LLMDistill4Ads: Using Cross-Encoders to Distill from LLM Signals for Advertiser Keyphrase Recommendations at eBay

Soumik Dey
eBay Inc.
San Jose, CA, USA
sodey@ebay.com

Benjamin Braun
eBay Inc.
Amsterdam, Netherlands
bbraun@ebay.com

Naveen Ravipati
eBay Inc.
San Jose, CA, USA
navravipati@ebay.com

Hansi Wu
eBay Inc.
San Jose, CA, USA
hanswu@ebay.com

Binbin Li
eBay Inc.
San Jose, CA, USA
binbli@ebay.com

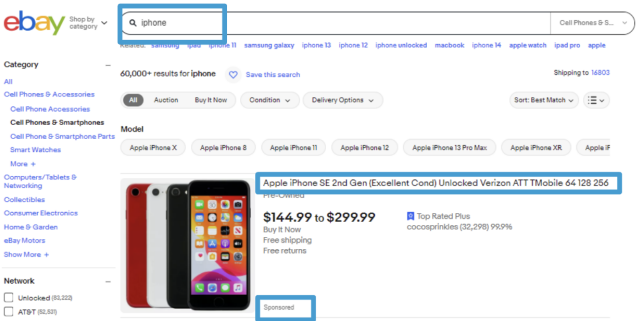
Abstract

Sellers at eBay are recommended keyphrases to bid on to enhance the performance of their advertising campaigns. The relevance of these keyphrases is crucial in avoiding the overcrowding of search systems with irrelevant items and maintaining a positive seller perception. It is essential that keyphrase recommendations align with both seller and Search judgments regarding auctions. Due to the difficulty in procuring negative human judgment at scale, employing LLM-as-a-judge to mimic seller judgment has been established as the norm in several studies. This study introduces a novel two-step LLM distillation process from a LLM-judge used to debias our Embedding Based Retrieval (EBR) model from the various biases that exist in click-data. We distill from an LLM teacher via a cross-encoder assistant into a bi-encoder student using a multi-task training approach, ultimately employing the student bi-encoder to retrieve relevant advertiser keyphrases. We show that integrating a knowledge distillation process from LLMs in a multi-task training setup enhances bi-encoder performance in retrieving relevant advertiser keyphrases at eBay.

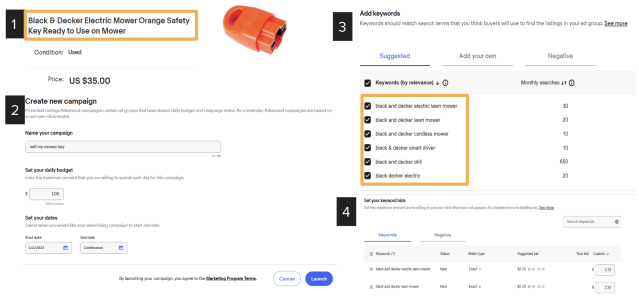
ACM Reference Format:
Soumik Dey, Benjamin Braun, Naveen Ravipati, Hansi Wu, and Binbin Li. 2025. LLMDistill4Ads: Using Cross-Encoders to Distill from LLM Signals for Advertiser Keyphrase Recommendations at eBay. In . ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Within the domain of e-commerce, sellers strategically leverage online advertising methodologies, such as keyphrase recommendations [1, 34–36, 62], to mitigate their typically inferior positions within organic search outcomes. This tactic allows them to establish a strategic presence on the search results page (SRP) and enhance interaction with prospective buyers, see Figure 1. The relevance of advertiser keyphrases is critical, as it influences seller perspectives and assists in avoiding the saturation of search systems with numerous non-relevant items competing for prominence in auctions. Models that assess advertiser keyphrase relevance are trained to identify general patterns in click and sales data. A keyphrase generating a substantial number of clicks or sales for an item suggests its relevance to the item. Essentially, clicks and sales serve as strong positive relevance indicators; however, they are inadequate for indicating non-relevance. E-commerce datasets suffer from missing-not-at-random (MNAR) conditions, due to a variety of biases [4, 7, 20, 21, 27, 46, 53, 60]. The lack of clicks for an item relative to a specific query does not automatically imply irrelevance. In the e-commerce context, buyers act as annotators, yet, unlike conventional annotators, they encounter a biased item presentation



(a) Buyer side



(b) Seller Side

Figure 1: Screenshot of our keyphrases for manual targeting in Promoted Listings Priority for eBay Advertising.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

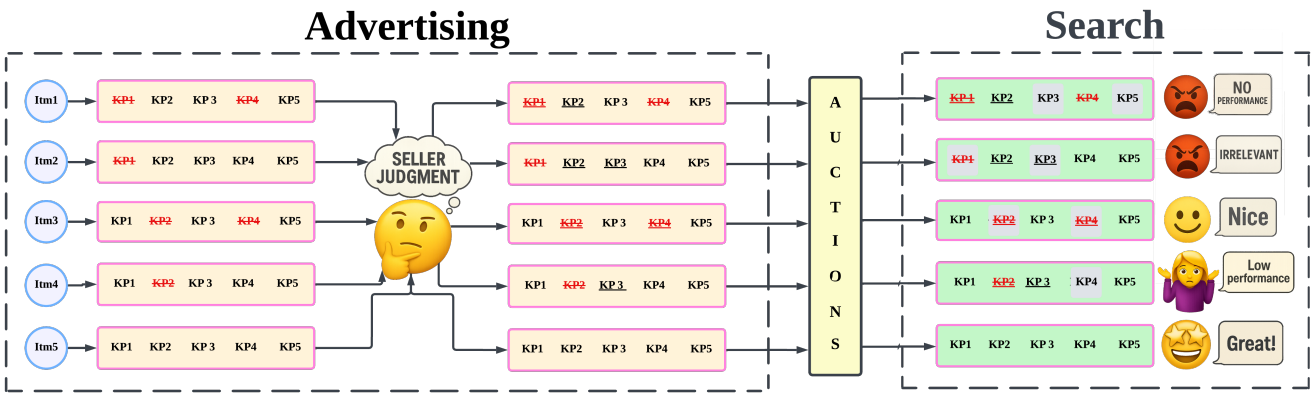


Figure 2: Auction mechanism of items (Itm) in relation to keyphrases (KP). Red strikethrough font represents filter of Advertising, the underline represents seller curation of keyphrases after advertising has filtered them while gray highlight represents the relevance filter of Search.

due to search rankings, which influences their selection in terms of clicks or sales. Consequently, an item with lower popularity will appear in a less favorable position on the SRP because search rankings are based on clicks, and consequently it may not receive any clicks or sales. This bias depreciates the dependability of negative relevance cues based solely on clicks or sales.

From the perspective of sellers, *eBay Advertising* promotes keyphrases to which they can bid on for their items. The items are then added to an auction matched to the keyphrase (query on the Search side). This auction process involves intricate interactions with *eBay Search*, which acts as an intermediary, aligning items with the advertised keyphrases proposed by eBay Advertising and applying a relevance filter to the auction items. Consequently, the logged click data encompasses only those keyphrases that pass the relevance filter (auction-winning, impression-gaining keyphrases are documented). *As a result, this introduces an additional bias, as training data contains only keyphrases approved by the Search relevance filter.* Training with this click data means the model is not exposed to keyphrases deemed irrelevant by Search, although Advertising does produce such keyphrases that need screening. This *middleman bias* constitutes a type of sample selection bias [12, 53], which further affects the click data within the domain of advertiser keyphrase recommendation.

The dynamics among Sellers, Advertising, and Search reflect an inherent asymmetry: Sellers have ultimate authority over adopting Advertising's keyphrases, whereas Search possesses the final discretion to dismiss these keyphrases during auctions. For example, consider `Itm1` from Figure 2; while eBay Advertising finds KP1 and KP4 to be irrelevant, *eBay Search* deems KP2, KP3, and KP5 irrelevant. Furthermore, the seller dismisses KP2 as a valid keyphrase, consequently preventing `Itm1` from participating in any auctions. *This scenario unfolds irrespective of the keyphrases' actual relevance; thus, even if* KP2, KP3, *and* KP5 *are truly relevant,* `Itm1` *remains excluded from those auctions due to the seller and eBay Search deeming them irrelevant.* Perfect concordance is exemplified by `Itm5` and, to a lesser extent, `Itm3`, where all parties achieve consistent agreement. In the case of `Itm3`, the situation is not as favorable as for `Itm5` since some keyphrases still do not enter auctions. Hence, an efficient retrieval model would not retrieve keyphrases KP2 and KP4 for `Itm3`, aligning with seller judgment, advertising judgment, and search judgment.

Understanding auction dynamics and mechanisms is crucial for effective advertising campaigns. Our product, Promoted Listings Priority, allows manual targeting where sellers can choose keyphrases or opt for our suggestions. For example, if sellers deem KP2 for `Itm1` irrelevant, they can ignore it, nullifying our advice. This often leads to *"irrelevant"* keyphrases overwhelming sellers, lowering Seller Satisfaction and wasting resources for eBay. Offering seemingly sensible but ineffective keyphrases can reduce campaign efficiency and cause sellers to exit. Thus, it is essential that keyphrases align with human judgment and maintain strong performance. The relevance of Advertiser keyphrases can be viewed as a complex interaction among three systems: sellers' judgment affecting adoption, Advertising providing bids and keyphrases, and Search managing keyphrase auctions as documented in previous work [10]. On the side of retrieval which is an upstream task, for efficiency purposes, the retrieved keyphrases must maximize all three relevance judgments while also driving revenue, sales and campaign performance for our sellers.

2 Related Work

As is common in the e-commerce domain, semantic search applications are often broken down into a three-stage process: candidates generation, relevance filtering and ranking. Recommending keyphrases for sellers to place bids on also follows this pattern. The candidates generation step consists in combining several recall models.[1] After this union of recalls, a relevance filtering step [11] is performed, followed by a ranker.

The advertiser keyphrase recommendations at eBay are generated using a zoo of models which include *fast-Text* (which creates word embeddings using the CBOW model and employs a straightforward linear neural network model with hierarchical softmax to improve the efficiency of training and inference processes), *Graphite*

[1]The terms *recall model* and *candidates generator model* are used interchangeably.

[34] (which uses bipartite graphs to map words/tokens to the data points and then map them to the labels associated with the data points), *SL-emb* (which uses embeddings of the item's title to find similar listings and then recommends the related queries), or *Rules-based heuristic models* (which use simple techniques of storing item-keyphrase associations based on their co-occurences in the search logs). Within this variety of recall solutions, Embedding-Based Retrieval (EBR) is a two-step solution which first embeds the buyers keyphrases and the item titles in a common vector space, and then recommends the k keyphrases closest to each item title in the vector space via an *Approximate Nearest Neighbor* search.

The two main architectures for encoders within the semantic search domain are cross-encoders and bi-encoders. Bi-encoders "encode" the item and the query independently (self-attention) and present them as vectors for downstream ANN or kNN retrieval. Whereas, cross-encoders can encode both the item and the query jointly (cross-attention), at the cost of an unfavorable inference contract at the item-query combination level, which makes it computationally expensive. Both bi-encoders and cross-encoders are tuned on some supervised signal, but since cross-encoders jointly process the query and the document, they are able to learn more intricate relationships between the query and the document in order to model the supervised signal, in comparison to bi-encoders. Bi-encoders remain the architecture of choice for most EBR, allowing for precomputation of items and query embeddings independently.

When fine-tuning bi-encoders, different labeling strategies can be employed to feed the model with relevant and irrelevant query/item pairs. As exposed in [4], training a relevance model solely on click-based signals is problematic as it is prone to reproducing the popularity and exposure bias present in the training data. However, in our case, we still want to keep the reliably positive pairs (query, item) that come from a training set labeled with CTR, while knowing that the negative labels from this dataset are not reliable indicators of irrelevance [28]. There are several ways of generating negative labels from only positive data namely ranging from trivial solutions like in-batch random negative sampling (IRNS) [18] to more complicated techniques like ANCE [58], NGame [5], GISTEmbed [45] which introduce additional complexity while still reaching a ceiling of performance in cases where reliable negative examples can be procured.

Since click data also suffers from middleman bias, i.e. a form of sample selection bias [53] due to the auction procedure and contract between eBay Search and eBay Advertising, [11] explores training on Search relevance signals to train a relevance filter. This signal gives validation on our keyphrases recommendations rather than buyer queries (which are only matched to items if Search deems them to be relevant); this has been shown to be a signal superior to clicks for our use case [11]. On the same note, since our advertisers also perform a manual check of our keyphrase offering, aligning with human judgement is also critical to the adoption of our keyphrases. As a proxy for human judgement, using LLMs to generate relevance labels offers a less biased alternative, especially as they arrive pre-loaded with a broad spectrum of world knowledge, eliminating the need for domain-specific pre-training or fine-tuning [26] for our massive inventory of diverse items (2.3 billion items over 100,000 categories). For our encoders to learn from the diversity of the above-mentioned labels, multi-task training can be employed and has seen much success in this domain. Piccolo2 [17] employed InfoNCELoss [52], CoSENTLoss, and a variation on InfoNCELoss without in-batch negatives and only hard negatives selected using BM25 [42] to reach state-of-the-art performance on Chinese.

Recent investigations, such as [13, 14, 32, 50, 54], examine the role of Large Language Models (LLMs) in generating labels to enrich search data and enhance retrieval. This method underscores the scalability of LLM-produced labels in minimizing the high costs of manual annotation. Beginning with a limited human-annotated dataset, researchers refined an LLM to produce an amplified set of labels to train a more compact model. LLMs, while gaining traction as assessment and data augmentation instruments, encounter critique about their suitability for evaluation and data generation, as highlighted in [2, 44, 47]. Recent findings in [10] reveal that labels generated by general-purpose LLMs optimally can be used to fine-tune cross-encoder models for assessing keyphrase relevance better than those derived from search logs or fine-tuned LLMs. It strongly advocates for the adoption of business-oriented metrics for relevance model evaluation, providing more actionable insights into model efficiency and business outcomes.

To effectively distill knowledge from LLM signals, training on soft outputs would be beneficial compared to training on the hard outputs from the LLM. Cross encoders are known to learn binary labels with much better accuracy. TwinBERT [30] and PROD [29] introduce the idea of distilling cross-encoders into a twin tower BERT model. TwinBERT processes input embeddings independently and then applies a crossing layer to determine the final score, using either a residual network or simple cosine similarity. PROD outlines a framework for progressive teacher and data distillation, where both model and data complexity are incrementally reduced. Similarly, ERNIE-search [31] adopts a method that uses a Teacher-Assistant [33] framework to distill from a cross-encoder (CE) to a late-interaction model like ColBERT [23, 43], and eventually to a bi-encoder (BE). Following ERNIE-Search, distilling knowledge from a LLM teacher to bi-encoder student using a cross-encoder assistant is a natural follow-up.

Although data augmentation/knowledge distillation from a cross-encoder to bi-encoder is nothing new [49] it has faced shortcomings — CUPID [3] states that the traditional pointwise MSE loss [24] for distillation does not work on cross-encoder outputs for distillation to bi-encoder. D2LLM [26] explores distilling knowledge from a LLM cross encoder into a bi-encoder (augmented with an Interaction Emulation Module) by using a multi-task training scheme that includes a Pearson-based rank imitation loss — which counters the claim made by CUPID [3] as to the distillation of cross-encoders to bi-encoders using simple pointwise losses. In summary, our paper explores navigating the various biases presented in click-data in the context of advertiser keyphrase recommendations and explores training on disparate signals from Search relevance scores and LLM labels. This framework of multi-task learning also is supplemented by a Teacher-Assistant framework using an LLM teacher, cross-encoder assitant and a bi-encoder student. Our paper also covers a multitude of ablation studies exploring the various loss function and labels in the multi-task framework and the various loss functions in the context of knowledge distillation from cross-encoders to bi-encoders.

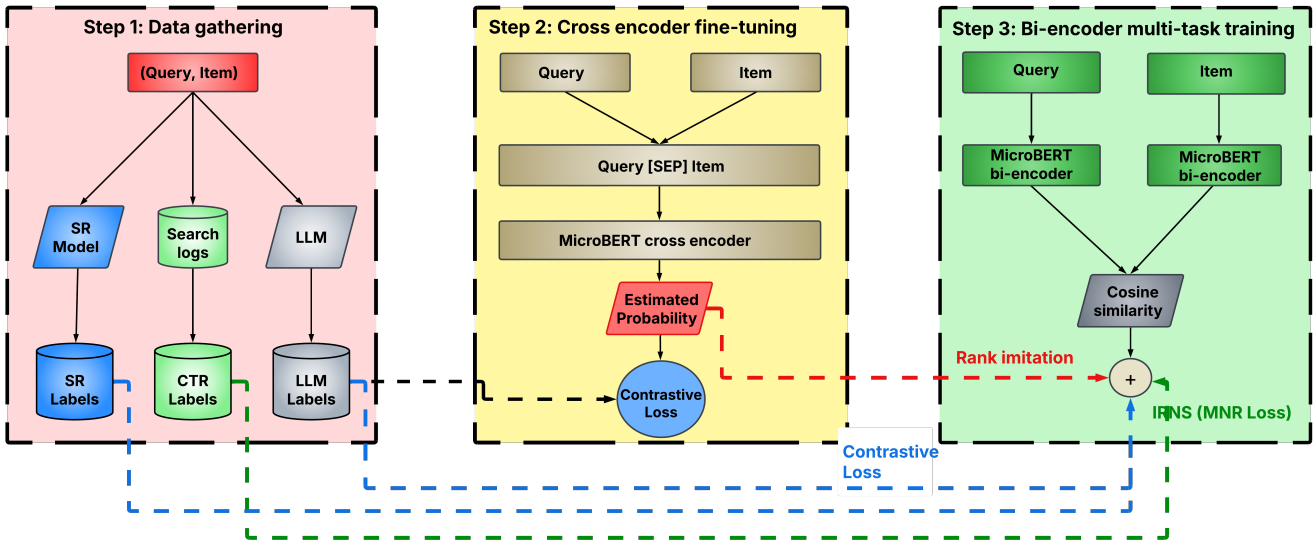


Figure 3: Our proposed architecture for multi-task knowledge distillation. The LLM is distilled to a cross-encoder, which is in turn distilled to the bi-encoder via multi-task hybrid training

3 Embedding Based Retrieval

A dual-tower architecture that independently processes keyphrases and items serves as an excellent framework for a recall model, especially when the focus is on delivering budget-conscious recommendations with acceptable latency. To mitigate biases present in click-based ground truth data, we supplement our dataset with additional signals derived from Large Language Models (LLMs) and search relevance scores, employing a hybrid training approach as illustrated in Figure 3. Within this framework, a cross-encoder is utilized to refine insights from the LLM-generated labels. Our discourse will commence with an outline of our cross-encoder architecture before delving into the bi-encoder design. Following the detailed discussion of these components, we shall elaborate on the multi-task training procedure that we have previously mentioned. Prior to engaging with these sections, we will provide a comprehensive overview of the methods employed in compiling our keyphrase-item dataset, establishing the foundation for the subsequent analysis.

3.1 Dataset curation

In this study, we systematically collected a range of data points encompassing user-query interaction records (referred to as click-data), Search Relevance (SR) metrics, and relevance scores derived from Large Language Models (LLM), all based on item-keyphrase pairs. Notably, click-data arises from item-queries (keyphrases) that successfully pass through a search relevance filter, commonly influenced by intermediary biases, and is inherently subject to biases introduced by the ranking mechanisms on the search side, known as sample selection bias. Conversely, both the SR and LLM datasets are item-keyphrase pairs that are generated through our recommendations, thereby inherently free from any form of sample selection or middleman biases.

3.1.1 CTR-based labels. As mentioned above, we collected the CTR scores for a (query, item) pair as the ratio of the total number of clicks over the total number of impressions on search activity logs from the past 30 days. Given a CTR ratio, the corresponding (query, item) pair was labeled as positive if the ratio if above the threshold of 0.05. As explained in [34], a low query-item CTR does not necessarily mean that the item is irrelevant for the query. Therefore, when using the CTR-based labels, *the positive labels are reliable but the negatives are not.* The minimum thresholds of CTR, impressions and clicks are used to eliminate noise such as 1 click 1 impression or 1 click and 1000 impression. An important thing to note is that due to the nature of the auction process, all click data is deemed relevant by eBay Search (i.e. they are only surfaced to buyers because it is deemed relevant by eBay Search). The dataset size for the click-based labels is 10,702,747 records.

3.1.2 Search Relevance labels. We also gathered relevance scores from eBay Search annotated during the auction process for (item, keyphrase) pairs from our keyphrase offering. These scores are the output of our Search Relevance (SR) model which takes buyer context into account, the scores reflect an average score distribution within a time period. Using these scores, we label a (query,item) pair as positive if its SR score is above a certain threshold (determined by business metrics for our promoted placement and varies across different countries), and negative otherwise. The training dataset comprises of 18,721,682 records.

3.1.3 LLM labels. We generated relevance judgements using Mixtral 8X7B [19] for each (item, keyphrase) pair.[2] Mixtral 8X7B demonstrates a 90% concordance with click data, which serves as an indicator of positive human judgments, with a fair level of agreement

[2]Other "open-source" models such as LLAMA 2 [51], DBRX [41], and Qwen-2 [59] were considered but faced distillation and licensing constraints for commercial use. For commercial models like GPT-4 [37, 38] usage is often hindered by rate limits or API call restrictions to external sources.

with independently collected human judgment data — see [10]. The training set is identical for the SR labels while the test set is 3,524,414 records. The prompt is illustrated below:

Prompt Design

```
Below is an instruction that describes a task. Write a
response that appropriately completes the request.

### Instruction:
Given an item with title: "{title}", determine whether the
keyphrase: "{keyphrase}", is relevant for cpc targeting or
not by giving ONLY yes or no answer:

### Response:
```

3.2 The cross-encoder

We first formulate the use of the cross-encoder in the context of keyphrase recommendations, and then introduce details about our training setup and our choice of architecture.

3.2.1 Problem formulation. The cross-encoder is a transformer model that jointly handles pairs of input sentences by applying cross-attention mechanisms. This is achieved by concatenating the sentence pair into a single sequence, which is subsequently fed into the transformer architecture. The model then produces an integrated representation for this concatenated sequence. Following this, a classification layer makes the final prediction regarding the label. For the cross-encoder inputs, one input is the user-provided keyphrase, while the second consists of both the item title and its corresponding category, combined together. Consequently, the comprehensive input delivered to the cross-encoder is structured as `query [SEP] category name [SEP] item title`.

3.2.2 Training. The base model we used for the cross-encoder is the microBERT model, a distilled version of eBERT [6] (pre-trained on a dataset that includes eBay item title corpus) with the architecture of mobileBERT [48]. It is a compact and efficient version of eBERT that retains high accuracy while significantly reducing model size and inference latency. More precisely, it is 4.3× smaller and 5.5× faster than eBERT while achieving comparable performance. We fine-tuned the cross-encoder on the labels coming from the LLM model of 50,078,315 records, with the cross-entropy loss on the dataset described above. When evaluated on a test set of 7,503,031 (item, keyphrase) pairs, it yielded a F1 score of 91% with a precision of 92% and recall of 90%. The cross-encoder achieved a F1 score of 96%, thus validating its use as an assistant model.

3.2.3 Inference phase. After the training phase described above, we used the fine-tuned cross-encoder to give scores for each sentence pair in the bi-encoder training set.

3.3 The bi-encoder

The bi-encoder model is a traditional BERT-based [9] bi-encoder model which encodes items titles (concatenated with their meta category name) on the one hand, and buyers keyphrases on the other. The projections from the items and the keyphrases are then compared using *cosine similarity*. Within this architecture, any transformer model can be used as the base model. Each input is first passed through the base transformer model, which produces token-wise contextualized representations. After this step, a mean pooling operation is applied to condense the token representations into a fixed-length vector. The choice of the base model significantly influences the quality of the derived embeddings. In the next section, we present the different base models that we have tried as part of our experiments.

3.3.1 The base model. We have experimented with 3 models, eBERT, MicroBERT and ModernBERT.

eBERT: Multilingual eBERT model [6], pre-trained on eBay item data and general domain. The architecture used is a BERT-base configuration with 12 layers, outputting a embedding of dimension 768.

MicroBERT: Compressed and distilled version of eBERT (around 4 times smaller, and around 5 times faster; trained with a procedure explained here [48]). It achieves a smaller size due to a smaller intermediate layer (size of the feedforward layer inside the transformer), of 384 compared to 3072 for the original; Output embedding dimension is still 768.

ModernBERT: [56] We used a version of modernBERT that is made multilingual through trans-tokenization and cross-lingual vocabulary transfers [40]. However, this base model was *not pre-trained on eBay data.* ModernBERT features many improvements over the original BERT architecture, including: longer sequence length (8192 tokens, compared to 512 for original BERT), the use of Rotary positional embeddings instead of absolute ones, alternating global and local attention (every third layer uses global attention; the rest use local sliding window attention - all of these use Flash attention). Generally speaking, it is deemed a better model than the original BERT model, with an overall GLUE score of 88.5 (compared to 80.5 for BERT-base).

3.3.2 The training process. As explained above, we would like to fine-tune a student bi-encoder model with a training scheme that *simultaneously* includes a rank imitation loss based on the assistant cross-encoder output and a multi-task hybrid training scheme based on separate ground truth labels. We therefore used different loss functions for each of our labels and put them together in a multi-task training process where each training/evaluation batch only contain samples from one of the datasets. For a training and evaluation batch, each dataset was sampled in proportion to its size. In the section below, we review the different labels and loss functions that we used for our multi-task training scheme.

Multiple Negatives Ranking Loss. The Multiple Negatives Ranking (MNR) Loss [16] is well-suited to cases where only positive pairs are available, as it does not require manually labeled negative samples. When fed with item-keyword pairs of positive examples, this loss uses one item as its anchor, uses its given keyword as a positive example, and considers all other keywords in the training batch as negative for this anchor item (IRNS). This approximation works well with highly-sparse datasets such as e-commerce and web datasets.[3] In our use-case, as explained in [34], CTR-based

[3]While a better solution would be to employ better hard negative mining as illustrated in D2LLM, ANCE and other works; due to the size of our dataset the process proved

signals provide reliable positive sequence pairs, but not reliable negative pairs. Therefore, we used the MNR loss on the CTR-based labels.

$$\mathcal{L}_{\text{MNR}} = -\log \frac{\exp\left(\frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\tau}\right)}{\sum_{k=1}^{K} \exp\left(\frac{\mathbf{z}_i \cdot \mathbf{z}_k}{\tau}\right)} \tag{1}$$

where:

- $\mathbf{z}_i$ and $\mathbf{z}_j$ are the embeddings of the positive pair,
- $\mathbf{z}_k$ is the embedding of a negative sample,
- τ is the temperature parameter,
- K is the total number of negative samples.

Contrastive Loss. Contrastive loss [15] explicitly optimizes the embedding space by bringing similar sentence pairs closer together and pushing dissimilar pairs apart. This loss function is therefore well-suited to cases like ours, that rely on Approximate Nearest Neighbor search at prediction time. We used this loss function on both our LLM labels and our SR labels (which both include positive and negative examples). Mathematically this loss is defined as:

$$\mathcal{L}_{\text{Contrastive}} = \frac{1}{2}\left(y \cdot d(u,v)^2 + (1-y) \cdot \max(0, m - d(u,v))^2\right) \tag{2}$$

where:

- y is a binary label: $y = 1$ if the pair is similar, and $y = 0$ if the pair is dissimilar.
- $d(u, v)$ is a distance function (cosine distance in our case).
- m is a margin hyperparameter that sets the minimum required separation for dissimilar pairs.

This loss function encourages smaller distances for similar sentence pairs (y=1) and larger distances for dissimilar pairs (y=0).

Pearson correlation Loss. As shown in [26], maximizing the Pearson correlation between the student's logits and the teacher's logits enables the student model to replicate the teacher's subtle ranking nuances. It does that by minimizing the Pearson rank imitation loss, defined as:

$$\mathcal{L}_{\text{RI}}^{Pearson} = 1 - \text{corr}(z_{T_i}, z_{S_i}) \tag{3}$$

where z_T and z_S are the scores from the teacher and the student.

In [26], this loss is especially employed on BM25 hard negatives. In our case, we already have the hard negatives from the teacher LLM and subsequently from the cross-encoder assistant and don't require any hard negative filtering. We therefore used this loss directly on all of the student's logits and teacher's logits.

CoSENT Loss (Cosine Sentence Loss). This is another loss used during knowledge distillation [22]. Mathematically, it is computed as:

$$\mathcal{L}_{\text{CoSENT}} = \log \sum_{(i,j),(k,l)} (1 + \exp(s(i,j) - s(k,l)))$$

to be quite expensive. We acknowledge a potential for improvement in this area and leave this up to future research.

Here, (i, j) and (k, l) are any input pairs in the batch such that the cross-encoder-based similarity of (i, j) is greater than (k, l). s is the bi-encoder-based similarity function.

MSE Loss. This is the traditional MSE loss, calculated as the Mean Squared Error between the cross-encoder similarity scores, and the cosine similarity scores for the bi-encoder embeddings for items and keyphrases.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - cos(u,v)_i)^2 \tag{4}$$

where u and v are the embeddings for the item and keyphrase respectively and y is the score of the cross-encoder.

3.3.3 Matryoshka embeddings. Embedding dimensionality directly impacts latency and computational cost. Using Matryoshka embeddings [25] significantly reduces the cost of nearest-neighbor search while not significantly impacting retrieval accuracy and precision. Matryoshka Representation Learning (MRL) is a technique that enables learning representations of varying sizes within a single high-dimensional vector. Instead of training multiple models for different embedding sizes, MRL optimizes a nested structure where the first m dimensions of an embedding are designed to be as informative as an independently trained m-dimensional representation. This hierarchical structure is achieved with minimal computational overhead and ensures efficient deployment across different tasks [25]. This is achieved by modifying the loss function: in addition to the initial loss, the loss value for each Matryoshka dimensionality is added to the overall loss function. We reduced our dimension size to 64 for faster ANN.

4 Experimentation and Ablation Studies

In this section, we present our offline experiments and ablation studies. Please note that, due to time and resource constraints, we performed our ablation studies on a subset of the 3 million (item,keyphrase) pairs test set. The few evaluations that we did run on the entire dataset were directionally similar to the ones observed on the smaller test set, with a F1-score improvement from 67% to 83% (between the baseline model and our proposed solution). Furthermore, as many different losses and architectures need to be compared, we chose to evaluate all of them *on the LLM-labeled test set only*, for reasons and clarity and readability. At the end of this section, we present our offline evaluation algorithm, which mimics production settings and incorporates estimations of uniqueness and diversity, similar to the detailed evaluation scheme presented in [34].

4.1 Modifying the base transformer model

In this first analysis, we have fine-tuned our bi-encoder model using the LLM-labeled training set with the contrastive loss function. The only parameter that we have changed here is the base model. We report the classification metrics we observed when using microBERT, modernBERT and eBERT as our base model.

As shown in Table 1a, both the microBERT and the eBERT models give better performance than modernBERT, even though modernBERT has a higher GLUE score than BERT (88.5 vs 80.5) and a

Base models	Recall	Precision	F1
MicroBERT	0.92	0.78	0.85
eBERT	0.92	0.81	0.86
ModernBERT	0.91	0.76	0.83

(a) Changing the base model

K.D. Loss	C.E. corr	F1	Precision	Recall
MSE	0.62	0.82	0.75	0.91
CoSENT	0.76	0.88	0.86	0.90
Pearson Loss	0.79	0.89	0.87	0.91

(b) Changing the KD Loss

Table 1: Ablation studies: (a) base models and (b) KD loss

much higher context length (8192 vs 512). This result illustrates *the importance of pre-training*, as the modernBERT version that we used here was not pre-trained on eBay's vocabulary.

Table 1a also shows that using eBERT as our base encoder yields slightly better results than microBERT. This is expected, as microBERT is a distilled version of eBERT. Due to the size of our dataset, we chose to use microBERT for the rest of this study, as it shortens the batch prediction time by 30% on average.

4.2 Assessing the multi-task framework

In this section, we present the influence of multi-task training on the bi-encoder performance. In previous studies involving multi-task learning [26], we felt the effect of each component was not clearly studied — albeit there is a limited amount of ablation studies one can perform. In this study however we set out to perform a more comprehensive set of ablation studies to segregate each section of the multi task framework. We first started with the basic CTR labels and then started adding the contrastive labels (SR and LLM) and then added the KD cross-encoder scores distilled from LLM for additional signal and more accurate calibration.

We also additionally test a variety of KD losses. During the knowledge distillation process, the bi-encoder is trained on the soft outputs of the cross-encoder. Therefore, it is valid to evaluate this process based on the *Pearson correlation* between the soft outputs of the bi-encoder and those of the cross-encoder[4]. We also evaluate these losses with the bi-encoder's F1 score on the test set. In CUPID [3] it was stated that MSE loss performs poorly for distillation from cross encoder to bi-encoder; we verify this in our own observations in Table 1b. Interestingly, we also tried other KD losses, like CoSENT which is technically a pairwise ranking loss with a calibration component to it, and the Pearson Correlation loss described in D2LLM [26] which has a batch-wise ranking calibration component to it expanding over the pairwise construction of CoSENT. Interestingly, the batch-wise rank imitation loss of Pearson-Correlation performs better than the pairwise CoSENT Loss which performs better than the pointwise MSELoss.

4.3 Offline Evaluation

The retrieval models at eBay are stacked with each retrieval model serving a different purpose (see [34] to review a comprehensive summary of what models are in production). Hence to obtain a comprehensive estimation of our solutions incremental impact, we must exclude keywords already suggested by other retrieval models in production — albeit the EBR (CTR) model we plan on replacing. Following this de-duplication, the remaining keywords are then passed through the downstream eBay Advertising relevance filter. Upon completing these steps, we can estimate the *median count of de-duplicated relevant keywords per item* (median kw cnt in Table 2) which are surfaced to the sellers. These procedures are outlined in algorithm 1. To also gain insights on alignment on seller judgment and search judgment a sample of 10000 records per model (after passing the relevance filter) is then put through our LLM (*Mixtral-8x7B-Instruct-v0.1*) and Search Relevance Model. What we observed was that for Search judgment we got a more than 99% alignment for all the models, indicative of auction efficiency that our model would bring. From our observations recorded in Table 2 we see that the LLM+CTR+KD and LLM+SR+KD labels perform the best in terms of efficiency with a median keyword 12 after the relevance filter that was surfaced to the sellers. In addition, conferring with LLM judgment which serves as a proxy for seller judgment we see that LLM+CTR+KD has the best performance with 70.57% of its keyphrases passing the LLM judgment, meanwhile the LLM+SR+CTR+KD is a close second. In lieu of these results we decided to test the best overall model LLM+CTR+KD in online settings.

Labels	median kw cnt	LLM pass rate
LLM+CTR+KD	12.0	70.57%
LLM+SR+KD	12.0	51.30%
LLM+KD	11.0	48.81%
LLM	11.0	60.85%
LLM+SR+CTR+KD	11.0	69.94%
KD	11.0	39.02%
SR+KD	11.0	45.79%
CTR	7.0	59.65%

Table 2: Ablation study - Effect of Labels and their corresponding training components. All models are retrieved with k=20.

Algorithm 1 Evaluation in Production settings

1: **Sample** a certain number of items from different sites.
2: Call all the other recall models on these sampled items.
3: **Embed** all items and keywords with the fine-tuned bi-encoder.
4: **for** each item **do**
5: Get the top k closest keywords in the embedding vector space.
6: Pass the resulting k keywords through the relevance model or the LLM.
7: Out of the resulting keywords from the previous step, take only the ones that are **not** in any other recall lists.
8: **Calculate** the median of this metric over all the items from step 1.

5 Production System Design

The production architecture depicted in Figure 4 comprises two main parts: *Near Real-Time (NRT)* Inference and Batch Inference. Batch inference handles items with a delay, while NRT prioritizes immediate items, particularly those newly created or updated by sellers. Batch inference has two components: 1) full batch inference

[4]We refer to this correlation as C.E. corr in our table

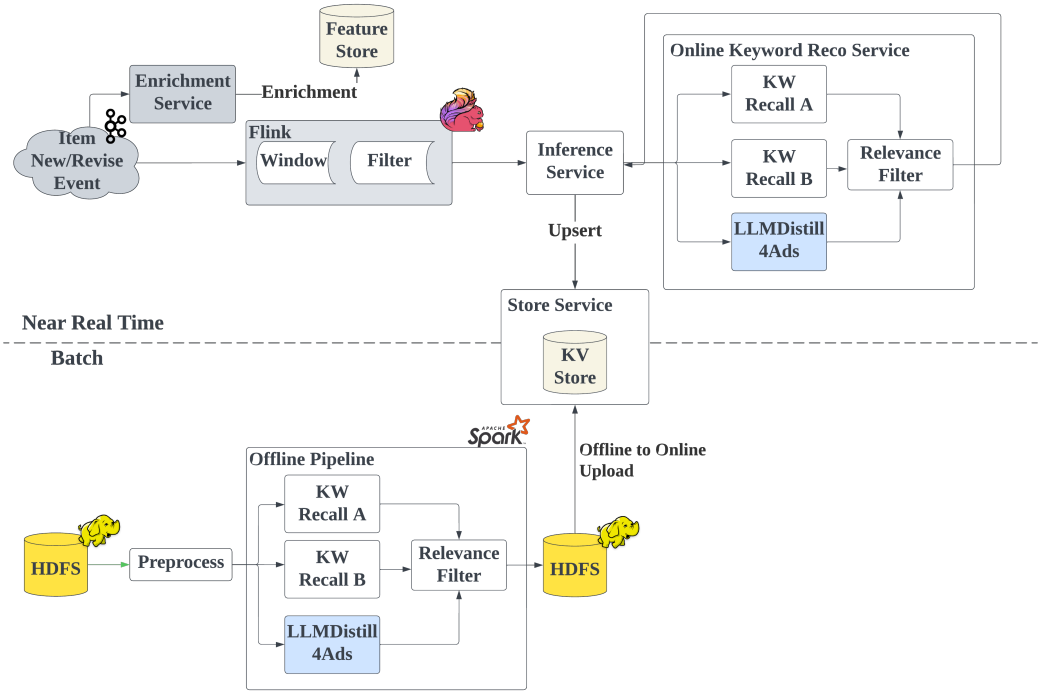


Figure 4: Production Serving Architecture for keyphrase recommendations.

for all items, and 2) daily differential (Diff) to integrate new and updated items with existing data. NRT inference utilizes triton and onnx serving using V100 GPUs, activated by item creation or updates managed by Flink processing and feature enrichment. The full batch handles approximately 2.3 billion items, while the daily Diff supports a churn of 20 million items. As the full batch runs just once, Diff latency determines model deployment viability, being about 35 minutes for bi-encoders. The ANN job downstream on a takes an additional 2.5 hours daily and for NRT our vector database service helps in that regard. Latency numbers reported for our batch inference use PySpark [61] (1500 executors, 20g memory, 4 cores), leveraging transformers [57] and onnxruntime [8].

6 Impact

We ran an A/B test with our new model as the treatment replacing the CTR only EBR model for 12 days in the US market. While we observed directional improvement in clicks (11.5%, $p = 0.25$), it was not statistically significant. However, we do see some statistically significant improvement in terms of GMB (Gross Merchandise Volume Bought, i.e. sales) which increased by 51.26% ($p = 0.01$) implying the better relevance of items converting same number of clicks to more sales. The return on advertising expenditure ($ROAS = \frac{GMB}{Ads\ Revenue}$), which marks the bottom line for the seller — the ratio of sales money to the money spent on advertising (which is proportional to clicks) experienced a notable improvement of 38.69% ($p = 0.02$).

7 Conclusion

This research thoroughly examines the constraints encountered when exclusively relying upon click-based indicators to fine-tune bi-encoder models specifically for the classification of sentence pairs within the e-commerce sector. Our investigations reveal that substituting traditional Click-Through Rate (CTR) signals with those generated by Large Language Models (LLMs) produces a marked enhancement in model performance. Furthermore, additional gains can be achieved by engaging in an intermediate cross-encoder model during the fine-tuning phase via knowledge distillation techniques. Notably, conducting knowledge distillation concurrently with training on supplementary *raw* labels confers further performance enhancements for the bi-encoder models. Our analysis further underscores the significance of the initial pre-training of the base model. Such preparatory steps enable comparatively smaller architectural models to surpass their more sizable counterparts in performance metrics. Moreover, our research identifies a rank imitation loss based on Pearson correlation as an exceptional knowledge distillation loss function, which notably outperforms both CoSENT and Mean Squared Error (MSE) loss functions. To supplement our findings, we introduce a rigorously structured evaluation protocol aimed at quantifying the business impact of potential candidate generator models under conditions that closely replicate those found in practical production environments.

References

[1] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 13–24, New York, NY, USA, 2013. Association for Computing Machinery.

[2] Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks, 2024.

[3] Arindam Bhattacharya, Ankith Ms, Ankit Gandhi, Vijay Huddar, Atul Saroop, and Rahul Bhagat. CUPID: Curriculum learning based real-time prediction using distillation. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 720–728, Toronto, Canada, July 2023. Association for Computational Linguistics.

[4] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Trans. Inf. Syst.*, 41(3), February 2023.

[5] K. Dahiya, N. Gupta, D. Saini, A. Soni, Y. Wang, K. Dave, J. Jiao, K. Gururaj, P. Dey, A. Singh, D. Hada, V. Jain, B. Paliwal, A. Mittal, S. Mehta, R. Ramjee, S. Agarwal, P. Kar, and M. Varma. Ngame: Negative mining-aware mini-batching for extreme classification. In *WSDM*, March 2023.

[6] Leonard Dahlmann and Tomer Lancewicki. Deploying a bert-based query-title relevance classifier in a production system: a view from the trenches, 2021.

[7] Romain Deffayet, Philipp Hager, Jean-Michel Renders, and Maarten de Rijke. An offline metric for the debiasedness of click models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 558–568, New York, NY, USA, 2023. Association for Computing Machinery.

[8] ONNX Runtime developers. Onnx runtime. https://onnxruntime.ai/, 2021. Version: 1.20.1.

[9] 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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[10] Soumik Dey, Hansi Wu, and Binbin Li. To judge or not to judge: Using llm judgements for advertiser keyphrase relevance at ebay, 2025.

[11] Soumik Dey, Wei Zhang, Hansi Wu, Bingfeng Dong, and Binbin Li. Middleman bias in advertising: Aligning relevance of keyphrase recommendations with search, 2025.

[12] Jingyue Gao, Shuguang Han, Han Zhu, Siran Yang, Yuning Jiang, Jian Xu, and Bo Zheng. Rec4ad: A free lunch to mitigate sample selection bias for ads ctr prediction in taobao. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4574–4580, New York, NY, USA, 2023. Association for Computing Machinery.

[13] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025.

[14] Omkar Gurjar, Kin Sum Liu, Praveen Kolli, Utsaw Kumar, and Mandar Rahurkar. Dashclip: Leveraging multimodal models for generating semantic embeddings for doordash, 2025.

[15] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[16] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*, 2017.

[17] Junqin Huang, Zhongjie Hu, Zihao Jing, Mengya Gao, and Yichao Wu. Piccolo2: General text embedding with multi-task hybrid loss training, 2024.

[18] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021.

[19] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[20] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7–es, apr 2007.

[21] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 781–789, New York, NY, USA, 2017. Association for Computing Machinery.

[22] Kexue.fm. Cosent(yi): Bi sentence-bert geng you xiao de ju xiang liang fang an. https://kexue.fm/archives/8847, 2022. Accessed: 2025-04-02.

[23] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery.

[24] Taehyeon Kim, Jaehoon Oh, Nak Yil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2021.

[25] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning, 2024.

[26] Zihan Liao, Hang Yu, Jianguo Li, Jun Wang, and Wei Zhang. D2LLM: Decomposed and distilled large language models for semantic search. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14798–14814, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[27] Daryl Lim, Julian McAuley, and Gert Lanckriet. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 309–312, 2015.

[28] Daryl Lim, Julian McAuley, and Gert Lanckriet. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 309–312, New York, NY, USA, 2015. Association for Computing Machinery.

[29] Zhenghao Lin, Yeyun Gong, Xiao Liu, Hang Zhang, Chen Lin, Anlei Dong, Jian Jiao, Jingwen Lu, Daxin Jiang, Rangan Majumder, and Nan Duan. Prod: Progressive distillation for dense retrieval. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3299–3308, New York, NY, USA, 2023. Association for Computing Machinery.

[30] Wenhao Lu, Jian Jiao, and Ruofei Zhang. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20)*, page 2645–2652, 2020.

[31] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval, 2022.

[32] Xueguang Ma, Xi Victoria Lin, Barlas Oguz, Jimmy Lin, Wen-tau Yih, and Xilun Chen. DRAMA: Diverse augmentation from large language models to smaller dense retrievers. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30170–30186, Vienna, Austria, July 2025. Association for Computational Linguistics.

[33] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.

[34] Ashirbad Mishra, Soumik Dey, Marshall Wu, Jinyu Zhao, He Yu, Kaichen Ni, Binbin Li, and Kamesh Madduri. Graphex: A graph-based extraction method for advertiser keyphrase recommendation, 2025.

[35] Ashirbad Mishra, Soumik Dey, Jinyu Zhao, Marshall Wu, Binbin Li, and Kamesh Madduri. Graphite: A graph-based extreme multi-label short text classifier for keyphrase recommendation. In *Frontiers in Artificial Intelligence and Applications*, pages 4657–4664. IOS Press, October 2024.

[36] Ashirbad Mishra, Jinyu Zhao, Soumik Dey, Hansi Wu, Binbin Li, and Kamesh Madduri. Broadgen: A framework for generating effective and efficient advertiser broad match keyphrase recommendations, 2025.

[37] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet,

Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022.

[39] Przemysław Pobrotyn and Radosław Białobrzeski. Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. *arXiv preprint arXiv:2102.07831*, 2021.

[40] François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp, 2024.

[41] Mosaic research team. Introducing DBRX: A New State-of-the-Art Open LLM — databricks.com. https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm. [Accessed 16-04-2025].

[42] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.

[43] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics.

[44] Ian Soboroff. Don't use llms to make relevance judgments. *Information Retrieval Research*, 1(1):29–46, Mar. 2025.

[45] Aivin V. Solatorio. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning, 2024.

[46] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, page 713–722, New York, NY, USA, 2010. Association for Computing Machinery.

[47] Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. Large language models are inconsistent and biased evaluators, 2024.

[48] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic BERT for resource-limited devices. *CoRR*, abs/2004.02984, 2020.

[49] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online, June 2021. Association for Computational Linguistics.

[50] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. Large language models can accurately predict searcher preferences. SIGIR '24, page 1930–1940, New York, NY, USA, 2024. Association for Computing Machinery.

[51] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[52] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[53] Francis Vella. Estimating models with sample selection bias: A survey. *The Journal of Human Resources*, 33(1):127–169, 1998.

[54] Han Wang, Mukuntha Narayanan Sundararaman, Onur Gungor, Yu Xu, Krishna Kamath, Rakesh Chalasani, Kurchi Subhra Hazra, and Jinfeng Rao. Improving pinterest search relevance using large language models, 2024.

[55] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. The lambdaloss framework for ranking metric optimization. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1313–1322, 2018.

[56] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.

[57] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[58] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021.

[59] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

[60] Yisong Yue, Rajan Patel, and Hein Roehrig. Beyond position bias: examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 1011–1018, New York, NY, USA, 2010. Association for Computing Machinery.

[61] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache spark: a unified engine for big data processing. *Commun. ACM*, 59(11):56–65, October 2016.

[62] Ranran Haoran Zhang, Bensu Uçar, Soumik Dey, Hansi Wu, Binbin Li, and Rui Zhang. From lazy to prolific: Tackling missing labels in open vocabulary extreme classification by positive-unlabeled sequence learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1–16, 2025.

8 Appendix

8.1 KD Losses

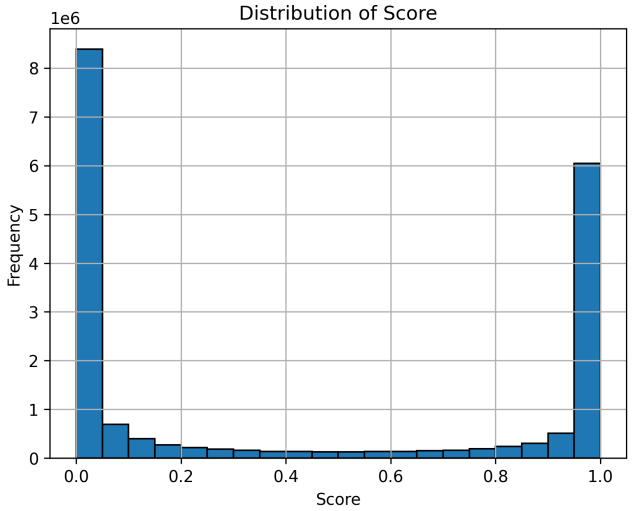
Following on this logic of MSE (pointwise) vs CoSENT (pairwise) vs. Pearson Correlation Loss (batch-wise), we tried additional advanced ranking losses, namely neural NDCG [39] and Lambda Loss [55], however, the results were horrible (less than 0.1 in recall and precision). Our suspicions are that these losses are generally applied with a seed query and have a rank misclassification penalty for disparate ranks — which might cause some issues, as the cross-encoder was never trained for specific ranking calibration, rather overall linear directionality and calibration appear to be main drivers of performance. This direction needs further research.

8.2 Fine-tuned vs General LLM

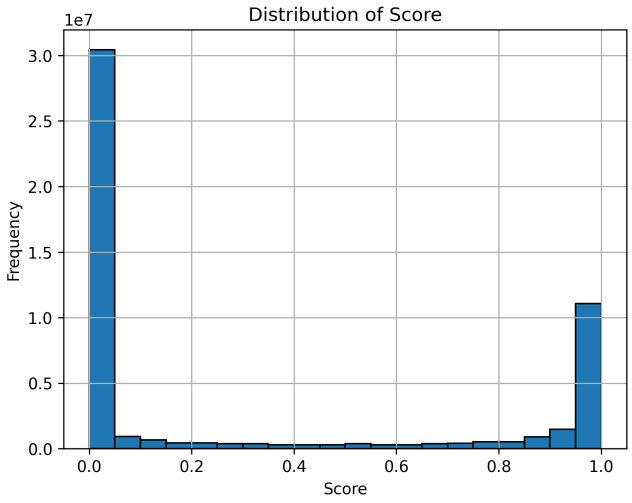
During LLM-as-a-judge data augmentation, we were faced with the option of using General vs fine-tuned LLM. General LLMs have an advantage in being less prone to biases that affect specialized models. Conversely, fine-tuned models are fine-tuned on human judgment and can propagate the biases present in the original small human judgment sample present. In a previous study [10] we found that the general LLM was more aligned with our business metrics than the fine-tuned LLM, although the fine-tuned LLM showed better alignment on the small amount of human judgment data. We later discovered problems with our human judgment data collection, and use general LLMs until we fix the issue with our human judgment data.

8.3 LLM Logits vs cross-encoder

In our study we could have skipped the cross-encoder and went directly for the LLM teacher generating the scores through logits of yes/no through constrained decoding. However, in our previous study [10] we found this to be not very helpful. We even tested this on our current EBR models and found results to be highly unsatisfactory. The main reason is the distribution of scores of cross-encoder which learns from the yes/no of the LLM and the softmax of the logits of the LLM as shown in Figure 5. As seen in the figure the cross-encoder has a much more even distribution for the extreme values of 0 and 1 where the golden class distribution is 50-50, whereas for the logits we see a much more disparate distribution. This is then propagated to downstream models which worsens the learning.



(a) Cross-encoder score distribution



(b) Softmax of logits of the LLM distribution

Figure 5: Kernel Density Estimation of the scores of cross-encoders and LLM probabilities.