Joint Information Retrieval and Recommendation: a Reproducibility Study*

Simone Merlo[1], Guglielmo Faggioli[1] and Nicola Ferro[1]

[1]University of Padua, Padua, Italy

Abstract

Information Retrieval (IR) and Recommender Systems (RS) represent the core components in the information access scenario. These two categories of systems are traditionally developed in isolation and have a very limited interaction. However, since the nineties it was clear that there were significant connections between IR and RS and in recent times systems performing retrieval and recommendation jointly have been created. This contributed to showing that developing joint IR and RS systems allows to improve the performance of both tasks. The current state-of-the-art in the joint IR and RS field is represented by the Unified Information Access (UIA) framework. Driven by the importance of reproducibility, in this work, we discuss the reproducibility, replicability and generalizability of UIA. First, we analyse the reproducibility degree of UIA. Then, we focus on its replicability by studying its behaviour on a public dataset. Finally, we explore its generalizability by altering the data processing and training algorithms. The obtained results show that the performance of UIA and, in general, of joint IR and RS systems, may strongly depend on the dataset used for the training and evaluation and that its stability may vary depending on the task.

Keywords

Information Retrieval, Recommender Systems, Large Language Models

1. Introduction

Information Retrieval (IR) systems and Recommender Systems (RS) are traditionally thought as independent systems, event if their results are frequently merged to provide users with a comprehensive answer to their needs. Nonetheless, Belkin and Croft [2] consider IR and RS as "two sides of the same coin". Indeed, both systems are mainly concerned with retrieving the most suitable piece of information — documents for IR or items for RS — in a collection according to a request. However, some differences are still present and they mainly concern the input, which represented by a textual query in IR and by an item or a set of user's historical interactions in RS. In recent times, systems performing IR and recommendation jointly started to be developed. Indeed, Si et al. [3] and Zamani and Croft [4, 5] showed that combining IR and RS models allows for improved performance by exploiting the knowledge held by a model to enhance the other. Moreover, two major efforts are SRJGraph, proposed by Zhao et al. [6], and the Unified Information Access (UIA) framework, developed by Zeng et al. [7], which represents the state-of-the-art. Anyway, the joint modeling of IR and RS is still in its early stages.

Reproducibility is a fundamental aspect of both IR and RS and it poses many challenges [8, 9, 10]. For this reason and given the recent interest in joint IR and RS we analyse the UIA framework [7] based on three axes[1]: reproducibility (i.e., different team, same experimental setup), replicability (i.e., different team, different experimental setup), and generalizability (i.e., different team, different experimental setup, different task). In particular, the main goal is articulated in three research questions:

- **RQ1 - Reproducibility**: is the performance of UIA, reported in [7], reproducible?
- **RQ2 - Replicability**: is the performance of UIA replicable on a publicly available dataset?

SEBD 2025: 33rd Symposium On Advanced Database Systems, June 16–19, 2025, Ischia, Italy

*This work is an extended abstract of [1].

simone.merlo@phd.unipd.it (S. Merlo); guglielmo.faggioli@unipd.it (G. Faggioli); ferro@dei.unipd.it (N. Ferro)
https://dei.unipd.it/~merlosimon (S. Merlo); https://dei.unipd.it/~faggioli (G. Faggioli); https://dei.unipd.it/~ferro (N. Ferro)
0009-0003-8003-4795 (S. Merlo); 0000-0002-5070-2049 (G. Faggioli); 0000-0001-9219-6239 (N. Ferro)

[1]https://www.acm.org/publications/policies/artifact-review-and-badging-current

- **RQ3 - Generalizability**: is the performance of UIA stable when using alternative approaches which are less computationally demanding and/or involve different data processing methods?

UIA was chosen since, by representing the state-of-the-art in the joint IR and RS field, its architecture may become a base to develop new systems. Moreover, UIA was originally trained and evaluated using both a private (Lowe's) and a publicly available (Amazon ESCI) datasets but the bulk of the experiments was conducted only on the private one, which also enabled more functionalities.

In this paper we discuss the results of our empirical evaluation[2] which showed that UIA can be reproduced and that the robustness and effectiveness of UIA depends on different factors including the training data and the training process. This work represents an extended abstract of a previous ECIR submission, the full version is available at [1].

The remainder of this work is organized as follows: in Section 2 we provide an overview of UIA; in Section 3 we explain how we reproduced, replicated and generalized the framework; in Section 4 we report and discuss the results obtained.

2. Highlights of the Reproduced Approach

In this section we provide an overview of the UIA framework and of the Amazon ESCI dataset, which is used for the reproducibility, replicability and generalizability study.

2.1. The UIA Framework

Zeng et al. [7] summarize an interaction between the user and the UIA framework (Figure 1) with three elements: an information access request $\mathcal{R}$, a task label (access functionality in [7]) $\mathcal{F}$, and a candidate information item $\mathcal{I}$. UIA supports three hybrid RS-IR tasks (functionalities in [7]) $\mathcal{F}$: i) Keyword Search (KS) where a short textual query is used to retrieve the most relevant items; ii) Query By Example (QBE) where an input item is used to retrieve other similar items; and iii) Complementary Item Recommendation (CIR) where an input item is used to retrieve items that "can be used together" (*i.e.*, complementary). The information access request $\mathcal{R}$ is task-dependent and corresponds to a keyword query (KS) or an item (QBE or CIR). Finally, the candidate information item $\mathcal{I}$, is a textual representation of the item (*i.e.*, its title, in the Amazon ESCI dataset) for which the system must estimate the relevance to $\mathcal{R}$. Thus, given a task $\mathcal{F}$ and a request $\mathcal{R}$, the goal of UIA, parametrized by θ, is ranking all the items $\mathcal{I}$ in the catalogue based on a relevance score $s = f(\mathcal{R}, \mathcal{F}, \mathcal{I}; \theta)$. To accomplish this, UIA relies on a bi-encoder architecture. Specifically, it employs a request encoder $\mathbf{E}_{\mathcal{R}}$ and an item encoder $\mathbf{E}_{\mathcal{I}}$ to embed a request $\mathcal{R}$ (jointly with the task label $\mathcal{F}$) and an item $\mathcal{I}$ within a latent space. More in detail, $\mathcal{R}$ is encoded as $\vec{\mathbf{R}} = \mathbf{E}_{\mathcal{R}}(\texttt{[CLS]}\ \mathcal{R}\ \texttt{[SEP]}\ \mathcal{F}\ \texttt{[SEP]})$, where $\mathcal{F}$ is the label of the task associated to the request, while [CLS] and [SEP] are the "class" and "separator" tokens, respectively. Similarly $\mathcal{I}$ is encoded as $\vec{\mathbf{I}} = \mathbf{E}_{\mathcal{I}}(\texttt{[CLS]}\ \mathcal{I}\ \texttt{[SEP]})$. The final representation is the embedding of the [CLS] token, as typical in this context [11, 12, 13]. Both $\mathbf{E}_{\mathcal{R}}$ and $\mathbf{E}_{\mathcal{I}}$ employ the BERT [14] model to encode their input. Finally, the score of $\mathcal{I}$ with respect to $\mathcal{R}$ is computed as $s = \vec{\mathbf{R}} \cdot \vec{\mathbf{I}}$. UIA is trained by minimizing a cross-entropy loss function. Each training instance is a tuple $(\mathcal{R}, \mathcal{F}, \mathcal{I}^+, \mathcal{I}^-)$, where $\mathcal{I}^+$ and $\mathcal{I}^-$ represent a positive and a negative example respectively. The negative examples, not available in the original dataset, are obtained with a two-phase negative sampling procedure. The first phase (*Phase 1*) samples the negatives among the items retrieved by BM25 [15] in response to each request. The second phase (*Phase 2*) employs the model trained using the data of Phase 1 to embed the items in the space and samples the negatives among the nearest neighbours of each item. The training procedure involves also the usage of in-batch negatives and mini-batches.

Zeng et al. [7] consider also a second training pipeline to handle users' data and personalize the output. Such pipeline requires accessing user's personal data (*i.e.*, previous interactions with the system and preferences). However, we focus exclusively on non-personalized data (*i.e.*, the Amazon ESCI Dataset), thus we describe only the non-personalized part of the pipeline.

[2]Code available at:https://anonymous.4open.science/r/UIAReproRepliGen-5CEE

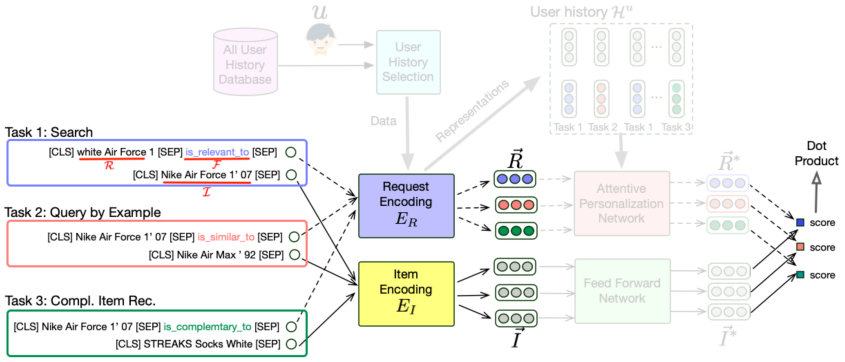


Figure 1: UIA framework architecture. Grayed-out areas are those concerning personalization, that we did not experiment with. Figure taken from [7].

2.2. The Amazon ESCI Dataset

Zeng et al. [7] trained and evaluated UIA on two datasets: the Lowe's dataset and the Amazon ESCI dataset [16]. The former is private and contains user data to enable personalization, the latter is public but does not contain user data and thus does not allow training/testing the personalization module. Due to its public availability and to the lack of public, joint IR and RS datasets, we focus exclusively on the Amazon ESCI dataset. The Amazon ESCI dataset [16] was released in the context of the KDD Cup 2022[3] Amazon ESCI challenge and it is a large, multilingual dataset of difficult Amazon search queries and results. In line with [7], we consider the product catalogue and the training data used for the Task 2 of the challenge. The training data contains triplets (query, item, label) where the label is one among: "Exact", *i.e.*, the item is an exact match for the query; "Substitute", *i.e.*, the item is related to the query but not a match; "Complement", *i.e.*, the item is not relevant to the query but can complement a relevant item; and "Irrelevant". The ESCI dataset contains only textual queries, thus is unsuitable for QBE and CIR, therefore, in line with [7], we split the full dataset into three separate datasets, one for each task. Specifically, we call Q the set of all the requests (queries), and $I_E(q)$, $I_S(q)$, and $I_C(q)$ the sets of items labelled "Exact", "Substitute", and "Complementary" for query q, respectively. The three task-specific datasets are defined as follows: (1) KS: $\{(q,i) : \forall q \in Q \wedge i \in I_E(q)\}$, (2) QBE: $\{(i_1,i_2) : \forall q \in Q \wedge i_1 \in I_E(q) \wedge i_2 \in I_S(q)\}$, and (3) CIR: $\{(i_1,i_2) : \forall q \in Q \wedge i_1 \in I_E(q) \wedge i_2 \in I_C(q)\}$. Following [7], we further split each dataset into training (80%), validation (10%), and test (10%) sets. The three datasets are used jointly during the training phase while, for evaluation, the performance is measured separately on each test set.

3. Reproduction and Experimental Methodologies

In this section, we detail the experiment to assess the reproducibility of UIA (RQ1), we then introduce the analyses done to determine its replicability (RQ2) and conclude with the tests carried out to gauge UIA generalizability (RQ3).

3.1. RQ1: Reproducibility

To reproduce UIA, we employed only publicly available datasets and the original code[4]. We operated independently on whether the original developers were available to share with us their knowledge, to ensure unbiased results and put ourselves in the most challenging reproducibility conditions. We report here the challenges we identified in reproducing UIA and the solutions we used to address them.

[3]KDD Cup 2022: https://amazonkddcup.github.io/
[4]https://github.com/HansiZeng/UIA

Second sample of the relevant items. While inspecting the available code base, we observed that, after the dataset splitting (Section 2.2), a second sampling is executed. Specifically, for QBE and CIR, for every unique query item (*i.e.*, i_1 in Section 2.2), 5 random relevant items are sampled (*i.e.*, 5 instances are added to the dataset). Similarly, for KS, each query is associated with only 10 relevant items. We ascribe this difference between the original paper and the code to efficiency reasons. Moreover, this second sampling prevents excessively large datasets and weighting too much queries which are too popular or generic items. We maintain this implementation choice to ensure reproducibility.

Negative sampling procedure. The available code base reports some differences from the procedure described in Section 2.1 concerning the *Phase 1* negative sampling for the QBE task. Indeed, for QBE the negatives are randomly sampled among all the items. We modified the code to sample the negatives from the items retrieved by BM25 also for QBE.

Another difference concerns the KS dataset. In detail, when sampling the negative examples for the KS task during *Phase 1*, for each pair request-item in the dataset, the negative is sampled from the items similar (i.e., labelled "Substitute") to the one considered as positive, if present, else from the items complementary to the one considered as positive, if present, else the negative is sampled using the request and BM25. We preserved this aspect of the code.

Computational Resources. Due to limited computational resources — especially concerning GPU memory — we reduced the batch size from 384 (used in [7]) to 48 (-86%) and we set the number of epochs to 24 instead of 48. Other hyperparameters, such as learning rate ($7e^{-6}$) and the number of warmup iterations (4,000), were left unchanged compared to the original paper.

3.1.1. Phase 1 Only

The double-phase training is computationally expensive, doubling the training time and cost (including the environmental impact). Thus, we evaluate UIA after a single training phase. While we reasonably expect a decrease in terms of performance, we are interested in assessing whether this represents an acceptable trade-off between effectiveness and efficiency. If this applies, UIA could be employed in resource-constrained environments, with a limited cost and environmental impact.

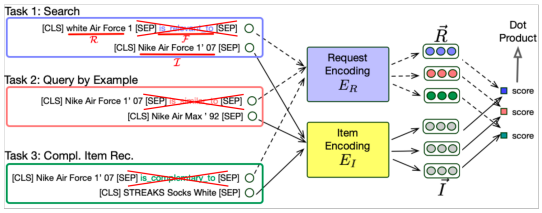
3.2. RQ2: Replicability

Zeng et al. [7] focus mostly on the Lowe's private dataset and only some of the analyses are carried out on the Amazon ESCI dataset. Thus, concerning replicability, we are interested in extending the analysis of UIA to the Amazon ESCI dataset by replicating on it the experiments done by Zeng et al. [7] only on the Lowe's dataset.

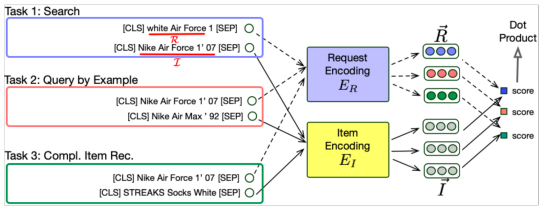
3.2.1. No Task Label (w/o $\mathcal{F}$)

We seek to examine the role of the task label $\mathcal{F}$ in UIA to understand if the framework is able to recognize that there are three different tasks or if it only learns from the huge amount of training data. To do this we modify UIA by removing $\mathcal{F}$ (this version of the framework is addressed as "w/o $\mathcal{F}$"). This allows to consider the training data related to the different tasks as belonging to a unique training set.

In detail, given its relatively large training data, the interchangeability of its input and output (*i.e.*, both items for QBE and CIR), and the similar nature of the tasks, we are interested in determining how important $\mathcal{F}$ is in correctly matching items to items. Furthermore, while KS uses queries as requests, QBE and CIR use items. We are thus interested in verifying if this aspect is already sufficient to diversify the two classes of tasks. Removing the task label $\mathcal{F}$, in practical terms, corresponds to modifying $\mathbf{E}_{\mathcal{R}}$ into $\mathbf{E}'_{\mathcal{R}}$ s.t. $\vec{\mathbf{R}}' = \mathbf{E}'_{\mathcal{R}}(\texttt{[CLS]}\ \mathcal{R}\ \texttt{[SEP]})$. Thus, for the candidate item $\mathcal{I}_i$, the new score can be computed as $s'_i = \vec{\mathbf{R}}' \cdot \vec{\mathbf{I}}_i$. In Figure 2 we highlight the portions that are removed (Figure 2a) and we show the new architecture without the task label (Figure 2b).



(a) Removing task label from UIA architecture.



(b) UIA architecture without task label.

Figure 2: UIA without the task label $\mathcal{F}$.

3.2.2. Isolated Tasks

Training and evaluating the framework on the tasks in isolation is equivalent to optimize and evaluate 3 separate instances of UIA, each one for each task. Zeng et al. [7] showed that when the Lowe's dataset is used, UIA benefits from the joint training. The employment of the Amazon ESCI dataset, though, implies deep changes in the architecture of the framework (*i.e.*, the personalization part is removed). Thus, we want to understand if UIA still benefits from joint training when the Amazon ESCI dataset is used and, therefore, when the personalization components are removed. To do this we train the framework on the tasks in isolation. For efficiency reasons and in light of the results achieved by the *Phase 1 Only* experiment (Section 3.1) we consider a single-phase training. Therefore, the results should be compared with those obtained for the experiment *Phase 1 Only*.

3.3. RQ3: Generalizability

We describe here the experimental methodology we adopt to test the generalizability of UIA, i.e., its resilience to major changes to its training procedure and to the training data.

3.3.1. Half QBE

Inspecting the generated datasets reveals that (after sampling, Section 3.1) the training set for QBE (1.07M tuples) is more than twice the KS one (452k tuples) and six times larger than the CIR one (184k tuples). While not explicitly mentioned in [7], this characteristic was also observed by Zeng et al.. Indeed, in the provided repository, some portions of code use only half of the QBE dataset. These results were not reported or explicitly mentioned in [7]. To assess the generalizability of the approach, we test the hypothesis that reducing the amount (i.e., halving) of data used for the QBE task does not impact severely on the final performance.

3.3.2. Early Split

The UIA task can be considered an example of "Knowledge Graph Completion". The idea underlying this task consists of predicting if, given a relation r and two entities h and t, the head entity h is in relation r with the tail entity t. For UIA, the head entity $\mathcal{R}$ is either a query or an item, the relation $\mathcal{F}$ is one among KS, QBE, or CIR, and the tail entity $\mathcal{I}$ is an item. The procedure to split the collection into training, validation, and test set adopted by Zeng et al. [7], consists of randomly partitioning all the possible triplets $(\mathcal{R}, \mathcal{F}, \mathcal{I})$ into the three sets. While this is frequent in the "Knowledge Graph Completion" domain [17, 18, 19, 20, 21, 22], it also is criticized [23, 24]. In particular, Akrami et al. [23], criticizes the so-called "Cartesian product relations". These relations are such that given a set of subjects and objects, the relation is valid for all the cartesian pairs between subjects and objects. If part of these pairs ends in the training set and part ends in the test, this inflates the performance of the knowledge graph completion algorithm.This occurs in the dataset used for UIA. Indeed, given a query q of the Amazon ESCI dataset, its "Exact" items are related to all the corresponding "Substitute" and "Complementary" items. We propose to modify this splitting procedure by dividing Q (the set of the queries) into training, validation, and test sets and generating the triples only afterwards, using the

procedure proposed by Zeng et al. [7] (described in Section 2.2). This ensures that all the information regarding a certain query is contained within same partition. This also appears natural from a "temporal" standpoint. Indeed, when a user issues a query it is possible to collect the training data up to that instant and the system is unaware of the next user's query (i.e., the test). Thus, employing information associated to training queries to test the model would correspond to predicting the past. Given this new version of the datasets, we retrain the model and test its performance. For efficiency reasons and in light of the results achieved by the *Phase 1 Only* experiment (Section 3.1) we consider a single-phase training. Therefore, the results should be compared with those obtained for the experiment *Phase 1 Only*.

4. Experimental Results

Here we discuss the outcomes of the reproducibility, replicability and generalizability experiments introduced in Section 3. We evaluate our results considering MRR@10, nDCG@10 and Recall@50, in line with [7]. Table 1 reports the original performance of UIA (first row) along with the one of our experiments.

Table 1
Reproducibility, replicability and generalizability results for the Keyword Search (KS), Query By Example (QBE) and Complementary Item Recommendation (CIR) tasks.

		KS			QBE			CIR		
	Model	**MRR**	**nDCG**	**Recall**	**MRR**	**nDCG**	**Recall**	**MRR**	**nDCG**	**Recall**
original	**UIA**	0.532	0.360	0.533	0.251	0.199	0.543	0.490	0.493	0.868
RQ1 (repr.)	**UIA**	0.491	0.327	0.484	0.442	0.374	0.673	0.463	0.459	0.833
	Phase1Only	0.477	0.313	0.461	0.294	0.227	0.531	0.361	0.353	0.760
RQ2 (repl.)	**w/o $\mathcal{F}$**	0.480	0.311	0.490	0.338	0.287	0.637	0.283	0.292	0.721
	w/o $\mathcal{F}$ (Phase1Only)	0.441	0.280	0.419	0.247	0.185	0.472	0.181	0.175	0.524
	IsolatedTasks	0.506	0.340	0.493	0.324	0.257	0.561	0.414	0.412	0.779
RQ3 (gene.)	**HalfQBE**	0.510	0.341	0.504	0.316	0.250	0.561	0.455	0.452	0.838
	HalfQBE (Phase1Only)	0.498	0.335	0.491	0.053	0.039	0.232	0.378	0.370	0.775
	Early Split	0.467	0.306	0.451	0.053	0.034	0.129	0.041	0.037	0.147

4.1. RQ1: Reproducibility Results

The reproducibility results are reported in the second row of Table 1. The obtained performance is really close to the original one for KS (-0.041 MRR, -0.033 nDCG and -0.049 Recall) and CIR (-0.027 MRR, -0.034 nDCG and -0.035 Recall). These results appear satisfactory, considering that we were forced to reduce the batch size and epochs, due to limited computing capabilities. In this regard, UIA achieves satisfactory performance even under stronger resource constraints. Differently, for QBE our results are significantly larger, from 24% to 88% (+0.191 MRR, +0.175 nDCG and +0.130 Recall), than those reported in [7]. We attribute this phenomenon to the changes in the Phase 1 negative sampling (from random to BM25, see in Section 3.1). We hypothesise that the results reported in [7] represent a lower bound of the actual performance UIA can achieve on the QBE task.

4.1.1. Phase 1 Only

Considering the model trained with a single phase (third line of Table 1), the performance drops. The magnitude of the drop depends on the task. For KS it is minor (-0.014 MRR, -0.014 nDCG, -0.023 Recall), suggesting that, for this task, the second training phase has a limited impact. For CIR the drop is larger (-0.102 MRR, -0.099 nDCG, -0.073 Recall). QBE, instead, is the most harmed task (-0.148 MRR, -0.147 nDCG, -0142 Recall). This suggests the importance of the hard negatives and additional training time for the two most RS oriented tasks. The drop in performance is not negligible but so are the computational resources saved: from 240 hours of computation to 120.

4.2. RQ2: Replicability Results

4.2.1. No Task Label (w/o $\mathcal{F}$)

The results achieved by removing the task label are reported in the " w/o $\mathcal{F}$ " row of Table 1. UIA behaves consistently on the Amazon ESCI dataset w.r.t. the ablation study on the Lowe's dataset reported in [7]. Again, the KS task appears as the most stable (-0.011 MRR, -0.016 nDCG, and +0.006 Recall). The most impacted task, instead, is CIR with a loss of 37% in performance for MRR and nDCG (-0.180 MRR, -0.167 nDCG, and -0.112 Recall). This indicates that, when there is no distinction between the tasks, the model is still able to operate on KS, while being less performing for QBE and CIR. A possible explanation lies in the differences in term distribution between queries and items, used as input for KS and QBE and CIR tasks. Moreover, the difference in the training sets sizes of QBE and CIR may be the cause of their difference in performance loss. Indeed, the QBE dataset is much larger than the CIR (580%), thus, during the training phase, it is "less harmful" for the model to optimize for the QBE task: this reflects on the test performance, where the QBE task is handled better.

The "w/o $\mathcal{F}$ (Phase1Only)" row of Table 1 reports the results achieved removing the task information and training the framework only according to *Phase 1*. These must be compared with the one of the "Phase1Only" experiment. By looking at the results we can conclude that, for this experiment, the framework behaves in the same way also when performing one training phase.

4.2.2. Isolated Tasks

The results achieved for the three instances of UIA, each optimized on a single task, are grouped in the row "IsolatedTasks" row of Table 1. The obtained performance shows that UIA performs better when trained on single tasks than when jointly optimized, if the Amazon ESCI dataset is exploited. The KS task appears as the most stable (+0.029 MRR, +0.027 nDCG, and +0.032 Recall compared to Phase1Only). The QBE task has the second-biggest increase (+0.030 MRR, +0.030 nDCG, and +0.030 Recall). Finally, CIR is the task that has the greatest advantage (+0.053 MRR, +0.059 nDCG, and +0.019 Recall).

This behaviour does not align with the results on the Lowe's dataset, discussed in [7] and in the other studies about joint IR and RS [4, 5, 6]. Indeed, the Amazon ESCI dataset, differently from the Lowe's, does not contain user data, leading to major differences also in UIA's architecture. In detail, when Amazon ESCI is employed, the personalization components must be removed from the framework. Thus, these outcomes reveal how the benefits derived from the joint training may depend on both the dataset, the data processing pipelines and architecture of the framework employed.

4.3. RQ3: Generalizabilty Results

For some generalizability experiments, as perviously stated, we adopt a more ethical approach towards IR research [25, 26, 27], training UIA with a single phase.

4.3.1. Half QBE

Halving the QBE training data highlights three patterns in the behaviour of UIA ("HalfQBE" row of Table 1): i) For KS the performance increases (+0.019 MRR, +0.014 nDCG, and +0.020 Recall compared to the reproduced UIA), suggesting that aligning the size of the datasets allows UIA to grasp more knowledge from the KS instances. ii) For QBE the performance decreases (-0.126 MRR, -0.124 nDCG, -0.112 Recall) due to the reduction of its training data. iii) For CIR the performance has minor changes (-0.008 MRR, -0.007 nDCG, +0.005 Recall), highlighting that CIR training phase is not affected by the training data used for the QBE. This behaviour stems from the semantics of the tasks. Indeed, the learning of KS and QBE is strongly correlated since they both aim to retrieve items "similar" to the input (query or item). Thus, the excessive amount of QBE data may overshadow KS. CIR, instead, expects as output an item that is explicitly not similar, thus its training is likely disentangled from the other tasks.

The "HalfQBE (Phase1Only)" row of Table 1 reports the results achieved using half of the data for QBE and training UIA only according to *Phase 1*. These must be compared with the ones of the "Phase1Only"

experiment. By looking at the results we can notice that, for KS nothing changes, for QBE the gap in performance grows, while for CIR the slight decrease turns to a small increase in performance. However, the same considerations made when performing both training phases apply.

4.3.2. Early Split

When we split the Amazon ESCI queries into training and test set before constructing the datasets used to train and test the framework, we observe negligible differences in performance (row "Early Split" of Table 1), compared to "Phase1Only", for KS. This stems from the fact that the KS dataset is obtained from Amazon ESCI by selecting the appropriate entries, without processing (differently to QBE and CIR), since Amazon ESCI is an IR dataset based on real world data. Thus, for KS, the independence from the preprocessing pipeline leads to a more stable performance. Differently, when the proposed processing pipeline is used, the performance for QBE and CIR drastically drops, highlighting that there exists scenarios in which UIA achieves unsatisfactory results. For example, if the model lacks prior knowledge of an item and this is used to query the system, then UIA will be bound to fail. Moreover, this provides valuable insights on how to test/train of this class of models. Indeed, it would be more informative to report results when the train-test split occurs both at a tuple level (as done in [7]) and at a query level (as proposed here), to obtain the complementary information of what would happen if the model was not able to learn from highly similar items – if not the item itself –, or from the item in relation to different ones. Finally, this should encourage the IR and RS research community, inspired by previous work on "knowledge graph completion" evaluation, to investigate, develop and adopt adequate evaluation protocols that effectively address corner cases and the various sides of the task.

5. Conclusions and Future Works

In this work we described the architecture of UIA and how the publicly available Amazon ESCI dataset is processed and employed to optimize it. Moreover, our experiments allowed to show that it is possible to reproduce the performance of UIA on the Amazon ESCI dataset. Differently, UIA's behaviour is not fully replicable, primarily because, when the Amazon ESCI dataset is used, the framework does not benefit from joint training. By generalizing the framework, we also discovered that the dataset used and the way in which it is manipulated have non-negligible consequences on the performance of UIA. Finally, our empirical results show that, when the Amazon ESCI dataset is used, the KS task appears to be the most robust while the recommendation tasks are more vulnerable.

Future work will focus on studying, employing and enhancing the benefits derived from the joint training. This includes understanding how the novelties introduced with UIA can be reused to develop innovative and robust systems. Moreover, we will continue our work towards the analysis and generalization of this framework. For this purpose, we will try to find (or create) and exploit publicly available datasets that include user data and can be adapted to fit in "joint retrieval and recommendation".

Acknowledgments

This work has received support from CAMEO, PRIN 2022 n. 2022ZLL7MW.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] S. Merlo, G. Faggioli, N. Ferro, A reproducibility study for joint information retrieval and recommendation in product search, in: Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part IV, Springer-Verlag, Berlin, Heidelberg, 2025, p. 130–145. URL: https://doi.org/10.1007/978-3-031-88717-8_10. doi:10.1007/978-3-031-88717-8_10.

[2] N. J. Belkin, W. B. Croft, Information filtering and information retrieval: Two sides of the same coin?, Commun. ACM 35 (1992) 29–38. URL: https://doi.org/10.1145/138859.138861. doi:10.1145/138859.138861.

[3] Z. Si, Z. Sun, X. Zhang, J. Xu, X. Zang, Y. Song, K. Gai, J. Wen, When search meets recommendation: Learning disentangled search representation for recommendation, in: H. Chen, W. E. Duh, H. Huang, M. P. Kato, J. Mothe, B. Poblete (Eds.), Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, ACM, 2023, pp. 1313–1323. URL: https://doi.org/10.1145/3539618.3591786. doi:10.1145/3539618.3591786.

[4] H. Zamani, W. B. Croft, Joint modeling and optimization of search and recommendation, in: O. Alonso, G. Silvello (Eds.), Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems, Bertinoro, Italy, August 28-31, 2018, volume 2167 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018, pp. 36–41. URL: https://ceur-ws.org/Vol-2167/paper2.pdf.

[5] H. Zamani, W. B. Croft, Learning a joint search and recommendation model from user-item interactions, in: J. Caverlee, X. B. Hu, M. Lalmas, W. Wang (Eds.), WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020, ACM, 2020, pp. 717–725. URL: https://doi.org/10.1145/3336191.3371818. doi:10.1145/3336191.3371818.

[6] K. Zhao, Y. Zheng, T. Zhuang, X. Li, X. Zeng, Joint learning of e-commerce search and recommendation with a unified graph neural network, in: K. S. Candan, H. Liu, L. Akoglu, X. L. Dong, J. Tang (Eds.), WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022, ACM, 2022, pp. 1461–1469. URL: https://doi.org/10.1145/3488560.3498414. doi:10.1145/3488560.3498414.

[7] H. Zeng, S. Kallumadi, Z. Alibadi, R. F. Nogueira, H. Zamani, A personalized dense retrieval framework for unified information access, in: H. Chen, W. E. Duh, H. Huang, M. P. Kato, J. Mothe, B. Poblete (Eds.), Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, ACM, 2023, pp. 121–130. URL: https://doi.org/10.1145/3539618.3591626. doi:10.1145/3539618.3591626.

[8] N. Ferro, Reproducibility challenges in information retrieval evaluation, ACM J. Data Inf. Qual. 8 (2017) 8:1–8:4. URL: https://doi.org/10.1145/3020206. doi:10.1145/3020206.

[9] N. Fuhr, Some common mistakes in IR evaluation, and how they can be avoided, SIGIR Forum 51 (2017) 32–41. URL: https://doi.org/10.1145/3190580.3190586. doi:10.1145/3190580.3190586.

[10] M. F. Dacrema, S. Boglio, P. Cremonesi, D. Jannach, A troubling analysis of reproducibility and progress in recommender systems research, ACM Trans. Inf. Syst. 39 (2021) 20:1–20:49. URL: https://doi.org/10.1145/3434185. doi:10.1145/3434185.

[11] V. Karpukhin, B. Oguz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, W. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 6769–6781. URL: https://doi.org/10.18653/v1/2020.emnlp-main.550. doi:10.18653/V1/2020.EMNLP-MAIN.550.

[12] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune BERT for text classification?, in: M. Sun, X. Huang, H. Ji, Z. Liu, Y. Liu (Eds.), Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings, volume 11856 of *Lecture Notes in*

Computer Science, Springer, 2019, pp. 194–206. URL: https://doi.org/10.1007/978-3-030-32381-3_16. doi:10.1007/978-3-030-32381-3_16.

[13] H. Choi, J. Kim, S. Joe, Y. Gwon, Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks, in: 25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021, IEEE, 2020, pp. 5482–5487. URL: https://doi.org/10.1109/ICPR48806.2021.9412102. doi:10.1109/ICPR48806.2021.9412102.

[14] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/V1/N19-1423.

[15] S. E. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Found. Trends Inf. Retr. 3 (2009) 333–389. URL: https://doi.org/10.1561/1500000019. doi:10.1561/1500000019.

[16] C. K. Reddy, L. Màrquez, F. Valero, N. Rao, H. Zaragoza, S. Bandyopadhyay, A. Biswas, A. Xing, K. Subbian, Shopping queries dataset: A large-scale ESCI benchmark for improving product search, CoRR abs/2206.06588 (2022). URL: https://doi.org/10.48550/arXiv.2206.06588. doi:10.48550/ARXIV.2206.06588. arXiv:2206.06588.

[17] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation, Mach. Learn. 94 (2014) 233–259. URL: https://doi.org/10.1007/s10994-013-5363-6. doi:10.1007/S10994-013-5363-6.

[18] A. Bordes, J. Weston, R. Collobert, Y. Bengio, Learning structured embeddings of knowledge bases, in: W. Burgard, D. Roth (Eds.), Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011, AAAI Press, 2011, pp. 301–306. URL: https://doi.org/10.1609/aaai.v25i1.7917. doi:10.1609/AAAI.V25I1.7917.

[19] D. Ayala, A. Borrego, I. Hernández, C. R. Rivero, D. Ruiz, AYNEC: all you need for evaluating completion techniques in knowledge graphs, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, K. Hammar (Eds.), The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings, volume 11503 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 397–411. URL: https://doi.org/10.1007/978-3-030-21348-0_26. doi:10.1007/978-3-030-21348-0_26.

[20] R. Socher, D. Chen, C. D. Manning, A. Y. Ng, Reasoning with neural tensor networks for knowledge base completion, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 926–934. URL: https://proceedings.neurips.cc/paper/2013/hash/b337e84de8752b27eda3a12363109e80-Abstract.html.

[21] S. Mazumder, B. Liu, Context-aware path ranking for knowledge base completion, in: C. Sierra (Ed.), Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ijcai.org, 2017, pp. 1195–1201. URL: https://doi.org/10.24963/ijcai.2017/166. doi:10.24963/IJCAI.2017/166.

[22] Z. Sun, S. Vashishth, S. Sanyal, P. P. Talukdar, Y. Yang, A re-evaluation of knowledge graph completion methods, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 5516–5522. URL: https://doi.org/10.18653/v1/2020.acl-main.489. doi:10.18653/V1/2020.ACL-MAIN.489.

[23] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, C. Li, Realistic re-evaluation of knowledge graph completion methods: An experimental study, in: D. Maier, R. Pottinger, A. Doan, W. Tan, A. Alawini, H. Q. Ngo (Eds.), Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020, ACM, 2020, pp. 1995–2010. URL: https://doi.org/10.1145/3318464.3380599. doi:10.1145/3318464.3380599.

[24] M. Gardner, T. M. Mitchell, Efficient and expressive knowledge base completion using subgraph feature extraction, in: L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, Y. Marton (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, The Association for Computational Linguistics, 2015, pp. 1488–1498. URL: https://doi.org/10.18653/v1/d15-1173. doi:10.18653/V1/D15-1173.

[25] H. Scells, S. Zhuang, G. Zuccon, Reduce, reuse, recycle: Green information retrieval research, in: E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, G. Kazai (Eds.), SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022, ACM, 2022, pp. 2825–2837. URL: https://doi.org/10.1145/3477495.3531766. doi:10.1145/3477495.3531766.

[26] G. Spillo, A. D. Filippo, C. Musto, M. Milano, G. Semeraro, Towards sustainability-aware recommender systems: Analyzing the trade-off between algorithms performance and carbon footprint, in: J. Zhang, L. Chen, S. Berkovsky, M. Zhang, T. D. Noia, J. Basilico, L. Pizzato, Y. Song (Eds.), Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023, ACM, 2023, pp. 856–862. URL: https://doi.org/10.1145/3604915.3608840. doi:10.1145/3604915.3608840.

[27] G. Chowdhury, An agenda for green information retrieval research, Inf. Process. Manag. 48 (2012) 1067–1077. URL: https://doi.org/10.1016/j.ipm.2012.02.003. doi:10.1016/J.IPM.2012.02.003.