A Study of Embedding-Based Retrieval Models

Vishravars Ramasubramanian

ABSTRACT

In a large-scale e-commerce platform, retrieving products with high semantic and behavioral relevance is critical for enhancing customer satisfaction. This report examines the recent advancements in Walmart's Embedding-Based Retrieval (EBR) system [1], which addresses challenges such as Noisy engagement signals, Query misspellings, and Static training labels. We focus on four core contributions: a revised labeling framework that uses weighted engagement data, a Relevance Reward Model (RRM) that leverages human-annotated semantic relevance, a typo-aware training strategy that increases robustness to user input errors, and a multi-objective loss function that balances relevance and engagement during optimization. To broaden our perspective, we compare Walmart's two-stage architecture to Facebook's unified multi-task learning approach[2], which jointly learns semantic match and engagement propensity within a single model. We also draw inspiration from other large-scale systems, such as Taobao's reinforcement learning-based retrieval engine, which dynamically learns query-sensitive weights for various engagement signals [3]. Building on these insights, this study proposes several research questions that aim to guide future directions in adaptive labeling, engagement-aware supervision, and unified learning frameworks in neural retrieval systems.

MOTIVATION

Walmart transitioned from a traditional keyword-based search model to an embedding-based retrieval system. The latter, however, is susceptible to noisy data (false positives and false negatives) and incorrect results due to user typos. This paper was selected as the primary study material because it highlights the inherent complexities introduced by embedding-based retrieval systems, and allows us to examine the limitations of proposed solutions—some of which are often underreported.

INTRODUCTION

In large-scale e-commerce platforms, keyword-based retrieval systems remain popular due to their speed and interpretability. However, they often fall short in accurately capturing user intent, especially when search queries involve synonyms, ambiguous terms, or typographical errors. To address these limitations, embedding-based retrieval (EBR) systems have emerged as a more semantically robust alternative. These systems represent both queries and products as dense vectors and retrieve items based on semantic similarity in the embedding space.

A common architecture in EBR is the dual-encoder model, where queries and products are encoded independently, and their relevance is measured via vector similarity. While efficient and scalable, dual encoders may struggle with nuanced query-product interactions. In contrast, cross-encoders jointly process query-product pairs, achieving higher precision by leveraging fine-grained token-level interactions. However, due to their computational overhead, cross-encoders are typically unsuitable for large-scale e-commerce retrieval, where performance and latency are critical concerns.

While embedding-based retrieval (EBR) offers clear improvements over traditional keyword search, it also introduces new challenges. These models are typically trained on user interaction signals such as clicks or purchases which may not reliably indicate semantic relevance. For instance, a user might click on a product due to an attractive image or a discounted price, even if the item is unrelated to their original query. Moreover, query misspellings, which are prevalent in real-world traffic, can further degrade retrieval quality if left unaddressed. The following issues have been identified as key sources of relevance degradation:

- Label noise in training data derived from click and purchase logs, which do not always reflect true semantic relevance.
- False negatives introduced during negative sampling, leading to undertrained semantic boundaries.
- Query misspellings, common in real-world search traffic, which were inadequately handled by earlier models.

This report focuses on enhancements from Walmart's system that aim to improve retrieval quality:

(1) A *revised labeling framework* that assigns improved engagement-based weights to products based on impressions, clicks, add-to-carts, and orders.
(2) A *Relevance Reward Model (RRM)* that leverages human-labeled judgments to refine product-query matching and reduce false positives.
(3) A *typo-aware training strategy* that injects realistic spelling errors during model training, increasing the system's resilience to noisy input.
(4) A *multi-objective loss function* that combines engagement signals with relevance scores derived from RRM.

TECHNIQUES

Revised labeling framework

In machine learning based retrieval systems, a label represents the ground truth signal used to train models to rank or retrieve relevant items. In embedding-based retrieval (EBR), a label quantifies how relevant a product is to a given query. To better reflect the true relevance of query product pairs, Walmart's EBR system introduces a revised labeling framework that assigns continuous-valued labels based on weighted user engagement signals. This is a improvement over the earlier step-function-based labeling scheme [4], which failed to meaningfully distinguish between products with high and low engagement—particularly when multiple types of signals (e.g., clicks vs. purchases) were present at similar frequencies.

The new label for a query–product pair is computed as:

$$S_{ij} = w_i \cdot \text{Impressions} + w_c \cdot \text{Clicks} + w_a \cdot \text{AddToCarts} + \text{Orders}$$

Where ($w_i = 0.001$, $w_c = 0.01$, $w_a = 0.1$, and orders are assigned a full weight of 1.

This allows the model to treat purchases as the strongest engagement signal while still accounting for softer signals like impressions and clicks. For example:

A product with 1 order, 2 add-to-carts, 10 clicks, and 200 impressions will receive a label:

$$S_{ij} = 0.001 \cdot 200 + 0.01 \cdot 10 + 0.1 \cdot 2 + 1 \cdot 1 = 0.2 + 0.1 + 0.2 + 1 = 1.5$$

A highly clicked product with no orders, say 0 orders, 1 add-to-cart, 50 clicks, and 1000 impressions, will receive:

$$S_{ij} = 0.001 \cdot 1000 + 0.01 \cdot 50 + 0.1 \cdot 1 + 0 = 1.0 + 0.5 + 0.1 = 1.6$$

Although both scores are close, the model now has the intent to distinguish which signals matter more through learning on the weighted targets through label revision using the RRM. This continuous labeling strategy helps mitigate issues such as products receiving high impressions but low conversion, or false positives arising from clickbait-like data.

Relevance Reward Model

The user engagements can be used to label relevance in retrieval systems but they can introduce noise. Its likely that a product may be clicked unrelated to the intent of purchase and this could lead to false positives in the training. And similarly, geneuinely relevant products may have been overlooked by not clicking and this can lead to false negatives. The Relevance Reward Model integrates the human annotated relevance to the training set. The core idea is to predict how semantically relavant a product is to a query. The idea is to collect dataset based on human raters for queries and product match on 3 point scale:

(1) Exact match
(2) Substitute
(3) Irrelevant

The BERT based cross-encoding is used to train this data and classify new query-product pairs into one of these categories. The idea of using cross-encoding is to bring in precision. Once the RRM is trained, the model is used in two ways:

(1) Label revision (adjusting the engagement labels) In this method, the model is applied on query-product pairs with high engagement and if its found to be symantically irrelevant, its relevance is downgraded. For example, a product that is purchased in a different context would not be treated as a strong example.
(2) Relevance label (create new relevance based on human judgement) In this method, instead of treating as descrete categories in Label revision, the trained model maps them into continuous scores.
    - Exact match: score in the range 0.4 to 1
    - Substitute: 0.05 to 0.5
    - Irrelevant: 0 to 0.05

Walmart ultimately chooses the second approach using RRM-based relevance labels because it affects every training instance directly through the loss function, not just a subset of high-engagement examples.

Typo-aware training

As per the study, roughly 13% of the customer queries on e-commerce platform are likely to contain typos and a robust system should tackle this noise. Since the Embedding based retrieval systems are trained primarily on clean query data, they underperform on noisy real-world inputs. To address this, Walmart introduced typo aware training strategy that injects synthetic spelling errors into their training set. This makes the model resilient to user errors. The strategy uses TextAttack, and OSS Python library to generate misspellings with following transformations:

- Deletion: Remove a character (airtag to aitag)
- Insertion: Add an extra character (airtag to airttag)
- Swap: Switch adjacent characters (airtag to airtga)
- Substitution: Replace a character with another (airtag to aisrtag)
- Keyboard error: Replace a character with a nearby key on the keyboard

The queries were only modified and the product-label remain unchanged. This helps the model to learn and map newer spelling combination to the correct product embeddings. Additionally, to maintain the symantic integrity, numerical tokens were not touched as they carry typical product details. The paper reports an improvement in retrieval quality and revenue gain.

Loss Function

To balance user behavior and semantic relevance, Walmart introduces a multi-objective loss function. The key idea is to combine two complementary training signals:

Engagement-based loss — derived from user behavior signals such as clicks, add-to-carts (ATCs), and purchases. This signal reflects what users did, which is useful but often noisy or biased.

Relevance-based loss — derived from the continuous relevance scores predicted by the Relevance Reward Model (RRM). This signal reflects what users should have done according to human judgment, helping to correct noisy engagement data.

The combined loss function is:

$$\text{loss} = \omega \cdot \text{engagement\_loss} + (1 - \omega) \cdot \text{relevance\_loss}$$

Here, ω is a weighting parameter that controls the the two objectives. Walmart sets $\omega = 0.5$, giving equal importance to both user behavior and semantic relevance during training. Consider a query "wireless charger adapter" and two products:

- **Product A**: Highly clicked and added to cart, but rated as **substitute** by human raters.
- **Product B**: Not clicked often, but judged as **exact match** by human raters. Assume:
- Engagement label for Product A: 1.2 (from clicks, ATCs, etc.)
- Relevance label for Product A: 0.3 (substitute)
- Engagement label for Product B: 0.3
- Relevance label for Product B: 0.9 (exact match)

Without the relevance loss, the model would learn to rank Product A higher because of stronger engagement. However, the **combined loss** encourages the model to consider both signals. By averaging them, Product B gains more weight due to its higher semantic relevance.This design allows Walmart to:

- **Mitigate noisy engagement data**, which may be influenced by visual appeal, price, or bias in impressions.
- **Improve semantic matching**, so that products aligned with user **intent** (not just behavior) are retrieved.

Facebook's joint learning of Relevance and Engagement

Facebook's Que2Engage system [2] proposes a unified approach to retrieval by jointly modeling semantic relevance and user engagement using multi-task learning. Rather than relying on offline label corrections, as in Walmart's Relevance Reward Model (RRM), Que2Engage trains a shared encoder architecture to optimize two complementary losses:

Contrastive Loss: This loss function is used to teach the model which query–product pairs are semantically relevant. During training, the model is given pairs of queries and products that users clicked on (positive examples), and pairs that are unrelated or randomly sampled (negative examples). The contrastive loss encourages the model to bring the embeddings of clicked (relevant) pairs closer together in the vector space, while pushing apart the embeddings of unrelated pairs. This helps the model capture deep semantic relationships—such as recognizing that "wireless earphones" and "Bluetooth earbuds" refer to the same type of product—even if the wording differs.

Binary Cross-Entropy (BCE) Loss on Hard Negatives: This loss is applied to products that were shown to users in search results but were not clicked. These products are often more challenging for the model because they might be semantically relevant (e.g., they match the query text), but the user chose not to engage with them. Reasons for non-engagement can include price, brand, appearance, or lack of trust. The BCE loss treats these hard negatives as examples the model should learn to avoid retrieving. It teaches the system to predict lower relevance scores for items that are textually plausible but behaviorally unappealing, improving alignment with real-world user preferences. The combined loss function is:

$$\mathcal{L}_{\text{joint}} = \lambda_1 \cdot \mathcal{L}_{\text{contrastive}} + \lambda_2 \cdot \mathcal{L}_{\text{engagement}}$$

This setup enables real-time learning of both semantic match and behavioral signals within a single, end-to-end training loop. It also simplifies deployment by avoiding task-specific encoders. Facebook's design highlights an important observation: semantic match alone does not guarantee user engagement. A product may be textually relevant but ignored due to factors such as price, brand, or image presentation.

Taobao's Multi-Objective Personalized Product Retrieval

While effective, Walmart's revised labeling framework uses fixed global weights for all engagement types across queries. However, user intent varies significantly depending on the nature of the query—for instance, informational queries like "wireless charger" versus transactional ones like "buy iPhone 14 Pro Max". This uniform weighting may fail to capture the intent-specific importance of signals such as impressions or purchases.

In contrast, the Multi-Objective Personalized Product Retrieval (MOPPR) system proposed by Zheng et al. [3] addresses this limitation by dynamically learning the importance of multiple engagement objectives—relevance, exposure, clicks, and purchases. MOPPR formulates the retrieval task as a reinforcement learning (RL) problem, where a reward signal is computed based on downstream business metrics, and the agent learns to adjust the contribution of each objective through a policy. Specifically, the model learns personalized reward weights that vary across user and query contexts. These weights are applied in a ranking policy that selects products not just based on semantic similarity, but also on expected long-term value (e.g., likelihood of purchase or customer satisfaction).

This demonstrates that dynamic, context-aware weighting of engagement signals is both feasible and beneficial at industrial scale. It suggests that adopting a query-sensitive or learned weighting strategy could further enhance the fidelity of label assignment in training retrieval models.

One possible improvement is to use query-dependent weighting functions, where each interaction signal is modulated by a learned function of the query:

Walmart's revised labeling framework uses a fixed-weight linear formula to compute the engagement score for a query–product pair:

$$S_{ij} = w_i \cdot \text{impressions} + w_c \cdot \text{clicks} + w_a \cdot \text{ATC} + \text{orders}$$

Here, the weights $w_i = 0.001$, $w_c = 0.01$, and $w_a = 0.1$ are chosen heuristically and remain constant across all queries. While simple and effective, this approach does not account for variations in user intent or query type. For example, in transactional queries, orders may be more important than impressions, but the model applies the same weights regardless.

In contrast, Taobao's MOPPR system learns to assign dynamic, context-aware weights to each engagement signal using reinforcement learning. Instead of applying fixed coefficients, MOPPR formulates retrieval as a multi-objective optimization problem, where the importance of objectives like relevance, exposure, clicks, and purchases is learned through reward feedback. The scoring becomes:

$$S_{ij} = \phi_i(q) \cdot \text{impressions} + \phi_c(q) \cdot \text{clicks} + \phi_a(q) \cdot \text{ATC} + \phi_o(q) \cdot \text{orders}$$

Where each $\phi_k(q)$ is a query-aware function (e.g., attention or policy output) that adapts the weight of signal k based on the nature of the query or context. In Summary:

- **Walmart**: Fixed weights → simple, but static and not intent-aware.
- **MOPPR**: Learned weights → dynamic and tailored to each query, capturing the true business value of different user signals.

RESEARCH QUESTIONS AND EXPERIMENT

RQ1: Can Walmart's retrieval system improve by jointly training for relevance and engagement, like Facebook's Que2Engage?

Walmart currently trains its model in two parts: it uses human-labeled relevance data (via the Relevance Reward Model, or RRM)

offline, and separately trains on engagement signals (e.g., clicks, orders) using softmax loss.

In contrast, Facebook's Que2Engage uses a multitask learning setup that optimizes for both relevance and engagement at the same time. This question explores whether Walmart can benefit from combining these signals into a single unified training objective to improve both semantic retrieval and downstream engagement.

RQ2: Can adaptive engagement weighting, as used in Taobao's MOPPR, outperform Walmart's fixed-weight strategy for better intent-aware retrieval?

Walmart currently uses fixed weights to combine engagement signals—impressions, clicks, add-to-carts, and orders—into a single label (e.g., 0.001, 0.01, 0.1, and 1 respectively). Taobao's MOPPR system instead learns dynamic, query-dependent weights to better reflect user intent in varying contexts.

This question investigates whether replacing Walmart's static weighting with a learned, context-aware strategy can produce more accurate training labels and improve retrieval quality, especially on intent-heavy queries.

EXPERIMENT SETUP

Variant Design (Grouped by Research Question)

- **For RQ1 (Joint Training)**:
  (1) **Two-stage baseline**: Walmart's current setup using separate training for relevance (via offline RRM) and engagement (via fixed softmax loss).
  (2) **Multitask model**: A joint training setup using shared dual encoders with multitask loss i.e contrastive loss for relevance and BCE loss for engagement.
- **For RQ2 (Adaptive Weighting)**:
  (1) **Fixed weights (baseline)**: The model uses static weights (e.g., impressions = 0.001, clicks = 0.01, etc.) to compute training labels.
  (2) **Adaptive weights**: The same architecture as above, but replaces static weights with learned query-conditioned functions $\phi_k(q)$, assigning dynamic weights per engagement signal based on query features.

Datasets

We use Walmart's large-scale dataset containing ~780 million query–product pairs with engagement signals (impressions, clicks, add-to-carts, orders) collected over one year.

For evaluation, we use three test sets defined in the Walmart paper:

- **Small index**: 122k human-labeled queries evaluated against 3M products. Used to test exact match relevance in a controlled setting.
- **Big index**: 1k high-traffic queries tested against 180M products. Measures real-world retrieval quality and scalability.
- **Purchased dataset**: 800k queries with actual purchases. Used to evaluate how well the model retrieves items aligned with buying behavior.

These datasets help assess both relevance-based accuracy and business impact across different conditions.

Metrics

- **For RQ1: Can training the model on both relevance and engagement improve performance?**
  - **Match with purchased products**: This tells us whether the model retrieves items that users are most likely to buy. It's a key sign of strong engagement performance.
  - **Overall user behavior**: If the model is tested online, we will measure how it affects actions like clicks, add-to-carts, and completed purchases. This shows whether it drives better user interaction.
- **For RQ2: Can using adaptive (learned) weights for engagement signals perform better than fixed weights?**
  - **Improvement in less common queries**: We will test if the model with adaptive weights performs better on queries that are rare or harder to understand. This would show that it adapts well to different user needs.

Expected Outcomes

- **For RQ1**:
  - If the new joint training model performs better than the current two-part system in retrieving purchased products and improving user actions (like clicks and conversions), it would support the idea that combining both training signals is beneficial.
  - If the model continues to perform reasonably well on relevance judged by humans, it would confirm that semantic quality is preserved even when optimizing for engagement.
- **For RQ2**:
  - If the adaptive weighting model performs better on rare or intent-specific queries compared to fixed weights, it would show that learning from context improves label quality.

REFERENCES

[1] Zhou, K., Modi, C., Sun, W., Uppaluru, A., Siddiqui, T., Maheshwari, P., Subhash, R., Kapoor, A., & Yan, Y. (2024). Enhancing Relevance of Embedding-based Retrieval at Walmart. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24). Association for Computing Machinery, New York, NY, USA.

[2] He, Y., Tian, Y., Wang, M., Chen, F., Yu, L., Tang, M., Chen, C., Zhang, N., Kuang, B., & Prakash, A. (2023). Que2Engage: Embedding-based Retrieval for Relevant and Engaging Products at Facebook Marketplace.

[3] Zheng, Y., Bian, J., Meng, G., Zhang, C., Wang, H., Zhang, Z., Li, S., Zhuang, T., Liu, Q., & Zeng, X. (2022). Multi-Objective Personalized Product Retrieval in Taobao Search. arXiv preprint arXiv:2210.04170.

[4] Alessandro Magnani, Feng Liu, Suthee Chaidaroon, Sachin Yadav, Praveen, Reddy Suram, Ajit Puthenputhussery, Sijie Chen, Min Xie, Anirudh Kashi, Tony Lee, et al. 2022. Semantic retrieval at walmart. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 3495–3503.