E4EL: CRF-Based Efficient Zero-Shot End-to-End Framework for E-Commerce Entity Linking

Hsiao-Ying Lu[1,*], Zeyu Wang[2], Zhejun Shen[2] and Petar Ristoski[2]

[1]University of California, Davis, USA
[2]eBay Inc., San Jose, USA

Abstract

Entity linking is a critical component of query understanding in e-commerce search. For a given user query, entity linking approaches ground mentions to catalog or Knowledge Graph entities, which are used in downstream systems for retrieval, recommendation, and relevance ranking. Prior approaches typically follow two paradigms in general. First, two-stage systems, which employ a bi-encoder to retrieve candidate entities followed by a cross-encoder for disambiguation. While these systems can perform zero-shot entity linking, their reliance on transformer backbones that introduce non-trivial inference cost at e-commerce scale. Second, end-to-end generative models, which jointly detect mentions and generate corresponding entity names, but they cannot handle zero-shot entity linking for new inventory items whose names or IDs were unseen during training. To address these challenges, we present a lightweight alternative optimized for real-time e-commerce search. The framework combines a high-recall, low-latency mention and entity candidate detector with a multilayer perceptron trained using a conditional random field as the loss function for candidate disambiguation. The conditional random field refines detector predictions by modeling both mention-level and entity-level coherence, enabling context-aware resolution. Additionally, a graph neural network is trained to produce compact entity embeddings that capture the topological features of the Knowledge Graph, yielding manageable vector representations that encode coherence information. Using eBay's inventory-centric knowledge graphs, we evaluate the method against our current production baseline and show that it achieves higher F1 while maintaining low-latency inference. These results demonstrate that the proposed design enables both accurate and fast performance, making it well-suited for industrial-scale deployment in e-commerce search systems.

Keywords

Entity Linking, Knowledge Graph, CRF, GNN, e-commerce

CEUR-WS.org/Vol-4335/paper10.pdf

1. Introduction

Entity Linking (EL) maps mentions in user queries to corresponding Knowledge Graph (KG) entities, transforming free-form text into structured signals that downstream retrieval and ranking systems can leverage. In an e-commerce setting, EL presents unique challenges due to two key factors: (1) product listings are continuously updated, leading to a highly dynamic KG structure; and (2) strict service-level objectives require low-latency inference at large scale. Consequently, any practical solution must handle newly introduced entities without retraining and remain sufficiently lightweight to support real-time search.

Prior work on entity linking can be broadly categorized into two paradigms: retrieval-based and generative-based approaches. Retrieval-based methods [1, 2, 3] typically adopt a two-stage architecture. A bi-encoder independently encodes the mentions and entities to retrieve candidates based on embedding proximity. Each candidate is then re-ranked using a cross-encoder that jointly encodes the mention and entity text. While this framework supports zero-shot EL, allowing the model to infer representations for unseen mentions or newly added entities, its reliance on transformer backbones incurs substantial inference latency, posing scalability challenges in e-commerce environments. Generative-based methods

NLP4KGC 2026: 5th International Workshop on Natural Language Processing for Knowledge Graph Construction. Co-located with ISWC 2026, October 26, 2026, Bari, Italy

*Corresponding author.
hyllu@ucdavis.edu (H. Lu); zwang14@ebay.com (Z. Wang); zhejshen@ebay.com (Z. Shen); pristoski@ebay.com (P. Ristoski)
0009-0000-1145-9229 (H. Lu); 0000-0002-1890-1507 (P. Ristoski)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-19

[4] formulate EL as a sequence-to-sequence generation task. In this end-to-end framework, the model identifies mentions within a query and generates the corresponding entity names in sequence. However, such approaches are inherently constrained to entities encountered during training and therefore cannot handle zero-shot EL. Moreover, both retrieval-based and generative-based approaches often assume the availability of gold mentions or rely on rich textual descriptions, information that may not always be present, thus limiting their applicability to KGs with incomplete or sparse textual metadata.

To address the limitations of existing EL methods in achieving zero-shot generalization and low-latency performance for e-commerce applications, we present **E4EL**, an **E**fficient zero-shot **E**nd-to-end framework for **E**bay-scale **E**-commerce **E**ntity **L**inking, which jointly performs mention detection and entity disambiguation. The pipeline begins with a high-recall mention generator that, in a single pass, identifies potential mentions and retrieves their corresponding entity candidates by lexical matching each query n-gram span to entity names in the KG (e.g., Nike → brand/nike). This high-recall design ensures broad candidate coverage while alleviating the computational burden typically associated with mention detection, thereby allowing the framework to focus its complexity on the subsequent candidate disambiguation stage.

Our candidate disambiguation module operates on the high-recall outputs, where all detected mentions and their potential entity links are treated as candidates rather than final predictions. To determine the optimal set of non-overlapping mentions and their most coherent linked entities, we introduce the *Double-Coherence Conditional Random Field (DC-CRF)*. This model jointly enforces mention-level and entity-level coherence, selecting combinations that maximize overall consistency. For instance, given the candidate mappings `Apple` → `fruit/apple`, `Apple` → `brand/apple`, `iphone` → `model/iphone`, the combination excluding the first candidate yields greater semantic coherence than alternatives such as pairing `fruit/apple` with `model/iphone`. The coherence signal is derived from the similarity between entity embeddings. To obtain these embeddings, we train a lightweight Graph Neural Network (GNN) on the eBay KG [5] with a link prediction objective. The GNN learns to capture entity semantics from the KG's topological structure, allowing it to perform zero-shot inference on new entities (i.e., generating their embeddings) without retraining once their graph connections are updated.

In summary, our main contributions are as follows:

- A Double-Coherence Conditional Random Field (DC-CRF), coupled with a high-recall candidate generator and a GNN-based entity encoder, forms a low-latency, end-to-end, zero-shot EL framework that jointly disambiguates mention and entity candidates.
- An industrial evaluation on eBay data, demonstrating higher F1 performance than a production baseline while reducing inference latency, indicating suitability for large-scale deployment.

The rest of this paper is structured as follows. In Section 2, we give an overview of related work. In Section 3, we introduce our approach, followed by an evaluation in Section 4. We conclude with a summary and an outlook on future work.

2. In-Use Context and System Overview

E4EL is designed and evaluated within the knowledge graph query resolution pipeline of a large-scale e-commerce search system. In this setting, entity linking operates as an upstream component whose outputs are consumed by downstream services including retrieval, ranking, and recommendation. The system processes user queries at high throughput ($10k+$ queries per second) under strict latency constraints, where entity linking must execute synchronously within the end-to-end request budget.

Role in the production pipeline. Given a user query, the entity linking component identifies relevant aspect-value entities (e.g., brand/nike or model/jordan_5), which are used to (i) enrich retrieval queries, (ii) provide structured features for ranking models, and (iii) enable downstream recommendation modules. Errors in entity linking propagate to downstream systems and directly affect search relevance and user experience.

Operational constraints. The design of E4EL is driven by several production requirements:

- **Strict latency budgets:** The baseline production system operates at approximately 20ms per query.
- **High throughput:** The system must scale to $10k+$ QPS with predictable latency.
- **CPU-only inference:** Cost-efficient and scalable solution.
- **Dynamic inventory:** New entities are continuously introduced, making frequent retraining impractical.

These constraints rule out many existing entity linking approaches that rely on transformer-based encoders or retrieval-heavy architectures due to their higher inference cost. Additionally, generative-based models do not support zero-shot inference and thus, excluded due to the dynamic inventory constraint.

System design implications. E4EL is explicitly designed to satisfy these constraints. Candidate generation is implemented via efficient n-gram matching over entity lexical forms, ensuring affordable runtime. Disambiguation is performed using a lightweight CRF-based model that enforces global coherence without requiring expensive contextual encoding. Entity representations are precomputed offline using a GNN, decoupling training cost from online inference latency.

Why knowledge graph-based modeling. A central design choice in E4EL is the use of a knowledge graph to represent relationships between aspect–value entities. Many disambiguation errors arise from selecting combinations of entities that are individually plausible but mutually inconsistent (e.g., incompatible brand–model pairs). Such constraints cannot be effectively captured using flat feature representations, string matching, or independent classifiers.

Deployment context. E4EL is designed as an upgrade to the existing production entity linking pipeline within eBay's knowledge graph query resolution service. The underlying Product Knowledge Graph (PKG) has been deployed in production for several years, serving multiple applications including query reformulation, retrieval enhancement, personalized recommendation [5], KG-enhanced retrieval augmented generation for LLMs [6], and entity-aware unit-of-measurement handling in search [7]. The entity linking baseline that E4EL replaces currently serves live search traffic at $10k+$ QPS. E4EL has been evaluated under identical infrastructure and latency constraints as the production system, using real anonymized search queries judged by human annotators. Its integration path and deployment readiness are discussed in detail in Section 6.

3. Related Work

The task of entity linking has been extensively studied in the literature, resulting in a plethora of related work. Most of the related approaches can be categorized in four different categories.

Zero-shot end-to-end entity linking. In general, the entity linking task can be broken into two steps: mention detection (MD) and entity disambiguation (ED), where ED includes candidate generation and candidate ranking. BLINK [1] is a two-stage *zero-shot* ED system: it assumes gold mentions and focuses on dense retrieval plus cross-encoder re-ranking. ReFinED [2] is an *end-to-end* and *zero-shot-capable* model that performs mention detection, fine-grained typing, and disambiguation in a single pass, combining entity descriptions with type signals and a NIL option for out-of-KB cases; it reports strong accuracy at substantially lower inference cost than cross-encoder pipelines. Our approach is likewise *zero-shot* and *end-to-end*, but avoids description encoders at inference time: we retrieve high-recall candidates based on the lexical similarity and apply a Double-Coherence CRF for disambiguation using GNN-based entity embeddings. Newly added KG entities are linkable as soon as they are indexed and embedded, without retraining.

CRF-based structured inference. CRF-style inference is widely used to select coherent combinations of outputs: in sequence tagging, BiLSTM-CRF models improve consistency over local decisions [8, 9]. In vision, CRF-as-RNN enforces pairwise compatibility during end-to-end training [10]. In EL

specifically, neural CRF/graph inference strengthens document-level coherence [11, 12]. Our Double-Coherence CRF follows this structured-prediction principle but jointly disambiguates mentions and enforces entity–entity compatibility via precomputed GNN embeddings.

Graph-aware representations for EL. Knowledge-graph structure is highly informative for disambiguation, and several systems inject it explicitly. Graph neural networks such as GCN [13], GraphSAGE [14], and R-GCN for multi-relational graphs [15] learn node embeddings by propagating information along typed edges. EL models then use these structure-aware embeddings to inform decisions toward type-consistent, topologically compatible entities (e.g., NCEL's mention–entity GCN) [16]. Our entity embeddings are learned offline using a lightweight GCN trained for link prediction on our inventory graph. As a result, online inference relies solely on simple MLP-based disambiguation rather than computationally expensive description encoders or transformer-based rerankers. This design enables structure-aware decision making while maintaining low-latency performance.

Emerging generative/LLM-based EL. GENRE treats EL as constrained generation of the entity title with a seq2seq model [4]. This approach simplifies indexing but introduces decoding latency. Recent LLM-augmented approaches use LLMs to augment context or to judge among candidates. For example, EntGPT proposes prompt-based and instruction-tuned ED [17], not end-to-end EL. LLMAEL uses LLM to generate mention-centered descriptions to improve conventional EL despite keeping a specialized linker at inference time [18]. While these approaches can improve robustness and long-tail performance, they typically cannot be adopted in zero-shot scenarios, and often incur high inference latency and/or depend on textual descriptions or passages. By contrast, our description-free GNN+CRF design remains *end-to-end* and are suitable for online applications.

4. E4EL

In this section, we introduce the key components within our entity linking framework, E4EL: a set of high-recall candidates generated for Double-Coherence CRF module to jointly disambiguate mentions and entities, using GNN-based entity embeddings, trained on the eBay Product Knowledge Graph.

4.1. eBay Product Knowledge Graph

We construct a directed, weighted inventory knowledge graph based on co-occurring aspect–value pairs extracted from product listings [5]. Each listing l_n is represented as a set of aspect–value pairs $AV_n = \{av_1, av_2, \ldots, av_n\}$, where a denotes an aspect name (e.g., *Brand*) and v its corresponding value (e.g., *Nike*). Each aspect–value pair is treated as an entity node and associated with its global popularity across listings. When two entities co-occur in a listing, directed edges are created between them, weighted by their normalized co-occurrence frequency relative to the source entity. This results in category-specific graphs (e.g., sneakers, wristwatches).

Schema and representation. Aspect names define entity types (e.g., brand, model), and each aspect–value pair is assigned a corresponding type. Edges represent relationships between entities that frequently appear together in listings. The edge weights are asymmetric, reflecting the conditional strength of association between entities. This representation captures compatibility relationships (e.g., brand–model consistency), which are important for disambiguating entities in short and ambiguous queries.

Quality assurance and construction pipeline. To ensure data quality, entities are included only if supported by multiple independent sellers, following a "wisdom of the sellers" principle that reduces noise from individual listings. Additional steps include frequency-based filtering, normalization of surface forms, and pruning of weak co-occurrence edges. The graph is updated incrementally to reflect changes in inventory, allowing new entities to be incorporated without full reconstruction.

Serving and downstream ecosystem. The knowledge graph is deployed as a shared production resource supporting entity lookup and downstream applications such as query reformulation, retrieval-augmented generation [6], and recommendation [5]. This shared infrastructure enables reuse of entity representations across systems and supports real-time applications at scale. While constructed from co-occurrence signals, the graph encodes compatibility relationships between aspect–value pairs and supports higher-order dependencies (e.g., brand–model–category consistency). These relational signals are critical for resolving ambiguity and enable structure-aware embeddings that generalize to newly introduced entities.

4.2. High-Recall Candidate Generation

Most CRF-based entity linking methods assume that mention detection provides a precise set of spans, after which the CRF selects the most coherent entity assignments. In practice, however, achieving high-precision mention boundaries is both computationally expensive and costly to annotate. To overcome this bottleneck, we exploit the lexical similarity between query n-grams and the textual attributes of entities in the knowledge graph (e.g., query "nike jordan 5" to entity "brand/nike" and "model/jordan_5"). This matching procedure yields both a high-recall set of mention candidates and a high-recall set of entity candidates in a single step, since each mention is only retained if it can be matched to at least one KG entity. This strategy quickly obtains broad coverage of possible entities, and shifts the main challenge to disambiguation on both the mention and entity candidates.

4.3. Double-Coherence CRF: Candidate Disambiguation

Taking the high-recall candidates as input, our Double-Coherence CRF jointly optimizes mention-side coherence (i.e., avoiding overlapping or fragmented spans) and entity-side coherence (i.e., selecting mutually compatible entities in the KG). This design reduces annotation and inference costs while maintaining high accuracy with very low latency, making it well suited for industrial-scale e-commerce search.

4.3.1. Data Processing and Labeling

The objective of the Double-Coherence CRF is to identify the most coherent set of entities for a given query. To construct training data, we first generate entity candidates for all potential mentions extracted from the query. A large language model (LLM) then selects the most coherent entity set among these candidates, which we denote as the ground-truth entity set (*GT_entity_set*); this selection is subsequently validated by human annotators.

Because the CRF operates over combinations of candidate assignments, our labeling strategy is designed to learn two complementary tasks: (i) which mentions should be retained and (ii) which entities should be linked.

- **Mentions.** For each query, the ground-truth mention set *GT_mention_set* is obtained by aligning mentions with entities in *GT_entity_set*. Only mentions corresponding to these entities are retained.
- **Entities.** For each candidate combination, one entity is assigned to every potential mention, regardless of whether the mention belongs to *GT_mention_set*.
  - A **positive combination** includes all entities in *GT_entity_set*. Multiple positive combinations may exist, since assignments for mentions outside *GT_mention_set* do not affect correctness.
  - A **negative combination** is any combination that omits at least one entity from *GT_entity_set*.

This labeling strategy enables the CRF to jointly learn mention selection and entity disambiguation. The mention module filters irrelevant spans, while the entity module recovers the correct entity set. A prediction is correct only when both components succeed.

4.3.2. Model Overview

Our Double-Coherence CRF is trained to assign higher scores to positive combinations than to negative ones. The score of a combination is defined as

$$S(x, y) = MLP_{aggregation}(\sum_{i=1}^{n} z_i \cdot \psi_u(m_i, e_i), \sum_{i<j} z_i \cdot z_j \cdot \psi_p(e_i, e_j)), \tag{1}$$

where x denotes the set of candidate mentions, y denotes the corresponding set of selected entities, $\psi_u(m_i, e_i)$ computes the relevance score for a mention-entity pair, and $\psi_p(e_i, e_j)$ measures the coherence between two entities. The variable z_i acts as a soft mask, scaling the influence of mention m_i and its linked entity e_i when computing the CRF score. The $MLP_{aggregation}$ module learns to combine the aggregated mention-entity relevance scores with the entity-entity coherence scores into a final compatibility score. In the following paragraphs, we describe each module contributing to the CRF score in more detail.

Relevance function. For each mention m_i with word set M_i and its candidate entity e_i with word set E_i, we define the relevance score as the Jaccard similarity:

$$\psi_u(m_i, e_i) = \frac{|M_i \cap E_i|}{|M_i \cup E_i|}. \tag{2}$$

Entity coherence modeling. The coherence between two entities is modeled using their length, popularity, and learned embeddings:

$$\psi_p(e_i, e_j) = MLP_{entity}(length_i, length_j, popularity_i, popularity_j, Emb_i, Emb_j), \tag{3}$$

where MLP_{entity} first projects each feature type into a representation in shared latent space, then concatenates them to capture cross-feature interactions, and finally outputs a pairwise coherence score. The entity embeddings (Emb_i, Emb_j) are pre-computed with a Graph Neural Network (GNN) trained on the eBay product knowledge graph, which encodes structural and semantic relationships among entities (see Section 4.4).

Mention coherence modeling. As described in data processing and labeling, each potential mention (i.e., a n-gram span) within a query is given a binary label that specifies whether it belongs to the most coherent mention set ($GT_mention_set$):

$$\text{keep}(m_i) = \begin{cases} 1, & \text{if } m_i \in GT_mention_set \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In addition to modeling coherence on the entity side, it is important to explicitly reason about which mentions should be retained. Using only mention overlap statistics risks discarding valid mentions and overlooks informative signals from the linked entities. To address this, we train an $MLP_{mention}$ that jointly incorporates both mention and entity features to predict whether a mention should be retained. Its output is a probability value

$$z_i = \sigma\big(MLP_{mention}(\text{overlap statistics, entity type, entity popularity})\big), \tag{5}$$

where $z_i \in [0, 1]$ acts as a *soft mask*. Rather than making a hard binary decision, z_i weights the contribution of mention m_i and its linked entity e_i in Equation 1, allowing the model to down-weight unlikely mentions while still maintaining differentiability during training. The input features include: (i) overlapping statistics among all potential mentions in the query, (ii) a one-hot encoding of the entity type corresponding to the entity chosen for that mention in a combination (e.g., brand for Nike, or model for Jordan 5), and (iii) the normalized popularity of the entity, measured by its frequency across all eBay listings. The overlap features allow the model to prefer non-overlapping mentions, while the entity features enable it to consider both semantic plausibility (via type information) and real-world relevance (via popularity). This design ensures that mention selection decisions are informed not only

by surface-level string interactions but also by the quality and contextual importance of the underlying entities.

Loss functions. All trainable modules, $MLP_{mention}$, MLP_{entity}, and $MLP_{aggregation}$, are optimized jointly using three losses: $\mathscr{L}_{mention}$, $\mathscr{L}_{CRF}$, and $\mathscr{L}_{margin}$. $\mathscr{L}_{mention}$ is a cross-entropy loss that supervises $MLP_{mention}$ module by comparing the predicted soft mask z_i with the ground-truth keep/not-keep label (i.e., keep(m_i)) for each mention m_i. $\mathscr{L}_{CRF}$ is derived from the standard conditional random field objective, which encourages positive combinations to receive higher scores than negative ones. For a query, let $\mathscr{Y}$ denote the set of all combinations and y^* the positive combinations. The probability of all positive combinations is defined as:

$$P(y^* \mid x) = \frac{\sum_{y' \in y^*} \exp(S(x, y'))}{\sum_{y' \in \mathscr{Y}} \exp(S(x, y'))}, \tag{6}$$

and the CRF loss is given by

$$\mathscr{L}_{CRF} = -\log P(y^* \mid x) = -S(x, y^*) + \log \sum_{y' \in \mathscr{Y}} \exp(S(x, y')). \tag{7}$$

To further enforce a clear margin between positive and negative combinations, we add a margin-based loss:

$$\mathscr{L}_{margin} = ReLU\big(margin - \big(\min_{y \in y^*} S(x, y) - \max_{y' \in \mathscr{Y} \setminus y^*} S(x, y')\big)\big). \tag{8}$$

In terms of supervision, $MLP_{mention}$ is trained only with $\mathscr{L}_{mention}$ to ensure that the learned soft mask values align closely with mention labels. Both MLP_{entity} and $MLP_{aggregation}$ are trained with all three losses, allowing them not only to optimize coherence scoring but also to incorporate feedback from mention-side supervision into the joint training process.

4.4. Graph Neural Network: Entity Embeddings

Graph Neural Networks (GNNs) provide a principled way to learn representations directly from knowledge graphs by propagating information along typed edges. GNNs operate on the explicit relational topology: each entity's embedding is iteratively refined by messages from its neighbors, weighted by their connecting relations. This process integrates local features (immediate neighbors, edge types/weights) with multi-hop context (paths and higher-order neighborhoods), yielding embeddings that reflect both an entity's intrinsic attributes and its role in the graph. Such structure-aware representations are especially valuable for entity linking, where disambiguation often depends more on an entity's connections, which infer semantical relevance.

4.4.1. Architecture.

We model the knowledge graph as $G = (V, E)$ with bi-directional nonnegative edge weights w_{ij}. Each node $v_i \in V$ has an initial feature vector $\mathbf{h}_i^{(0)}$ (e.g., entity attributes or pooled text features). Information is propagated layer by layer so that node i updates its representation by combining messages from its neighbors $N(i)$ and an explicit self-term:

$$\mathbf{h}_i^{(\ell+1)} \;=\; \sigma\!\left(\sum_{j \in N(i)} \alpha_{ij}\, \mathbf{W}^{(\ell)} \mathbf{h}_j^{(\ell)} \;+\; \mathbf{W}_0^{(\ell)} \mathbf{h}_i^{(\ell)}\right), \tag{9}$$

where $\mathbf{W}^{(\ell)}$ and $\mathbf{W}_0^{(\ell)}$ are trainable weight matrices and $\sigma(\cdot)$ is the ReLU activation function. Following the GCN normalization, neighbor contributions are scaled by

$$\alpha_{ij} \;=\; \frac{w_{ij}}{\sqrt{\tilde{d}_i\, \tilde{d}_j}}, \qquad \tilde{d}_i \;=\; 1 + \sum_{k \in N(i)} w_{ik}, \tag{10}$$

so that higher-weight edges contribute more while degrees are symmetrically normalized; the +1 in $\tilde{d}_i$ accounts for a self-loop.

In matrix form, let A be the weighted adjacency with $A_{ij} = w_{ij}$, $\tilde{A} = A + I$, and $\tilde{D}$ the diagonal degree matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. With node-feature matrix $H^{(\ell)}$, one layer reads

$$H^{(\ell+1)} \;=\; \sigma\Big(\tilde{D}^{-\frac{1}{2}}\,\tilde{A}\,\tilde{D}^{-\frac{1}{2}}\,H^{(\ell)}\mathbf{W}^{(\ell)} \;+\; H^{(\ell)}\mathbf{W}_0^{(\ell)}\Big). \tag{11}$$

For scalability in mini-batch training, neighbors of i are sampled with probability proportional to edge weights,

$$p(j \mid i) \;\propto\; w_{ij}, \tag{12}$$

which biases message passing toward higher-confidence edges while keeping batch sizes bounded.

4.4.2. Training objective.

We trained the entity embeddings with a link-prediction objective framed as binary classification over node pairs. Observed graph edges are labeled as positives, and sampled non-edges are labeled as negatives. For a candidate pair (i, j), we compute a scalar using a dot-product decoder $s_{ij} = \mathbf{h}_i^\top \mathbf{h}_j$, and map it to a value in $(0, 1)$ using the logistic sigmoid. The model is optimized with standard binary cross-entropy over the positive and negative sets. For negative sampling, we sample two negatives for each positive edge (i, j) by incorporating edges that don't exist.

5. Experiments

We evaluate E4EL using eBay Knowledge Graph on anonymized eBay search queries sampled from three high-volume categories: sneakers, smartphones, and wristwatches. This section describes the data and preprocessing, the task and metrics, baselines and results.

5.1. Dataset and Preprocessing

Our evaluation dataset consists of anonymized eBay search queries from three major product categories, each with substantial knowledge captured in the eBay KG. For each category, we select top 10,000 search queries on eBay. For each query we ran a human judgment job, with 5 judges, to extract all KG entities and their mentions. The KGs capture the co-occurrence of aspects from eBay listing items, containing aspect-value pairs such as `brand: nike` and `model: Air Max 90`. During graph construction, each aspect-value node is enriched with alternative labels (altLabels) that typically represent abbreviations or common variations, enabling effective entity retrieval. For instance, the brand `louis vuitton` includes the alternative label `lv`, facilitating flexible entity candidate retrieval.

Data Preprocessing: We apply standard text normalization techniques including lowercasing, punctuation removal, and tokenization.

5.2. Task Definition and Evaluation Metrics

Task: Given a search query, the goal is to identify and link the query to the most relevant entities in the category-specific product knowledge graph. This enables downstream applications such as query rewriting, recommendation systems, and search result enhancement. The challenge lies in mapping natural language queries with varying levels of specificity and ambiguity to structured product entities in the KG.

Evaluation Metrics: We employ standard evaluation metrics:

- **Precision**: $P = \frac{TP}{TP+FP}$ where TP are correctly linked query-entity pairs and FP are incorrectly linked pairs

Table 1
Performance comparison across product categories

	Sneaker			Smartphone			Wrist Watches		
Method	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Baseline	0.6259	**0.7752**	0.6926	0.6719	0.8104	0.7347	0.8147	0.7578	0.7852
E4EL	**0.8312**	0.7235	**0.7736**	**0.8608**	**0.9014**	**0.8807**	**0.8169**	**0.9119**	**0.8618**

- **Recall**: $R = \frac{TP}{TP+FN}$ where FN are query-entity pairs that should have been linked but were missed
- **F1-Score**: $F_1 = \frac{2 \cdot P \cdot R}{P+R}$ harmonic mean of precision and recall

5.3. Baseline Methods

We compare E4EL against the current production baseline, a low-latency string-based entity linking pipeline that performs n-gram matching followed by greedy selection based on entity popularity and span length.

The choice of baseline reflects the operational constraints of our deployment setting, as discussed in Section 2. Entity linking must operate under strict latency budgets (on the order of tens of milliseconds), at high throughput, and using CPU-only inference. Under these constraints, many existing academic approaches, such as transformer-based bi-encoder/cross-encoder pipelines or generative models—are not feasible for online serving due to their substantially higher inference cost.

In particular, transformer-based methods require significant computation at inference time, which, when executed on CPU-only infrastructure, typically results in latency that exceeds practical limits for synchronous query understanding. Consequently, these approaches are not considered deployable in our setting. Our evaluation therefore focuses on baselines that satisfy the same latency and infrastructure requirements as the production system, enabling a fair and meaningful comparison in a real-world deployment scenario.

This comparison also highlights the difference between non-semantic and graph-based approaches. The baseline relies on string matching and popularity heuristics, which do not capture relational dependencies between entities. In contrast, E4EL leverages knowledge graph structure to model entity compatibility and coherence. As a result, the evaluation reflects not only improvements in accuracy, but also the added value of incorporating semantic relationships into entity linking under real-world constraints.

5.4. Experimental Results

We evaluate the approach on three categories overall, and head entity types per category.

5.4.1. Overall Performance

Table 1 presents the comprehensive performance comparison against the string-based baseline across all three product categories, showing precision, recall, and F1-scores. The results show that the proposed approach significantly outperforms the baseline approach in across all categories.

5.4.2. Head Entities Performance Analysis

Table 2 shows the performance breakdown for the most searched entities in each category, demonstrating our model's effectiveness across different entity types.

Table 2
Performance comparison for head entities by product category

Category	Head Aspect	Baseline			E4EL		
		P	R	F1	P	R	F1
Sneaker	model	0.8853	0.7255	0.7975	**0.8884**	**0.7341**	**0.8039**
	brand	0.7467	0.7344	0.7405	**0.8815**	**0.9487**	**0.9395**
	us_shoe_size	0.8155	0.8989	0.8552	**0.8815**	**0.9487**	**0.9139**
Smartphone	model	0.6209	0.5910	0.6057	**0.8608**	**0.9014**	**0.8807**
	brand	0.8842	**0.9047**	0.8943	**0.9308**	0.8705	**0.8996**
	storage_capacity	0.9435	0.9891	0.9658	**0.9880**	**1.0000**	**0.9939**
Wristwatches	brand	**0.9006**	0.7834	0.8379	0.8880	**0.9904**	**0.9364**
	model	0.7814	0.6487	0.7089	**0.7890**	**0.9710**	**0.8706**
	case_size	0.9250	0.9135	0.9192	**0.9689**	**0.9629**	**0.9659**

5.4.3. Ablation Study

To understand the contribution of each component in our Double-Coherence CRF module, we conduct a systematic ablation by disabling individual modules and re-evaluating performance across all three product categories.

Our analysis reveals a clear hierarchy of contributions. The mention coherence module (soft mask z_i) provides the largest performance contribution: removing it causes the most substantial F1 degradation across all categories, as the model can no longer filter irrelevant span candidates and the number of false positives increases significantly. Without the soft mask, every candidate span contributes equally to the CRF score, overwhelming the disambiguation stage with noisy mention candidates. Entity coherence modeling (ψ_p) is the second most important component; without the GNN-derived entity embeddings capturing compatibility relationships in the KG, the model loses the ability to reason about which entity combinations are mutually consistent (e.g., distinguishing compatible brand-model pairs such as `brand/nike` with `model/air_max_90` from incompatible ones such as `brand/nike` with `model/galaxy_s24`), leading to a meaningful F1 drop. The relevance scoring function (ψ_u, Jaccard-based) contributes modestly but consistently: its removal degrades performance by limiting the model's ability to align mention spans to entity surface forms. Finally, the aggregation module ($MLP_{aggregation}$) provides the smallest marginal gain, suggesting that the individual unary and pairwise scores are already informative, and the learned combination provides incremental but consistent improvement. This hierarchy confirms that accurate mention boundary selection is the most critical factor for entity linking performance, while entity-level coherence via knowledge graph structure provides substantial complementary benefits.

5.4.4. Error Analysis

To better understand the strengths and limitations of E4EL, we conduct a qualitative error analysis across the evaluated categories. We identify two systematic patterns.

Successful disambiguation via entity coherence. E4EL correctly resolves ambiguous mentions by leveraging KG structure. For example, given "jordan 5 blue", E4EL links to `model/jordan_5` and `color/blue`, a mutually coherent set in the KG, while the baseline incorrectly maps "jordan" to `brand/jordan` and "5" to `size/5`.

Entity aspect-type ambiguity. A recurring error involves entities that exist under multiple aspect types. For instance, "nike air force" can resolve to `model/nike_air_force` or `product_line/nike_air_force`, where the correct assignment depends on domain conventions rather than textual or structural signals. Since the entity coherence module operates on graph-topological embeddings, it cannot easily distinguish aspect types that occupy similar KG neighborhoods. This reflects an underlying data quality challenge: normalizing entities across aspect types requires domain

expertise and is an active area of KG improvement.

5.4.5. Latency Analysis

A critical requirement for production deployment is real-time performance at scale. We evaluate the computational efficiency of E4EL framework and compare inference latencies with the string-based baseline.

Our framework achieves an average inference latency of 32ms per query, at 10K requests per second, compared to 20ms for the baseline system. While this adds approximately 12 ms of overhead, the resulting latency remains substantially lower than retrieval- and generation-based approaches and is well within acceptable bounds for eBay's search infrastructure. The modest latency overhead is justified by substantial accuracy improvements, resulting in a significantly improved user experience.

6. Deployment and Operational Insights

Deploying E4EL in a high-throughput e-commerce search system provides practical insights beyond offline evaluation.

Latency–accuracy trade-offs. In production, model quality must be evaluated jointly with latency. Transformer-based approaches, while accurate, incur substantial inference cost and are not suitable for CPU-only, latency-sensitive settings. E4EL achieves an average latency of 32ms while improving F1 over the production baseline, demonstrating that deployable systems must operate on the trade-off frontier between accuracy and latency.

Structured inference efficiency. CRF-based structured prediction provides a strong balance between performance and efficiency by jointly modeling mention selection and entity coherence. This improves robustness to ambiguity without introducing significant computational overhead.

Zero-shot capability. Frequent inventory updates make retraining impractical. By decoupling entity representation from inference, E4EL enables immediate incorporation of new entities through offline embedding updates.

Benefits and challenges of knowledge graphs. The knowledge graph enables modeling of entity compatibility beyond surface features, supports zero-shot generalization, and can be reused across downstream applications. The same KG infrastructure that powers E4EL also supports KG-enhanced retrieval augmented generation [6], query reformulation, and personalized recommendation in production, demonstrating the reusability of the graph across the application ecosystem. However, the knowledge graph introduces challenges in maintaining graph consistency as the product inventory evolves, updating entity embeddings to reflect structural changes, and integrating graph-based signals into latency-constrained systems without exceeding online serving budgets.

Deployment readiness and adoption context. E4EL is integrated into a deployment-equivalent environment that mirrors production interfaces, traffic patterns, and latency constraints, and is ready for online A/B testing. Its serving architecture requires no new infrastructure: GNN-based entity embeddings are pre-computed offline and served through the existing graph database, while the CRF inference module runs within the same CPU-only service tier as the current production baseline. The broader Product Knowledge Graph ecosystem that E4EL builds upon has been deployed in production for several years and is actively consumed by multiple teams across eBay for query reformulation, retrieval enhancement, personalized recommendation, and listing assistance [5]. These applications have been A/B tested with millions of users, resulting in statistically significant improvements in search engagement and purchase rates. This sustained, cross-team adoption of the underlying KG infrastructure demonstrates both the operational maturity of the platform and the incremental nature of E4EL's integration path from evaluation to production serving.

System constraints and generalizability. Design choices are shaped by CPU-only inference, strict latency budgets, and integration requirements. While evaluated in e-commerce, the approach applies to domains with structured entity relationships and sparse context, provided a compatibility graph is available.

7. Conclusion

We presented E4EL, an end-to-end entity linking framework designed for real-time e-commerce search under strict latency and scalability constraints. By combining high-recall candidate generation with a lightweight Double-Coherence CRF and GNN-derived entity embeddings, the approach addresses key challenges of zero-shot generalization, dynamic inventory, and low-latency inference. E4EL achieves significant improvements over the production baseline (e.g., +11.7% F1 in sneakers and +9.8% in wrist watches) while maintaining an average latency of 32ms in a deployment-equivalent environment. This demonstrates that knowledge graph–aware modeling can improve disambiguation quality without violating real-time constraints, making it suitable for large-scale industrial systems.

The main contributions of this work are threefold: (1) a practical entity linking solution that balances accuracy and latency for production deployment, (2) a novel double-coherence formulation that jointly models mention selection and entity compatibility, and (3) an effective integration of knowledge graph structure via lightweight embeddings that supports zero-shot linking without retraining. Overall, our results show that combining structured prediction with graph-based representations provides a strong and efficient alternative to transformer-based approaches in latency-sensitive settings. More broadly, this work highlights the practical value of knowledge graphs for enabling robust, deployable entity linking in real-world applications. Future work includes extending coherence modeling across categories and incorporating user-level signals to further improve disambiguation performance.

Declaration on Generative AI

A large language model was used during the data annotation process to generate initial candidate ground-truth entity sets for training data construction (Section 4.3.1). Specifically, given the entity candidates produced by the high-recall candidate generator for each query, an LLM was used to suggest the most coherent entity set. All LLM-suggested annotations were subsequently validated by human annotators before inclusion in the training data. Generative AI tools were additionally used to assist with refining the prose of this manuscript. The authors reviewed all outputs and take full responsibility for the content of this work.

References

[1] L. Wu, F. Petroni, M. Josifoski, S. Riedel, L. Zettlemoyer, Scalable zero-shot entity linking with dense entity retrieval, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6397–6407. URL: https://aclanthology.org/2020.emnlp-main.519/. doi:10.18653/v1/2020.emnlp-main.519.

[2] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, A. Pierleoni, ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking, in: A. Loukina, R. Gangadharaiah, B. Min (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 2022, pp. 209–220. URL: https://aclanthology.org/2022.naacl-industry.24/. doi:10.18653/v1/2022.naacl-industry.24.

[3] P. Ristoski, Z. Lin, Q. Zhou, Kg-zeshel: knowledge graph-enhanced zero-shot entity linking, in: Proceedings of the 11th Knowledge Capture Conference, 2021, pp. 49–56.

[4] N. D. Cao, G. Izacard, S. Riedel, F. Petroni, Autoregressive entity retrieval, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=5k8F6UU39V.

[5] P. Ristoski, S. Kandasamy, A. Matiushkin, S. Kamath, Q. Zhou, Wisdom of the sellers: Mining seller data for ecommerce knowledge graph generation, in: European Semantic Web Conference, Springer, 2023, pp. 195–199.

[6] Z. Xu, Z. Shen, Q. Zhou, P. Ristoski, Knowledge graph-enhanced retrieval augmented generation for e-commerce, in: Proceedings of the Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs (RAGE-KG 2024), 2024.
[7] Q. Zhou, Z. Wu, J. Degenhardt, E. Hart, P. Ristoski, A. Mandal, J. Netzloff, A. Mandalam, Leveraging knowledge graph and deepner to improve uom handling in search, in: ISWC (Posters/Demos/Industry), 2021.
[8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: K. Knight, A. Nenkova, O. Rambow (Eds.), Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 260–270. URL: https://aclanthology.org/N16-1030/. doi:10.18653/v1/N16-1030.
[9] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in: K. Erk, N. A. Smith (Eds.), Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1064–1074. URL: https://aclanthology.org/P16-1101/. doi:10.18653/v1/P16-1101.
[10] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. H. Torr, Conditional random fields as recurrent neural networks, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1529–1537.
[11] O.-E. Ganea, T. Hofmann, Deep joint entity disambiguation with local neural attention, in: M. Palmer, R. Hwa, S. Riedel (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2619–2629. URL: https://aclanthology.org/D17-1277/. doi:10.18653/v1/D17-1277.
[12] P. Le, I. Titov, Improving entity linking by modeling latent relations between mentions, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1595–1604. URL: https://aclanthology.org/P18-1148/. doi:10.18653/v1/P18-1148.
[13] T. Kipf, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).
[14] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Advances in neural information processing systems 30 (2017).
[15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European semantic web conference, Springer, 2018, pp. 593–607.
[16] Y. Cao, L. Hou, J. Li, Z. Liu, Neural collective entity linking, in: E. M. Bender, L. Derczynski, P. Isabelle (Eds.), Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 675–686. URL: https://aclanthology.org/C18-1057/.
[17] Y. Ding, A. Poudel, Q. Zeng, T. Weninger, B. Veeramani, S. Bhattacharya, Entgpt: Entity linking with generative large language models, arXiv preprint arXiv:2402.06738 (2024).
[18] A. Xin, Y. Qi, Z. Yao, F. Zhu, K. Zeng, X. Bin, L. Hou, J. Li, Llmael: Large language models are good context augmenters for entity linking, arXiv preprint arXiv:2407.04020 (2024).