



HAL
open science

RULKKG: Estimating User’s Knowledge Gain in Search-as-Learning Using Knowledge Graphs

Hadi Nasser, Dima El Zein, Célia da Costa Pereira, Cathy Esczut, Andrea G. B. Tettamanzi

► **To cite this version:**

Hadi Nasser, Dima El Zein, Célia da Costa Pereira, Cathy Esczut, Andrea G. B. Tettamanzi. RULKKG: Estimating User’s Knowledge Gain in Search-as-Learning Using Knowledge Graphs. CHIIR 2024: 2024 ACM SIGIR Conference on Human Information Interaction and Retrieval, Mar 2024, Sheffield - UK, United Kingdom. pp.364-369, 10.1145/3627508.3638331 . hal-04567545

HAL Id: hal-04567545

<https://inria.hal.science/hal-04567545>

Submitted on 3 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

RULK_{KG}: Estimating User’s Knowledge Gain in Search-as-Learning Using Knowledge Graphs

HADI NASSER, Université Côte d’Azur, I3S Lab, CNRS, UMR 7271, France

DIMA EL ZEIN, Université Côte d’Azur, I3S Lab, CNRS, UMR 7271, France

CÉLIA DA COSTA PEREIRA, Université Côte d’Azur, I3S Lab, CNRS, UMR 7271, France

CATHY ESCAZUT, Université Côte d’Azur, I3S Lab, CNRS, UMR 7271, France

ANDREA TETTAMANZI, Université Côte d’Azur, I3S Lab, CNRS, UMR 7271, Inria, France

In the context of search as learning, users engage in search sessions to fill their information gaps and achieve their learning goals. Tracking the user’s state of knowledge is therefore essential for estimating how close they are to achieve these learning goals. In this respect, we extend a recently proposed approach that uses the recognition of entities present in the text to track the user’s knowledge. Our approach introduces a more complete representation by considering both the entities and their relations. More precisely, we represent both the user’s knowledge and the user’s learning goals (or target knowledge) as knowledge graphs.

We show that the proposed representation captures a complementary aspect of knowledge, thus helping to improve the user knowledge gain estimation when used in combination with other representations.

CCS Concepts: • **Information systems** → **Information retrieval**; **Information retrieval**; • **Human-centered computing** → **Human computer interaction (HCI)**.

Additional Key Words and Phrases: Search-As-Learning, User Knowledge, Knowledge Graphs, Interactive IR, Retrieval system

ACM Reference Format:

Hadi Nasser, Dima El Zein, Célia da Costa Pereira, Cathy Escazut, and Andrea Tettamanzi. 2024. RULK_{KG}: Estimating User’s Knowledge Gain in Search-as-Learning Using Knowledge Graphs. In *Proceedings of the 2024 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR ’24)*, March 10–14, 2024, Sheffield, United Kingdom. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3627508.3638331>

1 INTRODUCTION AND RELATED WORK

Understanding the reason(s) why a user is looking for information can help an Information Retrieval (IR) system to make the most appropriate answers to meet what is supposed to be the user’s real need, especially in the context of Search-as-Learning (SAL). However, understanding the real needs of the user in search of information is not an easy task. Assuming that the user’s query perfectly matches his or her needs is utopian. Belkin *et al.* in their seminal work [3] defined the concept of “Anomalous State of Knowledge (ASK)” as the state representing what is missing in the user’s knowledge state that prevents them to be able to precisely formulate through a query, what they need.

Rose *et al.*, who highlight the fact that search engines have focused much more on how the user looks for information (e.g. queries), than on the reasons why they look for it [29], proposed a framework to be used to understand the reasons for searching. The framework helps identifying and organizing the goals that users have in mind when they carry out a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

search. More recently, Shah *et al.* underlined the importance of the role a user’s task plays in how and why they engage in a search, and what a search system should do to help them.

In particular, they stressed the fact that

“the IR community has been too focused on query processing and assuming a search task to be a collection of user queries, often ignoring if or how such an assumption addresses the users accomplishing their tasks.”

They propose insights on understanding, extracting and addressing task-based search [30]. Syed *et al.* who pointed out that Web search engines are best suited to provide results based on a generic concept of relevance, which is not suited to provide results for user’s learning purposes. The optimization model they proposed, incorporates a cognitive learning model and the algorithm used proposes an approximate solution with results that represent the best “learning set” of pieces of information for an information-seeking user with the aim of learning [32].

We are convinced that it is possible to further improve the quality of the answers provided to the *learning user*. Indeed, in addition to knowing the reasons or goals of the user’s search, it is also important to have an idea of the user’s state of knowledge as well as the evolution of both the user’s goals and knowledge state during the search process. Pirolli *et al.* introduced a personalized learning path model, which tracks the user’s knowledge development through online interactions, using a social tagging strategy to predict relevant documents based on sets of recognized documents related to specific areas of interest [26].

The recent theoretical framework proposed by El Zein *et al.* in [18] uses a cognitive rule-based agent that adapts the document ranking based on both the user’s current knowledge and search goal. Finally, the work on SAL presented by Gadiraju in [19], aims to improve the understanding of the knowledge acquired by users during the information search sessions. We share his point of view that the ability to estimate the state of the users’ knowledge during their interactions with search systems can improve the users’ learning experiences. However, tracking the learning process is not simple, as we need a representation of both the user’s learning goals (or target knowledge) and state of knowledge, that captures the results of the user’s interactions with the system. The recent work by El Zein *et al.* [6, 17], represents a step forward in that direction. They consider a keyword-based representation of both the user’s current state of knowledge as well as the user’s learning goals, the aim of the user being to fill the gap between both through the interactions with the system. The framework then updates the user’s knowledge state after the reading of each document, and estimates the similarity between the current state and the target knowledge [6]. This work has been extended in [17] with the aim of capturing the information content from the named entities [5, 25]. However, the relations between those entities have not been considered, and as Shen *et al.* rightly pointed out in [31], considering the links between entities enables the entity mentioned in the text to be linked to the corresponding real-world entity in the existing knowledge base.

We fill this gap by proposing a knowledge graph representation of knowledge states. More precisely, we extend the named-entity based representation proposed in [17] and consider, in addition to the named entities, the relations between them. To validate our model, a dataset containing the users’ actual knowledge gains and their interaction logs from an organized search session was used. The effectiveness of the model is demonstrated by the correlation measure between the model’s estimated knowledge gain and the users’ actual knowledge.

The paper is organized as follows. Section 2 presents the existing RULK version that we will extend in this paper. Section 3 describes the novel graph-based implementation of RULK, while Section 4 presents and discusses the experiments and results. Finally, Section 5 concludes the paper.

2 BACKGROUND

In this section, we introduce the RULK framework, explaining the primary functions of its three components in tracking the user’s knowledge state throughout a search session with the aim of estimating their knowledge gain. Additionally, we provide a brief overview of the operation of its existing implementations.

2.1 Representing User Knowledge in Search-As-Learning (RULK)

The estimation of the user’s knowledge gain has a significant importance in the Search as Learning domain. We share the point of view according to which such an estimation can help increasing the effectiveness of information retrieval systems by illustrating the value and impact of the retrieved information on users’ understanding and learning. Indeed, by quantifying the knowledge gain, we can evaluate how well the system supports users in acquiring new knowledge and filling information gaps. This valuable information opens the opportunity to design personalized information retrieval systems that employ knowledge-adaptive approaches, thereby enhancing learning outcomes.

RULK [6] is one of the most recently proposed frameworks that estimates the user’s knowledge gain during a learning process. Its main goal is to estimate the user’s knowledge by comparing two representations of knowledge: the user’s target knowledge and the user’s current knowledge. On the one hand, the target knowledge refers to the specific knowledge that a user aims to acquire or obtain through an information retrieval process. As in [6], we suppose that the user’s target knowledge, i.e., the user’s actual need for each search topic, is the referenced Wikipedia document. On the other hand, it is assumed that the user’s current knowledge is updated after reading documents, except the Wikipedia page, on a specific topic during the search session. RULK [6] consists of three main components:

- The *Feature Extractor* (γ), which encodes any given document d into another representation, mainly a fixed-length vector.

$$\vec{v}_d = \gamma(d). \quad (1)$$

Both the Wikipedia document and the clicked document are encoded by γ in the same way.

- The *Updater* (σ): the state of the user’s knowledge is represented by a vector \vec{c}_{ks} . When a user clicks on a document, the σ component combines \vec{c}_{ks} with \vec{v}_d to obtain the updated knowledge state \vec{c}'_{ks} :

$$\vec{c}'_{ks} = \sigma(\vec{c}_{ks}, \vec{v}_d). \quad (2)$$

- The *Estimator* (θ), estimates the degree of achievement of the user’s learning goal as the similarity between the user’s current knowledge \vec{c}_{ks} and the target knowledge \vec{t}_{ks}

$$\tilde{G} = \theta(\vec{c}_{ks}, \vec{t}_{ks}), \quad (3)$$

from which the estimate of the user’s knowledge gain can be computed.

2.2 RULK Versions

There are 3 versions of the RULK framework, each one using a different technique to encode the document into an embedding vector.

The first one, $RULK_{KW}$, is based on a keyword and vocabulary learning model, the second one, $RULK_{LM}$, is a transformer-based model, and finally, the last model, $RULK_{NE}$, uses the Named Entity Recognition method.

$RULK_{KW}$ adopts the vocabulary learning model [1]. This model considers that a user achieves their need to learn a topic T when they learn a set of related vocabulary keywords. The keywords to be learned are defined using the Yet

Another Keyword Extraction (YAKE) method [8]. This method selects the most important keywords of a text relying on statistical feature extraction.

$RULK_{LM}$ uses BERT [11] as *Feature Extractor*.

$RULK_{NE}$ executes two natural language processing tasks, namely Named Entity Recognition and Entity Linking, to extract a set of entities, denoted as K_{tks} , from the target knowledge document and another set of entities, denoted as K_d , from the clicked document. The γ then transforms the target knowledge into a vector, denoted as \vec{V}_{tks} , by counting the occurrences of the 10 most common entities in K_{tks} [17]. Similarly, the clicked document d is encoded as a vector by counting the occurrences of the top-10 entities in K_{tks} within d .

In all versions, the σ updates the current knowledge state with a simple addition of two count vectors. Finally, the θ uses the cosine similarity measure to compute the similarity between two vectors representing the current and target knowledge states.

3 A KNOWLEDGE-GRAPH-BASED IMPLEMENTATION OF RULK

We propose a novel instantiation of RULK, named $RULK_{KG}$, using knowledge graphs (KG) to represent both the user and the target knowledge states.

Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities [21]. The entities (concepts and instances) are the vertices of the graph; the relations between entities, which are the arcs of the graph, can be represented as triples (s, p, o) , where s and o are entities and p is the name of a relation or property.

$RULK_{KG}$ is a knowledge-graph-based extension of the $RULK_{NE}$ [17] in the sense that, besides considering named entities, it also considers the relations between those entities. The purpose of *Named Entity Recognition* (NER) is to automatically extract and label specific named entities mentioned in a document. On the other hand, *Relation Extraction* (RE), focuses on identifying and extracting relationships or associations between these named entities.

3.1 Entity Recognition and Relation Extraction

Named Entity Recognition and Relation Extraction are crucial challenges for natural language processing research in general, and information extraction in particular. A *named entity* can be defined as a linguistic expression often associated, for example, with people (personalities), places and organizations. Named entity recognition plays a very important role in many information retrieval tasks [5, 14, 15, 25, 34]. In particular, the pre-trained REBEL [5] model has demonstrated its prowess, by outperforming several times the state-of-the-art in the field.

REBEL is an end-to-end relation extraction model based on BART [22] and uses a sequence-to-sequence (Seq2Seq) architecture to extract over 200 distinct relation types. Its primary objective is identifying and extracting triplets (subject, predicate, object) from raw text. We use REBEL to implement the NER and RE tasks. Thus, our *Feature Extractor* component represents a document as a *KG*. The *KG* that represents the user’s current knowledge is updated by the *Updater* each time the user clicks on a document by adding the triplets provided by the newly clicked document.

3.2 Similarity Measure

As we have already pointed out, the primary function of the *Estimator* component is to compute the similarity between two representations: the *Target* and the *Current Knowledge State* of the user. In the previous 3 versions of *RULK*, similarity computation relied on cosine similarity, wherein documents were transformed into vector representations. In

our case, a document is represented by a *KG*. Therefore we consider three *Graph-based measure* methods to compute the similarity.

3.2.1 Graph-based measure. Measuring the similarity between two graphs refers to calculating the resemblance between their structural representations. It is a numerical indicator of how similar two graphs are. Several measures are commonly used for this purpose, and the choice often depends on the specific application.

We adopted one of the most commonly used graph-similarity measures, the *Maximum Common Subgraph* (MCS) [4, 10, 16, 23]. It relies on the largest shared structural information between two graphs. Suppose we have two graphs, G_1 and G_2 , each representing a network of interconnected nodes and edges. To retrieve their MCS, we need to identify the largest set of nodes and edges that are present in both. The MCS-based similarity between G_1 and G_2 is calculated using the following formula:

$$MCS_Sim(G_1, G_2) = \frac{|MCS(G_1, G_2)|}{\max(|G_1|, |G_2|)}, \quad (4)$$

where $|MCS(G_1, G_2)|$ is the number of nodes belonging to the maximum common subgraph, $|G_1|$ and $|G_2|$ are the number of nodes in graphs G_1 and G_2 respectively. The larger the common subgraph, the higher the similarity score.

Daoud *et al.* [9] used the *WebJaccard* (WJ) similarity to compare two graphs, representing the user’s query and profile respectively, in an attempt to detect the user’s search boundary. Their application was based on the number of common concepts between both graphs. The *WebJaccard* similarity is in line with the perception of the “graph union” distance [33], which also involves the concept of intersection and union operations.

We use the WJ measure to quantify the similarity between the user’s target and current knowledge based on the number of entities belonging to the triplets intersection between the addressed *KGs* representations.

Given two knowledge graphs G_1 and G_2 , the *WebJaccard* similarity is calculated as follows:

$$WebJaccard(G_1, G_2) = \frac{N(G_1 \cap G_2)}{N(G_1 \cup G_2)}, \quad (5)$$

where $N(G_1 \cap G_2)$ is the number of entities belonging to the triplets in the intersection between G_1 and G_2 and $N(G_1 \cup G_2)$ is the number of entities belonging to the triplets in the union between the two graphs. A similarity value of 1 means complete identity, while a similarity value of 0 means that there are no common elements between the two graphs.

Finally, we propose a *triplet-based similarity measure*. We define the target knowledge state as the repository of information relevant to the topic the user would like to learn about—it essentially represents the user’s learning goal. The idea is that to reach the target knowledge, the users should click on documents allowing them to capture the content of the target knowledge. We would like to stress that once a triplet from the target knowledge is identical to a triplet in the current knowledge, we consider that the user successfully acquired this particular piece of the target knowledge.

$$TripletSim(G_t, G_c) = \frac{N_{identical}(G_t, G_c)}{N_{G_t}}, \quad (6)$$

where G_t and G_c are, respectively, the target and the current knowledge graphs, $N_{identical}(G_t, G_c)$ represents the number of the common identical triplets shared between G_t and G_c , and N_{G_t} represents the total number of triplets in G_t .

3.2.2 Isomorphism vs Semantic Similarity. Two objects are considered to be isomorphic when they share exactly the same structure and possess identical properties. However, it is important to note that similarity between two graphs does not always require isomorphism, as it is the case for *exact graph matching*. Two graphs with only certain

points in common are considered *somehow similar*, and this does not imply absolute identity in terms of structure and properties—*inexact graph matching* [13].

Bunke *et al.* [4] share the same perspective. They doubt the possibility of finding a perfect correspondence between graphs and support the idea of using fault-tolerant algorithms to calculate graph similarity. The same view is also shared by Riesen *et al.* [28], who propose a comparative study of exact and inexact graph matching methods.

In this paper, we consider that each user aims to acquire as much information as possible on the target knowledge document relating to a specific topic. Users search to acquire this knowledge by reading various documents semantically related to it, but whose precise content is not necessarily identical. The idea is that documents containing different entities, synonyms, and phrases may nevertheless contain the same information.

Based on that concept, we extended our similarity measures to consider not only the identical information matching but also the semantic relatedness matching. In both MCS and WJ , we consider that any triplet in the current knowledge, containing a node or entity strongly semantically related to the search topic (i.e., with a semantic similarity of 80% or higher), should be considered part of the target knowledge (because it will help reach the target knowledge that represents the topic). We measure such a similarity using *spaCy* [12], a Python open-source natural language processing library providing a built-in method to compute the semantic similarity between tokens. The resulting semantic similarity score varies between 0 and 1, where 1 represents full synonymy.

$$MCS_Sim(G_1, G_2) = \frac{|MCS(G_1, G_2)| + N_{semantic_ent}}{\max(|G_1|, |G_2|)}, \quad (7)$$

$$WebJaccard(G_1, G_2) = \frac{N(G_1 \cap G_2) + N_{semantic_ent}}{N(G_1 \cup G_2)}. \quad (8)$$

The components of both formulas remain unchanged; however, we have introduced a new variable $N_{semantic_ent}$ in the numerator. This variable represents the count of semantically related entities to the search topic extracted from the user’s current knowledge.

Also, we added the $N_{semantic_trp}$ variable to the triplet-based similarity measure (Equation 9), which represents the semantically related triplets between KGs.

$$TripletSim(G_t, G_c) = \frac{N_{identical}(G_t, G_c) + N_{semantic_trp}(G_t, G_c)}{N_{G_t}} \quad (9)$$

To our knowledge, there is no direct approach in the literature for comparing the similarity between two triplets from different graphs. To fill this gap, we transform triplets into sentences. For example, the triplet “entity:Great Recession; relation:has cause; entity:Subprime Mortgage Crisis” is transformed into “The Great Recession has cause the Subprime Mortgage Crisis”. Then, to measure the semantic similarity between two sentences representing two triplets, we used SpaCy. An 80% of semantic similarity threshold is adopted in our experiment.

4 EXPERIMENTS AND RESULTS

Our model was evaluated using the same dataset utilized in evaluating prior versions of RULK [7]. Three implementations of our model were employed: $RULK_{KG(trp)}$, $RULK_{KG(MCS)}$, and $RULK_{KG(WJ)}$ using *triplet-based*, MCS , and WJ similarity measures, respectively, to estimate the knowledge gain.

Table 1. Pearson correlation between the estimated knowledge gain of RULK versions and the actual knowledge gain of the users.

Model	ALG	RPL
$RULK_{KW}$	0.3022	0.3086
$RULK_{LM}$	0.2955	0.2923
$RULK_{NE}$	0.0931	0.118
$RULK_{KG(trp)}$	0.2651	0.2362
$RULK_{KG(MCS)}$	0.1815	0.1882
$RULK_{KG(WJ)}$	0.1475	0.1540

4.1 Dataset

The dataset was obtained from a study focusing on search-as-learning sessions, involving the participation of 126 crowd-workers tasked with conducting searches about 7 chosen topics belonging to different domains. The search system they utilized in that study was built on top of SearchX [27], an Interactive Information Retrieval research framework. This dataset incorporates also the user’s behavioral features such as query length, query count, click count, and session duration. Additionally, it includes details on the actual knowledge gain of each user, as assessed through two topic-related tests one before starting the search note vk_{pre} and the other one after finishing, note vk_{post} . The difference between these knowledge gain measurements serves to quantify both Absolute Learning Gain ALG and Realized Potential Learning RPL . Aligning with the methodologies of previous $RULK$ experiments, we used ALG and RPL values to illustrate the actual knowledge gain of the user. Details of how to calculate these metrics can be found in [7].

4.2 Model Comparison

To validate the effectiveness of the framework, we examine the correlation between the estimated gain \hat{G} and the user’s actual knowledge quantified as ALG and RPL . The outcomes presented in Table 1 indicate that our three implementations have outperformed the $RULK_{NE}$ version. That confirms our viewpoint, that is, adding the relation extraction task to named entity recognition enriches the knowledge representation, thus leading to better and improved estimation results. $RULK_{KG(MCS)}$ exhibits slightly better correlation values than $RULK_{KG(WJ)}$, standing at 0.18 for ALG and RPL . MCS and WJ demonstrate similar behavior in considering common entities between two graphs. Notably, WJ incorporates the size of the union of the two graphs in the denominator, enabling both graphs’ sizes to influence the measure. $RULK_{KG(trp)}$ exhibits the best result among our three implementations, showing a correlation score of 0.265 for ALG and 0.236 for RPL . That reveals the significance of involving the entire structure of a graph’s triplet in the similarity metric rather than prioritizing nodes over edges.

In general, the correlation results suggest that other factors not considered in the model may also contribute to the actual knowledge gain. For example, the study made by Yu *et al.* considered the users’ behavioral features in order to predict their knowledge gain [35].

4.3 Model Combinations

Multiple implementations of RULK can be combined as an interpolated estimator

$$\theta_{RULK_{comb}} = \sum_i w_i \theta_{RULK_i}, \quad (10)$$

Table 2. Pearson correlation between the estimated knowledge gain of different RULK model combinations and the actual knowledge gain of the users. The combinations are sorted by RPL correlation.

Model	ALG	RPL
$RULK_{KG} + RULK_{NE}$	0.3040	0.2904
$RULK_{LM} + RULK_{KG}$	0.3293	0.3031
$RULK_{KW} + RULK_{LM}$	0.3164	0.3192
$RULK_{KW} + RULK_{KG}$	0.3165	0.3225
$RULK_{LM} + RULK_{NE}$	0.3228	0.3309
$RULK_{KW} + RULK_{NE}$	0.3184	0.3333
$RULK_{KW} + RULK_{LM} + RULK_{KG}$	0.3332	0.3333
$RULK_{KW} + RULK_{LM} + RULK_{NE}$	0.3378	0.3490
$RULK_{LM} + RULK_{NE} + RULK_{KG}$	0.35020	0.34752
$RULK_{KW} + RULK_{KG} + RULK_{NE}$	0.35465	0.3555
$RULK_{KW} + RULK_{LM} + RULK_{NE} + RULK_{KG}$	0.36044	0.36163

where each i is a distinct implementation of RULK and the weights w_i , such that $\sum_i w_i = 1$, are set so as to optimize the result.

As suggested by previous work, combining knowledge-state representation models of different natures can boost the accuracy of knowledge gain prediction. That happens because each model covers a different and complementary aspect of knowledge. For instance, [17] pointed out that even though $RULK_{NE}$ is by far the worst in terms of gain prediction accuracy when used alone, the combinations using it outperform the combinations not using it. This observation underlines that the real interest of each model is its ability to capture aspects of knowledge that the other models fail to capture.

In this spirit, we now assess all potential model combinations, in order to determine to which degree the model we are proposing is capable of capturing an aspect of knowledge that the others do not, namely what we might call the *relational* aspect. We use the $RULK_{KG(trp)}$ implementation, which exhibits the best result among our three implementations.

Table 2 shows the ALG and RPL measures for all possible combinations of models. Although $RULK_{KG}$ ranks only third out of four when used solo, we can observe that:

- the pair combinations can be ranked as $LM + KG > LM + NE > KW + NE > KW + KG > KW + LM > KG + NE$ according to *ALG* and as $KW + NE > LM + NE > KW + KG > KW + LM > LM + KG > KG + NE$ according to *RPL*, which suggests that *KG* tends to perform better in combination with *LM* or *KW* than in combination with *NE*; indeed, *NE* and *KG* focus on similar aspects of knowledge;
- the trio combinations can be ranked as $KW + KG + NE > LM + NE + KG > KW + LM + NE > KW + LM + KG$ according to *ALG* and as $KW + KG + NE > KW + LM + NE > LM + NE + KG > KW + LM + KG$ according to *RPL*; both measures agree in ranking the combination of *KG* with *KW* and *NE* first;
- the all-model combination clearly outperforms all the other combinations and solo models, according to both measures, confirming the hypotheses that the four models are indeed complementary.

5 CONCLUSION

In this paper, we introduced $RULK_{KG}$, a graph-based representation of the RULK framework designed for tracking a user’s knowledge state during a web search session, with the aim of estimating the knowledge gain. When assessing the effectiveness of our model by measuring the correlation between the estimated and actual knowledge gain, it shows

a higher correlation value compared to $RULK_{NE}$. In addition, the results of the model combinations show that the knowledge-graph representation is complementary to the others and, in combination with them, helps obtain better correlations.

A limitation of our framework to be pointed out is the assumption that the whole document is considered when a user clicks on it, which ignores the fact that users often only read parts of documents. One solution to this problem could be to take into account the user's eye movements [2, 20].

Future work might include identifying a method to learn the user's target knowledge during a search session, providing the opportunity to evaluate $RULK$ framework against other models across diverse datasets that encompass different types of knowledge assessment tests, such as [24].

REFERENCES

- [1] Peter Bailey, Liwei Chen, Scott Grosenick, Li Jiang, Yan Li, Paul Reinholdtsen, Charles Salada, Haidong Wang, and Sandy Wong. 2012. User task understanding: a web search engine perspective. In *NII shonan meeting on whole-session evaluation of interactive information retrieval systems, Kanagawa, Japan*.
- [2] Michael Barz, Omair Shahzad Bhatti, and Daniel Sonntag. 2021. Implicit Estimation of Paragraph Relevance From Eye Movements. *Frontiers Comput. Sci.* 3 (2021), 808507.
- [3] N. J. Belkin, R. N. Oddy, and H. M. Brooks. 1982. Ask for information retrieval: Part I. background and theory. Issue 2. <https://doi.org/10.1108/eb026722>
- [4] Horst Bunke. 2000. Graph Matching : Theoretical Foundations, Algorithms, and Applications. *Algorithmica* 2000 (2000). Issue 2.
- [5] Pere-Luís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2370–2381.
- [6] Arthur Câmara, Dima El Zein, and Célia da Costa Pereira. 2022. RULK: A Framework for Representing User Knowledge in Search-as-Learning. In *Proceedings of the 2022 Conference on Design of Experimental Search & Information REtrieval Systems – DESIRES'22*.
- [7] Arthur Câmara, Nirmal Roy, David Maxwell, and Claudia Hauff. 2021. Searching to Learn with Instructional Scaffolding. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval (Canberra ACT, Australia) (CHIIR '21)*. Association for Computing Machinery, New York, NY, USA, 209–218.
- [8] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings 40*. Springer, 806–810.
- [9] Mariam Daoud, Lynda Tamine Lechani, and Mohand Boughanem. 2009. Towards a graph-based user profile modeling for a session-based personalized search. *Knowledge and Information Systems* 21 (2009). Issue 3. <https://doi.org/10.1007/s10115-009-0232-0>
- [10] Mariam Daoud, Lynda Tamine, and Mohand Boughanem. 2010. A personalized graph-based document ranking model using a semantic user profile. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6075 LNCS. https://doi.org/10.1007/978-3-642-13470-8_17
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [12] NV Dharani and Neha Vittal Annam. 2022. SPACY SIMILARITY CHECK USING NATURAL LANGUAGE PROCESSING. 4, 9 (2022).
- [13] Shri Prakash Dwivedi. 2022. Inexact Graph Matching Using Centrality Measures. *CoRR* abs/2201.04563 (2022).
- [14] Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. *Frontiers in Artificial Intelligence and Applications* 325. <https://doi.org/10.3233/FAIA200321>
- [15] Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*. <https://doi.org/10.18653/v1/2021-eacl-main.319>
- [16] Hans Christian Ehrlich and Matthias Rarey. 2011. Maximum common subgraph isomorphism algorithms and their applications in molecular science: A review. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1 (2011). Issue 1. <https://doi.org/10.1002/wcms.5>
- [17] Dima El Zein, Arthur Câmara, Célia da Costa Pereira, and Andrea Tettamanzi. 2023. RULKNE: Representing User Knowledge State in Search-as-Learning with Named Entities. In *CHIIR*. ACM, 388–393.
- [18] Dima El Zein and Célia da Costa Pereira. 2021. A Cognitive Agent Framework in Information Retrieval: Using User Beliefs to Customize Results. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12568 LNAI. https://doi.org/10.1007/978-3-030-69322-0_21
- [19] Ujwal Gadiraju. 2018. "A new age of search systems" by Ujwal Gadiraju with Martin Vesely as coordinator: understanding knowledge gained by users through search. *SIGWEB NewsL*. 2018, Summer (2018), 4:1–4:11. <https://doi.org/10.1145/3266231.3266235>

- [20] Daniel Hienert, Dagmar Kern, Matthew Mitsui, Chirag Shah, and Nicholas J. Belkin. 2019. Reading Protocol: Understanding what has been Read in Interactive Information Retrieval Tasks. In *CHIIR*. ACM, 73–81.
- [21] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems* 33 (2022). Issue 2. <https://doi.org/10.1109/TNNLS.2021.3070843>
- [22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).
- [23] A. Marcelli, S. Quer, and G. Squillero. 2019. The Maximum Common Subgraph problem: A portfolio approach.
- [24] Christian Otto, Markus Rokicki, Georg Pardi, Wolfgang Gritz, Daniel Hienert, Ran Yu, Johannes von Hoyer, Anett Hoppe, Stefan Dietze, Peter Holtz, Yvonne Kammerer, and Ralph Ewerth. 2022. SaL-Lightning Dataset: Search and Eye Gaze Behavior, Resource Interactions and Knowledge Gain during Web Search. In *CHIIR*. ACM, 347–352.
- [25] Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. STRUCTURED PREDICTION AS TRANSLATION BETWEEN AUGMENTED NATURAL LANGUAGES. *ICLR 2021 - 9th International Conference on Learning Representations*.
- [26] Peter Pirolli and Sanjay Kairam. 2013. A knowledge-tracing model of learning from a social tagging system. *User Modeling and User-Adapted Interaction* 23 (2013). Issue 2-3. <https://doi.org/10.1007/s11257-012-9132-1>
- [27] Sindunuraga Rikarno Putra, Felipe Moraes, and Claudia Hauff. 2018. SearchX: Empowering collaborative search research. *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*. <https://doi.org/10.1145/3209978.3210163>
- [28] Kaspar Riesen, Xiaoyi Jiang, and Horst Bunke. 2010. *Exact and Inexact Graph Matching: Methodology and Applications*. https://doi.org/10.1007/978-1-4419-6045-0_7
- [29] Daniel E. Rose and Danny Levinson. 2004. Understanding user goals in web search. In *WWW*. ACM, 13–19.
- [30] Chirag Shah, Ryen White, Paul Thomas, Bhaskar Mitra, Shawon Sarkar, and Nicholas J. Belkin. 2023. Taking Search to Task. In *CHIIR*. ACM, 1–13.
- [31] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. LINDEN: linking named entities with knowledge base via semantic knowledge. In *WWW*. ACM, 449–458.
- [32] Rohail Syed and Kevyn Collins-Thompson. 2017. Retrieval Algorithms Optimized for Human Learning. In *SIGIR*. ACM, 555–564.
- [33] W. D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. 2001. Graph distance using graph union. *Pattern Recognition Letters* 22 (2001). Issue 6-7. [https://doi.org/10.1016/S0167-8655\(01\)00022-8](https://doi.org/10.1016/S0167-8655(01)00022-8)
- [34] Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. <https://doi.org/10.18653/v1/2020.emnlp-main.133>
- [35] Ran Yu, Rui Tang, Markus Rokicki, Ujwal Gadiraju, and Stefan Dietze. 2021. Topic-independent modeling of user knowledge in informational search sessions. *Information Retrieval Journal* 24 (2021). Issue 3. <https://doi.org/10.1007/s10791-021-09391-7>