A Reproducibility Study for Joint Information Retrieval and Recommendation in Product Search

Simone Merlo[1 [0009−0003−8003−4795]], Guglielmo Faggioli[1 [0000−0002−5070−2049]], and Nicola Ferro[1 [0000−0001−9219−6239]]

University of Padua, Padua, Italy

Abstract. Information Retrieval (IR) systems and Recommender Systems (RS) are ubiquitous commodities, essential to satisfy users' information needs in digital environments. These two classes of systems are traditionally treated as two isolated components with limited, if any, interaction. Recent studies showed that jointly operating retrieval and recommendation allows for improved performance on both tasks. In this regard, the state-of-the-art is represented by the Unified Information Access (UIA) framework. In this work, we analyse the UIA framework from the reproducibility, replicability and generalizability sides. To do this, we first reproduce the original results the UIA framework achieved – highlighting a good reproducibility degree. Then we examine the behavior of UIA when using a public dataset – discovering that UIA is not always replicable. Moreover, to further investigate the generalizability of the UIA framework, we introduce some changes in its data processing and training procedures. Our empirical assessment highlights that the robustness and effectiveness of the UIA framework depend on several factors. In particular, some tasks, such as the Keyword Search, appear to be more robust, while others, such as Complementary Item Retrieval, are more vulnerable to changes in the underlying training process.

Keywords: Information Retrieval · Recommender Systems · Large Language Models.

1 Introduction

From the user's perspective, it is natural to see Information Retrieval (IR) systems results and Recommender Systems (RS) results merged, such as the "suggested products" presented by most search engines when interrogated. Presenting together the results of these two types of systems allows for providing users with a comprehensive answer to their information needs, by extracting the most satisfactory piece of information from a corpus – or catalogue – of information – or items. Along this line, Belkin and Croft [3] consider these two tasks as "two sides of the same coin". Even though there are some differences between IR and RS concerning their input (textual queries and historical interactions with the system, respectively), both techniques aim at satisfying the user's information

need [11, 21], by ranking either documents or items. This pattern is particularly evident in scenarios such as e-commerce, where queries usually are short, keyword-based descriptions of a product the user is looking for, and the difference between IR and RS is blurred [3, 16, 23, 30]. At the same time, when searching for information across a search session, it is common for users to seek related information over multiple queries and the system could benefit from previous interactions with the user [6, 29].

Despite the relationships between the two tasks, historically, IR systems and RS have been developed independently, with only recent efforts devoted to investigating these tasks jointly. In particular, Si et al. [23] and Zamani and Croft [30, 31] showed that combining IR and RS models allows for improved performance by exploiting the knowledge held by a model to enhance the other. Such efforts focus on either refining a RS model by exploiting the search data [23], or on gaining additional knowledge to improve on one task using the data held by the other [31]. However, the joint modelling of both IR and RS tasks is still underdeveloped. In this regard, the two major efforts are SRJGraph, proposed by Zhao et al. [33], and the Unified Information Access (UIA) framework, developed by Zeng et al. [32]. The main advantage of developing joint IR and RS models is that it is possible to create a shared knowledge base between the two tasks [30, 33]. This allows for improving the performance of both tasks by enabling them to support each other [18, 31, 32].

The importance of reproducibility is well-recognized by both IR and RS research communities, but so are the challenges in achieving it [9, 12, 13]. Motivated by this, and given the recent interest in joint IR and RS, in this paper, we analyse the UIA framework [32] from the reproducibility point of view. UIA represents the current state-of-the-art in the joint IR and RS field and its architecture may become a base to develop new systems. UIA was originally trained and evaluated using both a private (Lowe's) and a publicly available (Amazon ESCI) datasets and the bulk of the experiments was conducted on the private one, which also enabled more functionalities. We consider the ACM "Artifact Review and Badging" guidelines[1] and evaluate the approach based on three axes: 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).

Our main goal is to reproduce the performance results obtained by Zeng et al. [32] for UIA (reproducibility), to understand if the observations made in the ablation study by Zeng et al. [32], using the private dataset hold when using the publicly available dataset (replicability) and to analyse the behaviour of UIA when the training procedure/data is modified (generalizability). In this perspective, we articulate our work on three research questions:

- **RQ1 - Reproducibility**: is the performance achieved by UIA, and reported in [32], reproducible?
- **RQ2 - Replicability**: is the performance of UIA replicable on a publicly available dataset?

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

– **RQ3 - Generalizability**: how does the performance UIA change when using alternative approaches which are less computationally demanding and/or involve different data processing methods?

Our empirical evaluation shows that UIA can be reproduced and that the robustness and effectiveness of UIA depends on different factors. Specifically, we discovered that the Amazon ESCI dataset may not be the best to be used in conjunction with UIA and that due to the nature of this dataset and the way in which it is processed, the Keyword Search (KS) task appears to be more robust to changes in the training process while the Query By Example (QBE) and Complementary Item Recommendation (CIR) tasks are more vulnerable.

The code we used to answer our research questions is publicly available[2].

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

2 Highlights of the Reproduced Approach

In this section, we introduce the UIA framework and the Amazon ESCI dataset that we used to reproduce, replicate and generalize it.

2.1 The UIA Framework

According to Zeng et al. [32], an interaction between the user and the UIA framework (Figure 1), is defined by three elements: an information access request $\mathcal{R}$, a task label (access functionality in [32]) $\mathcal{F}$, and a candidate information item $\mathcal{I}$. UIA supports three main hybrid RS-IR tasks (functionalities in [32]) $\mathcal{F}$: i) Keyword Search (KS) where a short textual query is used to retrieve the most relevant items; ii) Query By Example (QBE) where an item is used as input to retrieve other similar items; and iii) Complementary Item Recommendation (CIR) that consists of retrieving items that "can be used together" (i.e., complementary) with the item given as input. Depending on the scenario, the information access request $\mathcal{R}$ is a keyword query (in case of KS) or an item (in case of QBE or CIR). Finally, the candidate information item $\mathcal{I}$, is a textual representation of the candidate item (e.g., its title, in the Amazon ESCI dataset) for which the system must estimate its relevance to $\mathcal{R}$. Thus, given a task $\mathcal{F}$ and a request input $\mathcal{R}$, the objective of the UIA model, parametrized by θ, is to sort all the items $\mathcal{I}$ in the catalogue based on a relevance score s, computed as $s = f(\mathcal{R}, \mathcal{F}, \mathcal{I}; \theta)$. To do so, the UIA framework relies on a bi-encoder architecture. In particular, it employs a request encoder $\mathbf{E}_{\mathcal{R}}$ and an item encoder $\mathbf{E}_{\mathcal{I}}$. These two components embed a request $\mathcal{R}$ (jointly with the task label $\mathcal{F}$) and an item $\mathcal{I}$ within a latent space, respectively. More in detail, $\mathcal{R}$ is encoded as $\mathbf{R} = \mathbf{E}_{\mathcal{R}}(\texttt{[CLS]}\ \mathcal{R}\ \texttt{[SEP]}\ \mathcal{F}\ \texttt{[SEP]})$, where $\mathcal{F}$ is the label corresponding to the

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

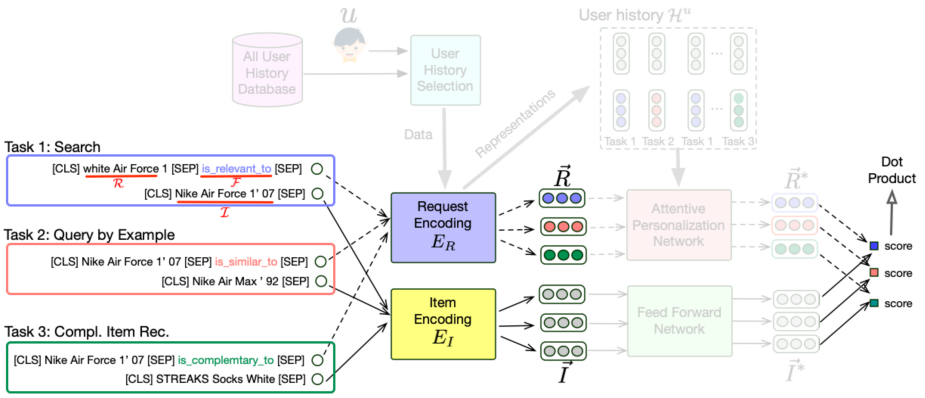


Fig. 1: UIA framework architecture. Grayed-out areas are those concerning personalization, that we did not experiment with. Figure taken from [32].

task associated to the request, while `[CLS]` and `[SEP]` are the "class" and "separator" tokens respectively. Similarly $\mathcal{I}$ is encoded as $\mathbf{I} = \mathbf{E}_{\mathcal{I}}(\texttt{[CLS]}\ \mathcal{I}\ \texttt{[SEP]})$. As commonly done in this setting [7, 15, 26], the final representation is the embedding of the `[CLS]` token. Both $\mathbf{E}_{\mathcal{R}}$ and $\mathbf{E}_{\mathcal{I}}$ employ the BERT [10] model to encode their input. Finally, the score of the item $\mathcal{I}$ in response to the request $\mathcal{R}$ is computed as $s = \mathbf{R} \cdot \mathbf{I}$.

To determine the (optimal) parameters θ, the UIA framework minimizes a cross-entropy loss function. Thus, 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. To obtain the negative examples, not available in the original dataset, Zeng et al. [32] define a two-phase negative sampling. The first phase (*Phase 1*) samples a set of negatives among the items retrieved by BM25 [20] 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. Importantly, the model trained during Phase 2, is initialized with the weights learned during Phase 1. The training procedure involves also the usage of in-batch negatives and mini-batches.

Notice that, Zeng et al. [32] propose a second training pipeline to handle users' data and personalize the output. Notice that, such a pipeline requires accessing user's personal data (i.e., previous interactions with the system and preferences). In this paper, we focus exclusively on non-personalized data (i.e., the Amazon ESCI Dataset), thus we describe only the non-personalized part of the pipeline.

2.2 The Amazon ESCI Dataset

In [32], UIA was trained and evaluated on two datasets: the Lowe's dataset and the Amazon ESCI dataset [19]. The former is private and contains user data

to enable personalization, the latter is public but does not contain users' 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 [19] 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 [32], we consider the product catalogue and the training data used for Task 2 [19] of the Amazon ESCI challenge. In detail, 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". Notice that, the ESCI dataset contains only textual queries, thus is unsuitable for QBE and CIR tasks. To address this, in line with [32], we split the full dataset into three separate datasets, one for each task (KS, QBE and CIR). More in detail, 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 [32], 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 used the code available at: https://github.com/Hansi Zeng/UIA. Importantly, our experiments are based only on publicly available datasets and code. We operated independently on whether the original developers were available to share with us their knowledge, to put ourselves in the most challenging reproducibility conditions and work in the most aseptic way.

We report here the challenges we identified in reproducing the approach and the solution we employed to address them.

Second sample of the relevant items. While inspecting the available code base, we observed that a second sampling is executed after the dataset splitting described in Section 2.2. In particular, for QBE and CIR, for every unique query item

[3] KDD Cup 2022: https://amazonkddcup.github.io/

(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, in response to each query, only 10 relevant items are used to construct instances in the dataset. We ascribe this difference between the original paper and the code to efficiency reasons and avoid excessively large datasets. At the same time, this second sampling prevents weighting too much queries which are too popular or generic items; thus, it limits the contribution of each entity in the original dataset.

To ensure reproducibility, we maintain this implementation choice, reducing the size of the constructed datasets.

Negative sampling procedure. In the available code base to construct the QBE dataset, differently from KS and CIR, the *Phase 1* negative sampling only partially follows the procedure described in Section 2.1. More in detail, the negative samples used during the first training phase are randomly sampled among all the items. In line with the paper [32], we modified the provided code and sampled the negatives from the items retrieved by BM25, also for QBE.

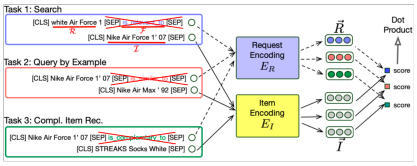
Another difference we observed concerns the KS dataset. In detail, when sampling the negative examples for the KS task during *Phase 1*, for each pair request-item in the dataset the negative is randomly sampled from the items similar (i.e., labelled "Substitute") to the one considered as positive, if present, else the negative is randomly sampled 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 provided by Zeng et al. [32].

Computational Resources. As an additional caveat, we point out that, due to limited computational resources — especially concerning GPU memory — we reduced the batch size from 384 (used in [32]) to 48 (-86%). For the same reason, we set the number of epochs to 24 while, according to [32], the optimal epochs are 48. Other hyperparameters, such as learning rate and the number of warmup iterations, were left unchanged compared to the original paper, e.g., the learning rate used is set to $7e^{-6}$ and the number of warmup iterations is 4,000.

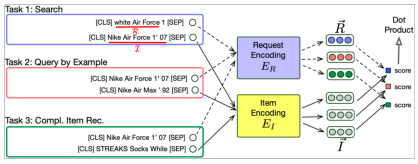
Phase 1 Only The double-phase training is computationally expensive, doubling the training time and cost (including the carbon footprint and environmental impact). Therefore, we assess the performance of the UIA framework after a single training phase. While we reasonably expect a decrease in terms of performance, we are interested in assessing whether this represents an acceptable trade-off between effectiveness and efficiency. If this is the case, the UIA framework could also be used in a resource-constrained environment, e.g., by small companies, with a reduced cost and environmental impact.

3.2 RQ2: Replicability

The experiments in [32] focus mostly on the Lowe's dataset, which is private, and only some of the analyses are carried out on the Amazon ESCI dataset.



(a) Removing task label from UIA architecture.



(b) UIA architecture without task label.

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

Therefore, concerning replicability, we are interested in extending the analysis of the UIA for the Amazon ESCI dataset. More in detail, we replicate on the Amazon ESCI dataset the experiments done by Zeng et al. [32] only on the Lowe's dataset.

No Task Label (w/o $\mathcal{F}$) An aspect of the UIA framework we are interested in investigating is the role that the task label $\mathcal{F}$ plays in it. Specifically we want 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 the task label $\mathcal{F}$ (we will refer to the version of the framework without the functionality as "w/o $\mathcal{F}$"). This allows to consider the training data related to the different tasks as data belonging to a unique training set.

More in detail, given its relatively large training data, the interchangeable nature 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 the task labels are 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. $\mathbf{R}' = \mathbf{E}'_{\mathcal{R}}(\texttt{[CLS]}\ \mathcal{R}\ \texttt{[SEP]})$. The new score s'_i for the candidate item $\mathcal{I}_i$ is computed as $s'_i = \mathbf{R}' \cdot \mathbf{I}_i$. In Figure 2 we show the original structure of the framework highlighting the portions that are removed (Figure 2a) and the new architecture without the task label (Figure 2b).

Isolated Tasks Training and evaluating the UIA framework on the tasks in isolation corresponds to optimizing and evaluating three separate instances of UIA, each one for each task. The authors of the paper [32] 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). For this reason, 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 (described in Section 3.1), for this experiment, we consider a single-phase training instead of the original two phases. 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 the UIA framework, i.e., its resilience to major changes to its training procedure and, especially, to the training data.

Half QBE An aspect that can be observed by inspecting the generated datasets is that (after the sampling, Section 3.1) the training set for QBE (composed of 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 [32], this characteristic was also observed by Zeng et al.. In fact, in the provided repository, some portions of code use only half of the QBE dataset. These results were not reported or explicitly mentioned in [32]. 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.

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 the UIA framework, 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 (i.e., the retrieved or recommended item). The procedure to spit the collection into training, validation, and test set adopted by Zeng et al. [32], consists of considering all the possible triplets $(\mathcal{R}, \mathcal{F}, \mathcal{I})$ and randomly partitioning them into the three sets. While this procedure is commonly adopted in the "Knowledge Graph Completion" domain [2, 4, 5, 17, 24, 27], it also is criticized by other authors [1, 14]. In particular, Akrami et al. [1], 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.

By construction, this occurs in the dataset used to train UIA. In fact, 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. Once these sets have been defined, we use the procedure proposed by Zeng et al. [32], and described in Section 2.2, to generate the corresponding triplets. This ensures that all the information regarding a certain query is contained in the same partition. This also appears natural from a "temporal"

standpoint: the user will issue a query at a certain moment and it will be possible to collect the training data, up to that point. The system does not have any knowledge of the next user's query (i.e., the test). Using part of the information derived from such a query 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 (described in Section 3.1), for this experiment, we consider a single-phase training instead of the original two phases. Therefore, the results should be compared with those obtained for the experiment *Phase 1 Only*.

4 Experimental Results

In this section we discuss the results that we obtained while reproducing the work of the paper and performing the experiments described in Section 3. In table 1 we report the original performance of UIA (first row) and the performance of its reproduced version and its variations. Following [32], we evaluate our results according to MRR@10, nDCG@10 and Recall@50.

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 second row of Table 1 contains the results we achieved when reproducing UIA. Concerning KS and CIR, we observe relatively close performance. More in detail, for KS, we achieve -0.041 (-7.7%) MRR points and similar results also for nDCG (-0.033) and Recall (-0.049). Similarly, for CIR, we obtain -0.027 (-5.5%) in terms of MRR, -0.034 nDCG, and -0.035 for Recall. These results appear satisfactory, considering that, as previously mentioned, due to limited computing capabilities, we were forced to reduce the batch size and epochs. In this regard, UIA achieves satisfactory performance even under stronger resource constraints. Interestingly, our results for the QBE task are by far larger than those reported in the original paper. Indeed, we obtain +0.191 (+76%) in terms of MRR, with comparable improvements also for nDCG (+0.175) and Recall

(+0.130). We explain this phenomenon considering that we changed the Phase 1 negative sampling (from random to BM25, as explained in Section 3.1), aligning it with the one used for the other tasks. We hypothesise that the results reported in [32] represent a lower bound of the actual performance UIA can achieve on the QBE task.

Phase 1 Only In this case, we consider the model obtained after a single training phase (third line of Table 1). As reasonably expected, the performance drops but the drop magnitude depends on the task. For KS we notice a minor drop in performance (-0.014 MRR, -0.014 nDCG, 0.023 Recall). This suggests that the second training phase has a limited impact on this specific task. CIR is the task with the second-biggest drop (-0.102 MRR, -0.099 nDCG, -0.073 Recall). Finally, QBE is the task where removing the second phase has the direst consequences (-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 and so are the computational resources saved: from approximately 240 hours of computation to 120 with a reduction of 50%.

4.2 RQ2: Replicability Results

No Task Label (w/o $\mathcal{F}$) The " w/o $\mathcal{F}$ " row of Table 1 reports the result we achieve when removing the task information. In this case, the behaviour of UIA on the Amazon ESCI dataset is consistent with the ablation study on the Lowe's daset reported in [32]. Interestingly, the KS task is the least vulnerable (-0.011 MRR, -0.016 nDCG, and +0.006 Recall compared to Phase1Only results). On the contrary, CIR is the most affected task with approximately 37% drop in performance for both MRR and nDCG (-0.180 MRR, -0.167 nDCG, and -0.112 Recall). This suggests that, if the task label is not expressed, the model is still able to operate on KS, while being less performing for QBE and CIR – this might be due to a different term distribution between queries and items, used as input for KS and QBE and CIR tasks. Furthermore, the difference in performance loss between QBE and CIR might be explained by the different sizes in training sets. The QBE dataset is much larger than the CIR (580%). In this sense, 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 row "w/o $\mathcal{F}$ (Phase1Only)" of Table 1 reports the result we achieve when removing the task information and training the framework only according to *Phase 1*. This performance 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.

Isolated Tasks The "IsolatedTasks" row of Table 1 reports the results achieved for the three versions of UIA which are optimized on a single task, grouped together. The obtained performance highlights that, when the Amazon ESCI

dataset is exploited, UIA has lower performance when is jointly optimized than when is trained on a single task. Again, the KS task appears to be the most stable (+0.029 MRR, +0.027 nDCG, and +0.032 Recall compared to Phase1Only results). 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 when considered in isolation (+0.053 MRR, +0.059 nDCG, and +0.019 Recall). This behaviour is not consistent with the previous studies about joint IR and RS [30, 31, 33] and with the UIA results reported in [32] when the Lowe's dataset is used. With respect to using the Lowe's dataset, exploiting the Amazon ESCI dataset implies to both change the data and the structure of UIA. In particular the Amazon ESCI dataset does not contain any user data and, thus, the personalization components are removed from the architecture of the framework. For this reason, the results highlight that the advantages gained from the joint training may depend on the nature of the datasets used, on the way in which they are processed and on the architecture of the framework chosen.

4.3 RQ3: Generalizabilty Results

As mentioned before we employ the model trained with a single phase when experimenting with some generalizability aspects, to adopt a more ethical approach towards IR reseach [8, 22, 25].

Half QBE This experiment ("HalfQBE" row of Table 1) aims to evaluate UIA behaviour when halving the QBE training data. In this case, we notice three interesting patterns: i) Compared to our implementation of the UIA framework the KS performance increases (+0.019 MRR, +0.014 nDCG, and +0.020 Recall). This indicates that by reducing the imbalance between the different datasets, the UIA model was more effective in learning how to deal with KS instances. ii) The performance on QBE decreases (-0.126 MRR, -0.124 nDCG, -0.112 Recall): this can be naturally explained considering that the training data on this task was reduced. iii) On the CIR task, we observe a slight decrease in performance for MRR (-0.008), and nDCG(-0.007) and an increase in Recall (+0.005). The negligible change for this task suggests that its training phase is not influenced by the training data used for the QBE task. This can be explained by the different semantics of the three tasks. Both KS and QBE require to retrieve an item "similar" to the input (either a query or an item) and thus their learning is tightly coupled, with the excessive amount of QBE data overshadowing KS. On the contrary, when it comes to CIR, the expected output is a related item that is explicitly not similar, thus its training is likely disentangled from the two other tasks.

The row "HalfQBE (Phase1Only)" of Table 1 reports the result we achieve when usign half of the data for QBE and training the framework only according to *Phase 1*. This performance must be compared with the one of the "Phase1Only" experiment. By looking at the results we can notice that, when avoiding *Phase 2*, for KS nothing changes, for QBE the gap in performance is

bigger, while for CIR instead of having a slight decrease we have a small increase in performance. Nonetheless, the considerations made for this experiment when performing both training phases still hold.

Early Split The "Early Split" row of Table 1 reports the results we achieve when we first split the Amazon ESCI queries into training and test set and then we construct the datasets used to train and test the framework. Compared to Phase1Only, we do not notice major differences for KS. This happens because the Amazon ESCI dataset is an IR dataset based on real human behaviour and the dataset for KS is obtained from it by selecting the appropriate entries, without requiring special assumptions/processing (differently to QBE and CIR). Therefore, regardless of the preprocessing pipeline, the KS dataset results to be more realistic, leading to a more stable performance. Vice-versa, if the UIA framework is trained by first splitting the queries into training and test sets, the performance on QBE and CIR tasks is extremely low. This result suggests that there are some scenarios in which the model will achieve unsatisfactory performance. For example, if a new item is used to query the system and the model does not have prior knowledge of such an item, it will be doomed to fail. Furthermore, this gives us important information on handling the testing/training of this class of models. In general, it would be more informative to report results when the split occurs at a tuple level (as done in [32]) but also split into training and test sets 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. This should also motivate the joint IR and RS research community, inspired by previous work on "knowledge graph completion" evaluation, to investigate, develop, codify, and adopt proper evaluation protocols that can address and correctly represent corner cases and the various sides of the task.

5 Conclusions and Future Works

In this paper we presented the architecture of the UIA framework and how the publicly available Amazon ESCI dataset has been processed and used to optimize it. Furthermore, with our studies and experiments, we discovered that is possible to reproduce the results obtained by UIA on the Amazon ESCI dataset. We highlighted the fact that the behavior of the framework is not completely replicable and, in particular, that when the Amazon ESCI dataset is exploited the framework does not benefit from the joint training. We generalized the framework discovering that the dataset used and the way in which it is manipulated has an impact on the performance of UIA. Eventually, all the experiments carried out showed that, when the Amazon ESCI dataset is used, the KS task appears to be more robust while the recommendation tasks are more vulnerable.

We will continue our work towards the analysis and generalization of this framework. In particular, we will try to find and exploit publicly available datasets which contain user data and can be processed and adapted to fit in the field of

"joint retrieval and recommendation". The Amazon Reviews dataset, which has been frequently used in the field of recommendation [28], seems to be a good candidate for this role. Our future studies will focus on enhancing the benefits deriving from the joint training and on understanding how the concepts and novelties introduced with UIA can be reused to develop innovative joint retrieval and recommendation systems.

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

References

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

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

[3] Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: Two sides of the same coin? Commun. ACM **35**(12), 29–38 (1992), https://doi.org/10.1145/138859.138861, URL https://doi.org/10.1145/138859.138861

[4] Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Mach. Learn. **94**(2), 233–259 (2014), https://doi.org/10.1007/S10994-013-5363-6, URL https://doi.org/10.1007/s10994-013-5363-6

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

[6] Carterette, B., Clough, P.D., Hall, M.M., Kanoulas, E., Sanderson, M.: Evaluating retrieval over sessions: The TREC session track 2011-2014. In: Perego, R., Sebastiani, F., Aslam, J.A., Ruthven, I., Zobel, J. (eds.) Proceedings of the 39th International ACM SIGIR conference on Research and

Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016, pp. 685–688, ACM (2016), https://doi.org/10.1145/2911451.2914675, URL https://doi.org/10.1145/2911451.2914675

[7] Choi, H., Kim, J., Joe, S., Gwon, Y.: 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, pp. 5482–5487, IEEE (2020), https://doi.org/10.1109/ICPR48806.2021.9412102, URL https://doi.org/10.1109/ICPR48806.2021.9412102

[8] Chowdhury, G.: An agenda for green information retrieval research. Inf. Process. Manag. **48**(6), 1067–1077 (2012), https://doi.org/10.1016/J.IPM.2012.02.003, URL https://doi.org/10.1016/j.ipm.2012.02.003

[9] Dacrema, M.F., Boglio, S., Cremonesi, P., Jannach, D.: A troubling analysis of reproducibility and progress in recommender systems research. ACM Trans. Inf. Syst. **39**(2), 20:1–20:49 (2021), https://doi.org/10.1145/3434185, URL https://doi.org/10.1145/3434185

[10] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (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), pp. 4171–4186, Association for Computational Linguistics (2019), https://doi.org/10.18653/V1/N19-1423, URL https://doi.org/10.18653/v1/n19-1423

[11] Dong, Z., Wang, Z., Xu, J., Tang, R., Wen, J.: A brief history of recommender systems. CoRR **abs/2209.01860** (2022), https://doi.org/10.48550/ARXIV.2209.01860, URL https://doi.org/10.48550/arXiv.2209.01860

[12] Ferro, N.: Reproducibility challenges in information retrieval evaluation. ACM J. Data Inf. Qual. **8**(2), 8:1–8:4 (2017), https://doi.org/10.1145/3020206, URL https://doi.org/10.1145/3020206

[13] Fuhr, N.: Some common mistakes in IR evaluation, and how they can be avoided. SIGIR Forum **51**(3), 32–41 (2017), https://doi.org/10.1145/3190580.3190586, URL https://doi.org/10.1145/3190580.3190586

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

[15] Karpukhin, V., Oguz, B., Min, S., Lewis, P.S.H., Wu, L., Edunov, S., Chen, D., Yih, W.: Dense passage retrieval for open-domain question answering. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pp. 6769–6781, Association for Com-

putational Linguistics (2020), https://doi.org/10.18653/V1/2020.EMNLP-MAIN.550, URL https://doi.org/10.18653/v1/2020.emnlp-main.550

[16] Luo, C., Goutam, R., Zhang, H., Zhang, C., Song, Y., Yin, B.: Implicit query parsing at amazon product search. In: Chen, H., Duh, W.E., Huang, H., Kato, M.P., Mothe, J., Poblete, B. (eds.) Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, pp. 3380–3384, ACM (2023), https://doi.org/10.1145/3539618.3591858, URL https://doi.org/10.1145/3539618.3591858

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

[18] Penha, G., Vardasbi, A., Palumbo, E., Nadai, M.D., Bouchard, H.: Bridging search and recommendation in generative retrieval: Does one task help the other? In: Noia, T.D., Lops, P., Joachims, T., Verbert, K., Castells, P., Dong, Z., London, B. (eds.) Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, October 14-18, 2024, pp. 340–349, ACM (2024), https://doi.org/10.1145/3640457.3688123, URL https://doi.org/10.1145/3640457.3688123

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

[20] Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009), https://doi.org/10.1561/1500000019, URL https://doi.org/10.1561/1500000019

[21] Sanderson, M., Croft, W.B.: The history of information retrieval research. Proc. IEEE **100**(Centennial-Issue), 1444–1451 (2012), https://doi.org/10.1109/JPROC.2012.2189916, URL https://doi.org/10.1109/JPROC.2012.2189916

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

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

https://doi.org/10.1145/3539618.3591786, URL https://doi.org/10.1145/3539618.3591786

[24] Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (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, pp. 926–934 (2013), URL https://proceedings.neurips.cc/paper/2013/hash/b337e84de8752b27eda3a12363109e80-Abstract.html

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

[26] Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings, Lecture Notes in Computer Science, vol. 11856, pp. 194–206, Springer (2019), https://doi.org/10.1007/978-3-030-32381-3_16, URL https://doi.org/10.1007/978-3-030-32381-3_16

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

[28] Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., Xiong, H., Chen, E.: A survey on large language models for recommendation. World Wide Web (WWW) **27**(5), 60 (2024), https://doi.org/10.1007/S11280-024-01291-2, URL https://doi.org/10.1007/s11280-024-01291-2

[29] Xue, Y., Cui, G., Yu, X., Liu, Y., Cheng, X.: ICTNET at session track TREC2014. In: Voorhees, E.M., Ellis, A. (eds.) Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014, Gaithersburg, Maryland, USA, November 19-21, 2014, NIST Special Publication, vol. 500-308, National Institute of Standards and Technology (NIST) (2014), URL http://trec.nist.gov/pubs/trec23/papers/pro-ICTNET_session.pdf

[30] Zamani, H., Croft, W.B.: Joint modeling and optimization of search and recommendation. In: Alonso, O., Silvello, G. (eds.) Proceedings of the First Biennial Conference on Design of Experimental Search & Information Re-

trieval Systems, Bertinoro, Italy, August 28-31, 2018, CEUR Workshop Proceedings, vol. 2167, pp. 36–41, CEUR-WS.org (2018), URL https://ceur-ws.org/Vol-2167/paper2.pdf

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

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

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