using BGE-M3 embeddings [7]. Both rank candidates by cosine similarity without LLM capabilities or attribute extraction.

(ii) **DR+raw-ranker.** This baseline represents an LLM ranking approach without structured attributes. We retrieve the top 50 candidates using the same retrieval mechanism as above and then prompt Claude 3.5 Sonnet v2 to rank candidates based on their full product information (unstructured text).

Our proposed method is denoted as **DR+graph-ranker.** It enhances the ranking process by retrieving the top 50 candidates using the same retrieval mechanism, then replacing raw product text with structured attribute data from our product graph. Using an LLM to perform attribute-based comparison and ranking demonstrates the value of our structured attribute representation approach. Given our zero-shot setting, we do not compare against cross-encoders or GNN based methods requiring task-specific training data. Our baseline design targets a controlled comparison: all systems share the same retrieval stage, and the two LLM rankers (raw-ranker vs. graph-ranker) use the same LLM, differing only on structured attribute graph input. Given the absence of comprehensive ground truth labels, we focus on precision-based metrics rather than recall, following established practices for top-k result evaluation [22].

Human Evaluation: We sample 40 diverse query product entities, yielding 200 unique query-candidate pairs per system (40×5 candidates). Three expert annotators assessed all four systems, producing 800 total annotated pairs. Annotators reviewed product information including titles, descriptions, and extracted attributes, providing binary judgments `<SIMILAR / NOT_SIMILAR>` on whether a candidate can serve as a substitute for the query product. We obtained a high pairwise inter-annotator (Cohen's κ) agreement of 0.71.

Table 1 shows that our **DR+graph-ranker** method outperforms all baselines. Notably, precision@5 substantially improves from 39.52% to 47.14%, indicating that our method is particularly effective at retrieving relevant items deeper into the ranked list, thereby providing more comprehensive coverage of correct matches. We also observe consistent improvements in Mean Reciprocal Rank (MRR) from 60.00% to 63.97% and Mean Average Precision (mAP) from 57.99% to 60.42%, suggesting that our structured, graph-based ranking framework yields not only higher-precision top results but also a more robust overall ranking. A qualitative error analysis reveals that our approach correctly matches products on specific attributes. E.g., for a query product entity titled "*kangaroo Home Security System | 5-Piece Kit | Compatible with Alexa and Google Home | App-Based | Pet-Friendly | Reduces Insurance Premium*", our method correctly retrieves competing smart home security kits with matching component count, while the *raw-ranker* tends to surface less relevant smart home devices with differing key attributes.

LLM-based Evaluation: For automatic evaluation, we employ the Nova Pro LLM [2], a different LLM than the ranking model (Claude 3.5 Sonnet v2) to avoid self-evaluation bias. In a preliminary comparison on 500 examples, Claude 3.5 Sonnet v2 achieved 6% higher precision@10 as a ranker but Nova Pro provided more conservative and discriminative relevance judgments. The evaluation LLM assesses query-candidate pairs based on attribute alignment and

Method	nDCG Metrics (%)				Precision-based Metrics (%)					
	nDCG@1	nDCG@3	nDCG@5	nDCG@10	P@1	P@3	P@5	P@10	MRR	mAP
Eval Threshold ≥ 80										
Sparse Retrieval (SR)	58.58	55.80	54.75	56.35	43.86	33.42	27.29	18.71	50.18	42.01
Dense Retrieval (DR)	75.27	75.60	77.30	86.59	49.08	38.71	33.02	25.40	55.57	50.40
DR + raw-ranker	77.66	77.87	79.39	88.00	51.95	41.18	34.89	26.36	57.70	52.78
DR + graph-ranker	**79.96**	**80.02**	**81.47**	**89.86**	**56.85**	**43.49**	**36.71**	**27.28**	**63.13**	**58.02**
Eval Threshold ≥ 50										
Sparse Retrieval (SR)					58.71	49.41	43.05	32.19	66.10	50.63
Dense Retrieval (DR)					72.73	65.54	61.31	54.40	78.59	72.11
DR + raw-ranker					74.78	68.04	63.55	55.90	80.12	74.15
DR + graph-ranker					**78.12**	**69.39**	**64.66**	**56.90**	**83.53**	**77.35**

Table 2: Entity similarity search evaluated using Nova Pro LLM. The evaluation threshold represents the minimum similarity score required for a query-candidate pair to be considered relevant. SR and DR denote sparse and dense retrieval respectively.

semantic similarity under a strict standard. Table 2 shows our system's performance (DR+graph-ranker) at two evaluation thresholds.

At a higher similarity threshold (≥ 80), our method consistently outperforms all baselines. We observe substantial gains over the basic retrieval baseline: i.e., an nDCG@1 increase from 75.27% (DR) to 79.96% and a precision@1 gain from 49.08% to 56.85%. Moreover, the MRR score significantly improves from 55.57% to 63.13%, indicating that our proposed system both retrieves relevant entities and places them higher in the ranked results. While adding the RAW-RANKER component provides incremental improvements over the baseline retrieval, the substantial gap between BASELINE-DR+RAW-RANKER and our attribute-based ranking underscores the significant advantage of explicitly incorporating structured attribute information into ranking. At a more lenient threshold (≥ 50), our proposed method's precision@1 again rises from 74.78% (BASELINE-DR+RAW-RANKER) to 78.12%, and MRR increases from 80.12% to 83.53%. These robust performance improvements across thresholds demonstrate the stable and meaningful benefits of structured attribute reasoning in entity similarity ranking tasks, effectively capturing nuanced similarities even under relaxed evaluation criteria.

We also separately evaluate the attribute extraction accuracy using Claude 3.5 Sonnet v2 as a judge on 20,000 samples, achieving 83.47% F1. Hallucination is minimized through the predefined attribute schemas and explicit extraction instructions that constrain the LLM to find attributes in the provided text.

3.1 Live Production Deployment

A first version of our proposed system is deployed in production at a leading global e-commerce service. The structured and standardized attribute schemas extracted from unstructured product information serve dual purposes: (i) as input to the graph-aware LLM ranking pipeline described above, and (ii) as training data for a high-quality in-house product embedding model. This embedding model captures nuanced product relationships, enabling retrieval of high-quality candidate products similar to the query product, which are then to be re-ranked via a product graph-aware LLM ranker, as described above. Additional business factors such as price and customer ratings can be incorporated as extracted attribute values into the attribute graph, and time-sensitive attributes (e.g., fashion trends) are captured through features like release year or technical specifications that naturally indicate recency. Existing structured data from the product catalog (e.g., brand fields) is directly incorporated into the attribute graph, with LLM extraction used only for attributes not already available in structured form.

Our system serves as input for multiple downstream e-commerce applications, enabling additional business rules (e.g., pricing, availability) to be applied as a subsequent layer. Our embedding-based retrieval stage limits the online graph-aware LLM ranking component to a manageable candidate set (50–80 items). Structured attribute representations significantly reduce the context length fed to the LLM ranker – approximately 300 tokens per product versus 700 for raw unstructured text descriptions, achieving a 57% token reduction. For ranking 50 candidates, this reduces the total input by approximately 20,400 tokens, directly lowering attention complexity and KV cache memory requirements. In practice, this yields approximately 250 ms faster inference per request with Claude 3.5 Sonnet v2 compared to raw-text based ranking. While the online ranking component incurs inference costs relative to retrieval-only approaches, the latency remains feasible for high-value search scenarios such as entity similarity search in specialized product categories where precise attribute matching is critical for better quality outputs, user satisfaction and higher conversion rates. In-depth evaluation revealed metrics and trends matching or surpassing those reported above, with a preliminary human evaluation on 50 samples yielding a precision@3 of 0.76.

4 Conclusion

We present a two-stage system for entity similarity search that decouples LLM-driven attribute graph construction from graph-aware LLM ranking. By reasoning over structured attributes rather than raw text, our approach achieves a 6% improvement in average precision over strong baselines in zero-shot settings, under practical considerations of scalability. Both human and LLM-based evaluations confirm consistent gains across metrics and product categories, with encouraging potential in a live deployment setting.

Presenter Biography

Nikhita Vedula is a Senior Applied Scientist at Amazon. She received her Ph.D. from the Ohio State University. Her research interests and contributions span the fields of natural language processing, conversational AI and information retrieval. She has published in leading peer-reviewed conferences such as SIGIR, WSDM, the Web conference, EMNLP, NAACL and ACL. Her research has been recognized with one Best Paper award, one Best Paper Honorable Mention, and one Outstanding Paper award. She also regularly serves as a Program Committee member and Area Chair of multiple top tier conferences in her field.

References

[1] Mofetoluwa Adeyemi, Akintunde Oladipo, Ronak Pradeep, and Jimmy Lin. 2024. Zero-Shot Cross-Lingual Reranking with Large Language Models for Low-Resource Languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 650–656. doi:10.18653/v1/2024.acl-short.59

[2] Amazon AGI. 2024. The Amazon Nova Family of Models: Technical Report and Model Card. https://www.amazon.science/publications/the-amazon-nova-family-of-modelstechnical-report-and-model-card

[3] Anthropic. 2024. Claude 3.5 Sonnet. https://www.anthropic.com.

[4] Krisztian Balog. 2018. *Entity-oriented search*. Springer Nature.

[5] Ansel Blume, Nasser Zalmout, Heng Ji, and Xian Li. 2023. Generative Models for Product Attribute Extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Mingxuan Wang and Imed Zitouni (Eds.). Association for Computational Linguistics, Singapore, 575–585. doi:10.18653/v1/2023.emnlp-industry.55

[6] Shubham Chatterjee and Laura Dietz. 2021. Entity Retrieval Using Fine-Grained Entity Aspects. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1662–1666. doi:10.1145/3404835.3463035

[7] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 2318–2335. doi:10.18653/v1/2024.findings-acl.137

[8] Zhongfen Deng, Hao Peng, Tao Zhang, Shuaiqi Liu, Wenting Zhao, Yibo Wang, and Philip S. Yu. 2023. JPAVE: A Generation and Classification-based Model for Joint Product Attribute Prediction and Value Extraction. In *2023 IEEE International Conference on Big Data (BigData)*. 1087–1094.

[9] Kaustubh Dhole, Nikhita Vedula, Saar Kuzi, Giuseppe Castellucci, Eugene Agichtein, and Shervin Malmasi. 2025. Generative product recommendations for implicit superlative queries. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*. 77–91.

[10] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM* 51, 12 (Dec. 2008), 68–74. doi:10.1145/1409360.1409378

[11] Jiaying Gong and Hoda Eldardiry. 2024. Multi-Label Zero-Shot Product Attribute-Value Extraction. In *Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 2259–2270. doi:10.1145/3589334.3645649

[12] Parastoo Jafarzadeh, Zahra Amirmahani, and Faezeh Ensan. 2022. Learning to Rank Knowledge Subgraph Nodes for Entity Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2519–2523. doi:10.1145/3477495.3531888

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[14] Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. TXtract: Taxonomy-Aware Knowledge Extraction for Thousands of Product Categories. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 8489–8502. doi:10.18653/v1/2020.acl-main.751

[15] Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9410–9421. doi:10.18653/v1/2023.findings-emnlp.631

[16] Yanzeng Li, Bingcong Xue, Ruoyu Zhang, and Lei Zou. 2023. AtTGen: Attribute Tree Generation for Real-World Attribute Joint Extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 2139–2152. doi:10.18653/v1/2023.acl-long.119

[17] Robyn Loughnane, Jiaxin Liu, Zhilin Chen, Zhiqi Wang, Joseph Giroux, Tianchuan Du, Benjamin Schroeder, and Weiyi Sun. 2024. Explicit Attribute Extraction in e-Commerce Search. In *Proceedings of the Seventh Workshop on e-Commerce and NLP @ LREC-COLING 2024*, Shervin Malmasi, Besnik Fetahu, Nicola Ueffing, Oleg Rokhlenko, Eugene Agichtein, and Ido Guy (Eds.). ELRA and ICCL, Torino, Italia, 125–135. https://aclanthology.org/2024.ecnlp-1.13/

[18] Steffen Metzger, Ralf Schenkel, and Marcin Sydow. 2013. QBEES: query by entity examples. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (San Francisco, California, USA) *(CIKM '13)*. Association for Computing Machinery, New York, NY, USA, 1829–1832. doi:10.1145/2505515.2507873

[19] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2876–2885.

[20] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. arXiv:2309.15088 [cs.IR] https://arxiv.org/abs/2309.15088

[21] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. (2022). arXiv:2206.06588

[22] Tetsuya Sakai and Noriko Kando. 2008. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval* 11, 5 (01 10 2008), 447–470. doi:10.1007/s10791-008-9059-7

[23] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=nnVO1PvbTv

[24] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14918–14937. doi:10.18653/v1/2023.emnlp-main.923

[25] Christophe Van Gysel, Maarten de Rijke, and Marcel Worring. 2016. Unsupervised, Efficient and Semantic Expertise Retrieval. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. 1069–1079.

[26] Ellen Voorhees and D Tice. 2000. The TREC-8 Question Answering Track Evaluation. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=151446

[27] Guipeng Xv, Chen Lin, Wanxian Guan, Jinping Gou, Xubin Li, Hongbo Deng, Jian Xu, and Bo Zheng. 2023. E-commerce Search via Content Collaborative Graph Neural Network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. 2885–2897. https://doi.org/10.1145/3580305.3599320

[28] Li Yang, Qifan Wang, Jianfeng Chi, Jiahao Liu, Jingang Wang, Fuli Feng, Zenglin Xu, Yi Fang, Lifu Huang, and Dongfang Liu. 2024. EAVE: Efficient Product Attribute Value Extraction via Lightweight Sparse-layer Interaction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 1491–1505. doi:10.18653/v1/2024.findings-emnlp.80

[29] Danqing Zhang, Zheng Li, Tianyu Cao, Chen Luo, Tony Wu, Hanqing Lu, Yiwei Song, Bing Yin, Tuo Zhao, and Qiang Yang. 2021. QUEACO: Borrowing Treasures from Weakly-labeled Behavior Data for Query Attribute Value Extraction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. 4362–4372.

[30] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[31] Yilun Zhu, Nikhita Vedula, and Shervin Malmasi. 2025. Hint-Augmented Reranking: Efficient Product Search using LLM-Based Query Decomposition. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*. 200–216.

[32] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2023. Large Language Models for Information Retrieval: A Survey. *CoRR* abs/2308.07107 (2023). arXiv:2308.07107 https://arxiv.org/abs/2308.07107