RULKNE: Representing User Knowledge State in Search-as-Learning with Named Entities

Dima El Zein, Arthur Câmara, Célia da Costa Pereira, Andrea G. B. Tettamanzi

RULK$_{NE}$: Representing User Knowledge State in Search-as-Learning with Named Entities

Dima El Zein
elzein@i3s.unice.fr
Université Côte d'Azur, Laboratoire I3S, CNRS, UMR 7271
Sophia Antipolis, France

Arthur Câmara
A.BarbosaCamara@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Célia da Costa Pereira
Celia.DA-COSTA-PEREIRA@univ-cotedazur.fr
Université Côte d'Azur, Laboratoire I3S, CNRS, UMR 7271
Sophia Antipolis, France

Andrea Tettamanzi
andrea.tettamanzi@univ-cotedazur.fr
Université Côte d'Azur, Laboratoire I3S, CNRS, UMR 7271
Sophia Antipolis, France

ABSTRACT

A reliable representation of the user's *knowledge state* during a *learning* search session is crucial to understand their real information needs. When a search system is aware of such a state, it can adapt the search results and provide greater support for the user's learning objectives. A common practice to track the user's knowledge state is to consider the content of the documents they read during their search session(s). However, most current work ignores entity mentions in the documents, which, when linked to knowledge graphs, can be a source of valuable information regarding the user's knowledge. To fill this gap, we extend RULK—*Representing User Knowledge in Search-as-Learning*—with entity linking capabilities. The extended framework RULK$_{NE}$ represents and tracks user knowledge as a collection of such entities. It eventually estimates the user knowledge gain—learning outcome—by measuring the similarity between the represented knowledge and the learning objective. We show that our methods allow for up to 10% improvements when estimating user knowledge gains.

KEYWORDS

Search-As-Learning, User Knowledge, Named Entities, Interactive IR, Retrieval system

ACM Reference Format:
Dima El Zein, Arthur Câmara, Célia da Costa Pereira, and Andrea Tettamanzi. 2023. RULK$_{NE}$: Representing User Knowledge State in Search-as-Learning with Named Entities. In *ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '23), March 19–23, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3576840.3578330

1 INTRODUCTION

Estimating user knowledge while they search is primordial for an effective learning-oriented search system. By doing so, such systems can provide the user with content that is relevant to their query *and* adapted to their actual knowledge level.

A common practice is to represent that state of *user knowledge* using the content of the documents they read during the search session. We recently proposed in [2] RULK—A framework for *Representing User Knowledge in Search-as-Learning*—two different types of representation: keyword-based and language-model-based, exploiting large pre-trained language models. However, neither of these two types exploits an explicit annotation of named entities (NE) and their links to resources of extensive knowledge graphs, which can be much more precise and specific in identifying what a given document is about, and, consequently, what kind of knowledge it entails.

A knowledge graph (KG) is a knowledge base where knowledge is represented as a set of entities (or *resources*), which are the vertices of a graph, connected by binary relationships (or *roles*, *properties*, *attributes*), which constitute the labeled edges of the graph. In modern approaches to information access, knowledge graphs are ubiquitous [4]. Specifically, they can be used in information retrieval to support semantic search [12].

The resources described by a knowledge graph not only allow documents to be semantically annotated, but also represent the epistemic state of a user and provide a background knowledge enriching queries and refining results.

The use of KGs to represent information about the user is not new; personal knowledge graphs organised user's personal information [1], life events [18] or profiling interests [5, 6]. The main distinctive feature of our work is what we aim to make of the information described by a KG, especially the links between entities, as a way to represent and measure users' knowledge during their search sessions (i.e., their *epistemic state*). What we propose here is to take a further step and use links to resources described by knowledge graphs not only to represent information about entities personally related to a user, but also what an information retrieval system may presume its user already knows (i.e., what we might call the user's *epistemic state*), based, for instance, on past interaction, documents that the user has read, or direct feedback from the user.

More specifically, we propose *another* instantiation of RULK that leverages the coverage of a large KG as a measure of user knowledge. Such entity-based representation, which we name RULK$_{NE}$, is a highly structured and rich representation that promises to be more accurate and detailed than a representation based simply on

CHIIR '23, March 19–23, 2023, Austin, TX, USA

ACM ISBN 979-8-4007-0035-4/23/03…$15.00
https://doi.org/10.1145/3576840.3578330

keywords or latent semantic components. Therefore, in this paper, we answer the following research questions:

RQ1 Are the estimated knowledge gains produced by RULK_{NE} correlated to actual users' knowledge?

RQ2 How does RULK_{NE} compare to other existing RULK implementations?

Our framework has two main requirements: (1) a way to recognize named entities in a text and link them to a knowledge graph, and (2) a similarity measure among collections of linked named entities allowing one to estimate the knowledge gain provided by a set of documents to a user. We empirically compare our proposal to a learning task's keyword- and language model-based representations.

2 BACKGROUND

In this section, we first give a brief overview of our previously proposed RULK framework [2]. Secondly, we quickly introduce knowledge graphs (KGs), and how they connect to our new proposal.

2.1 The RULK Framework

The framework proposed in [2] is composed of three interacting components: **Feature Extractor** ($\boldsymbol{\gamma}$), **Updater** ($\boldsymbol{\sigma}$) and **Estimator** ($\boldsymbol{\theta}$).

As users interact with the search results,[1] they learn about a given topic of interest. RULK tracks the user's knowledge through an internal state represented by a vector *current knowledge state* $\vec{c_{ks}}$. The search system that instantiates RULK is assumed to have access to a *target knowledge*, or reference document, covering the "ideal" knowledge to be acquired regarding a specific topic T. We consider a user's search need—or learning objective—is represented in this document. For the sake of simplicity, we also make the assumption that the user has no previous knowledge about T. Figure 1 shows an overview of the framework RULK as presented in [2].

Feature Extractor ($\boldsymbol{\gamma}$). A *Feature Extractor* is a component that, given a document d,[2] *encodes* it into a fixed-length representation, using a method like TF-IDF, Word2Vec or any other encoding method. $\boldsymbol{\gamma}$ encodes a document read by the user into $\vec{v_d}$, and the *reference* document into $\vec{t_{ks}}$.

Updater ($\boldsymbol{\sigma}$). When the user reads a new document, they acquire new knowledge that is added to the previous ones. $\boldsymbol{\sigma}$ updates the current state of the knowledge $\vec{c_{ks}}$ with the new information $\vec{v_d}$. The updated knowledge state $\vec{c'_{ks}}$ is the result of combining the user's current state $\vec{c_{ks}}$ to the $\vec{v_d}$: $\vec{c'_{ks}} = \sigma(\vec{c_{ks}}, \vec{v_d})$.

Estimator ($\boldsymbol{\theta}$). To *estimate* the user's knowledge gain—or learning outcome—on a specific topic during a session, the *Estimator* $\boldsymbol{\theta}$ compares the user's current knowledge state $\vec{c_{ks}}$ to the target knowledge state $\vec{t_{ks}}$: $\widetilde{G} = \theta(\vec{c_{ks}}, \vec{t_{ks}})$, where $\widetilde{G}$ is an estimation of the user's knowledge gain in the session and $\boldsymbol{\theta}$ is a similarity function (e.g., cosine similarity). The intuition behind $\boldsymbol{\theta}$ is that the user, by progressing in their session, "moves" their knowledge state toward the target. As both vectors are in the same embedding space, the similarity between them provides an estimation of how close the user is to acquiring the "ideal" knowledge.

[1] A search result can be a web pages, videos, online courses, etc. In this paper, a "document" refers to a search result.

[2] For the sake of simplicity, we assume that d is always a text-only page.

2.2 Keyword and Language Model Implementations of RULK

We briefly present the Keyword (KW) and Language Model (LM) implementations previously proposed in [2].

RULK_{KW}. The *target knowledge* in this implementation is represented as a set of n keywords that the user must read to fulfill their search need. The $\boldsymbol{\gamma}$ component encodes the *reference* document into $\vec{t_{ks}}$ containing the occurrence count number of the n keywords. Those numbers represent the number of times the user has to read a specific keyword to fulfill the need. Similarly, the clicked documents are encoded into a vector containing the frequency of n keywords. To update the user knowledge, $\boldsymbol{\sigma}$ adds the keywords' count in the documents to the one in the knowledge state. The user knowledge is assumed to increase monotonically. The $\boldsymbol{\theta}$ component estimates the user knowledge gain by calculating the cosine similarity between $\vec{c_{ks}}$ and $\vec{t_{ks}}$.

RULK_{LM}. In this implementation, both the $\vec{t_{ks}}$ and $\vec{v_d}$ are a BERT-embedding of fixed length m. Given a document d (or, conversely, a *reference* document) with k sentences $\{s_1, s_2 \dots s_k\}$, $\boldsymbol{\gamma}$ generates, for each sentence s_i, an embedding of size m given by:

$$\vec{v_{s_i}} = BERT([CLS]; s_{i:l}; [SEP]), \tag{1}$$

where ; is a concatenation, l the maximum input size of the model and $[CLS]$ and $[SEP]$ are special BERT tokens. $\vec{v_d}$ (conversely, $\vec{t_{ks}}$) is then given by an element-wise sum over all $\vec{v_{s_i}}$. The Updater $\boldsymbol{\sigma}$ is then a simple element-wise sum over all elements of $\vec{v_d}$ and $\vec{c_{ks}}$. The cosine similarity is also used in $\boldsymbol{\theta}$ to estimate the knowledge gain.

2.3 Knowledge Graphs

A knowledge graph is a knowledge base that uses a graph-structured data model to integrate data. Knowledge graphs are often used to store interlinked descriptions of entities—objects, events, situations or abstract concepts—while also encoding the semantics underlying the used terminology.

In this paper, we will assume that knowledge graphs follow a standard and technical infrastructure like the ones provided by the W3C for the Semantic Web, i.e.: OWL (the Web ontology language, based on description logics) based on the underlying data model RDF (the resource definition standard). This would allow a practical implementation of our proposal using state-of-the-art knowledge engineering technologies.

The basic statement of RDF is a *triple* $\langle s, p, o \rangle$, where s is called the subject, p the predicate, and o the object. The subject of a triple is a *resource* (or, in other words, an *entity*), represented by an internationalized resource identifier (*IRI*), like <http://example.org/resource/LHR>; the (binary) predicate represents a *property* of the subject, denoted by an IRI, like <http://example.org/ontology#city> or <http://example.org/ontology#iataCode>; the object, which represents a value of that property, may be a resource, denoted by an IRI, like <http://example.org/resource/London>, or a data value,

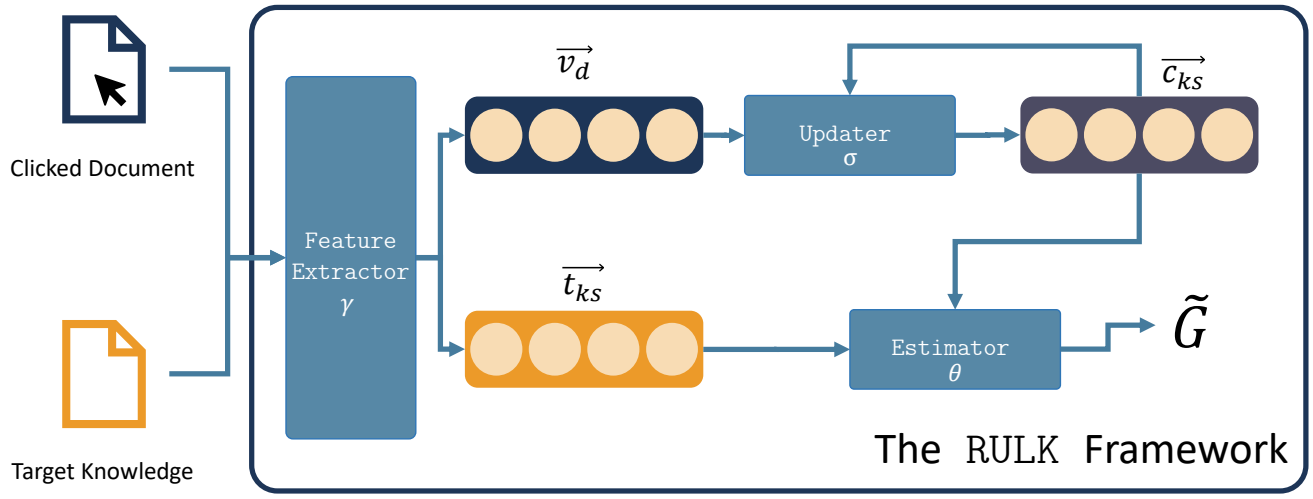

Figure 1: The RULK framework and its main components, extracted from [2]. At first, a clicked document d is transformed into $\vec{v_d}$ by γ. The *target knowledge document* is also encoded into $\vec{t_{ks}}$. Next, σ updates the current state $\vec{c_{ks}}$ with $\vec{v_d}$. Finally, θ compares $\vec{c_{ks}}$ to a target knowledge $\vec{t_{ks}}$ to get an estimation of the user's knowledge gain ($\widetilde{G}$) in the session.

such as a string, a number, or a date, denoted by a *literal*, like "LHR", 2, or 07:30. In addition, the subject and object can be so-called *blank nodes*, which correspond to anonymous resources and can be understood as a kind of existentially quantified variables.

For the sake of readability and conciseness, when several IRIs share the same base, a prefix may be defined, for instance

```
@prefix   :    <http://example.org/resource/> .
@prefix   o:   <http://example.org/ontology#> .
```

and the IRIs may then be abbreviated as :LHR instead of the full <http://example.org/resource/LHR> or o:iataCode instead of the full <http://example.org/ontology#iataCode>.

An important thing to observe is that, behind an IRI, which is essentially an identifier, many different notions can hide, like an instance (i.e., a constant, what is called an *individual name* in description logics), a binary predicate (i.e., a binary relation, what is called a *role* in description logics and a *property* in OWL), or a concept (i.e., a unary predicate, called a *class* in OWL). It is exactly this uniform naming convention that makes RDF so flexible and versatile. Thus, for instance, an assertion of the form $C(a)$, where C is a unary predicate (a concept) and a is the name of an individual, may be represented as an RDF triple $\langle a, \text{rdf:type}, C\rangle$, which may be paraphrased as "a is an instance of C", thanks to the rdf:type relation, and an assertion of the form $R(a, b)$, where R is a binary predicate and a and b are entity names, may be represented as an RDF triple $\langle a, R, b\rangle$, for example

```
:LHR o:city :London .
```

which may be paraphrased as "the city of the Heathrow Airport is London".

A collection of RDF triples, which may represent assertions and other OWL axioms with a uniform syntax, may be regarded as a knowledge base under the open-world assumption, from which other triples can be deduced using an inference engine called an OWL *reasoner*.[3] Furthermore, a collection of RDF triples intrinsically represents a directed multi-graph (an RDF graph), whose vertices are resources; every triple $\langle s, p, o\rangle$ then represents an arc of type p from vertex s to vertex o.

3 A NAMED-ENTITY-BASED IMPLEMENTATION OF RULK

We propose a novel instantiation of RULK using named entities (NE) to represent both the internal and target knowledge states. We will call RULK$_{NE}$ this *Named-Entity*-based variant of the framework.

The Feature Extractor $\boldsymbol{\gamma}$ here produces a collection of links to knowledge graph resources, corresponding to named entities. A collection $K_{t_{ks}}$ is extracted from the *reference* document and another K_d for every visited document d. Producing a collection of such links from a document requires a reference knowledge graph, to be used as background knowledge, as it were, and two NLP tasks to be carried out, namely (i) named entity recognition (i.e., spotting chunks of text that are likely to refer to specific entities such as people, places, organizations, etc.), and (ii) entity linking (i.e., establishing a link between a recognized named entity and a resource defined in the reference knowledge graph). These two tasks can be challenging, but in recent years, a wide range of emerging tools can accomplish them with acceptable performance, albeit not always perfectly. This enables one to visualize what we are suggesting. One can only foresee that these tools will be improved and new, even better tools will become available in a near future, thus contributing to making RULK$_{NE}$ more and more accurate. The reference knowledge graph can be any of the large general-purpose RDF datasets available in the Linked Open Data cloud, like DBpedia, Yago, or Wikidata. We encode the target knowledge as a vector $\vec{t_{ks}}$ of the counts of the occurrences of the 10 most common entities $K_{t_{ks}}$. Each document is then encoded as a vector $\vec{v_d}$ containing the counts of the top-10 entities $K_{t_{ks}}$ in d.

[3] Examples of popular OWL reasoners are Fact++, HermiT, and Pellet

	Total	Mean	Median
Number of users per topic	126	18.14 ± 2.79	19.0
Number of topics	7	-	-
Number of queries	1095	8.62 ± 6.47	7.0
Number of documents clicked	2116	16.66 ± 8.85	16.0
Number of snippets seen	15184	119.56±72.43	105.0
Documents Clicked per query	-	2.78 ± 2.50	2.11
Session duration (minutes)	-	56.18 ± 14.58	54.05
Document dwell time (seconds)	-	79.94 ± 69.77	60.0
Pre-test scores (vks^{pre})	-	1.07 ± 1.60	0.00
Post-test scores (vks^{post})	-	6.21 ± 4.09	6.00
Actual Learning Gain (ALG)	-	0.53 ± 0.38	0.50
Realised Potential Learning (RPL)	-	0.28 ± 0.20	0.25

Table 1: Statistics, per user, extracted from the dataset used by Camara et al. [3].

The Updater $\boldsymbol{\sigma}$ is then a simple addition of the two count vectors $\vec{c_{ks}}$ and $\vec{v_d}$.

Finally, the Estimator $\boldsymbol{\theta}$ should compute the similarity between the encoded current knowledge state and the encoded target knowledge state. That is done using the cosine similarity:

$$\widetilde{G} = \frac{\vec{c_{ks}} \cdot \vec{t_{ks}}}{|\vec{t_{ks}}|\,|\vec{c_{ks}}|}. \tag{2}$$

4 EXPERIMENT

4.1 Dataset

We test our proposed framework $\mathsf{RULK_{NE}}$ on the same dataset previously used in [2]. The dataset[4] originates from the study by Camara et al. [3] and logs search-as-learning sessions. The dataset was collected with a search system implemented on top of SearchX [11], a framework for Interactive Information Retrieval research. We also show some statistics about the dataset in Table 1.

The dataset contains the interaction logs of 126 crowd-workers. At the start of the study, the system measured the user's previous knowledge vks^{pre} on a specific topic, then users were given 45 minutes to perform their search about the topic. Finally, at the end of the search session, the user's knowledge was measured again vks^{post}. The logged interactions included behavioural features, issued queries, and clicked documents.

The dataset contained 1107 unique clicked documents. We were retrieved their related texts using a digital archive—The Wayback Machine[5]—of the World Wide Web at the time the study was conducted (August 2020).

[4]The data is available at https://github.com/ArthurCamara/CHIIR21-SAL-Scaffolding
[5]https://archive.org/web

4.2 Actual Knowledge Gain Measurement

The self-reported knowledge, vks^{pre} and vks^{post}, were measured with a *Vocabulary Knowledge Scale (VKS)* test [15, 17], a commonly used method to measure user knowledge [10, 13, 14, 16]. For that, users were presented with a 4-point scale questionnaire, asking about their familiarity with ten topic-related terms.

The user's learning during their session could therefore be measured by computing the difference between vks^{pre} and vks^{post}. The learning measures is defined as follows:

$$\begin{aligned} ALG &= \frac{1}{10}\sum_{i=1}^{10} max(0, vks^{post}(v_i) - vks^{pre}(v_i)) \\ MLG &= \frac{1}{10}\sum_{i=1}^{10} 2 - vks^{pre}(v_i) \\ RPL &= \frac{ALG}{MLG} \end{aligned} \tag{3}$$

where $vks(v_i)$ the score of the user for the i-th term. *ALG* is the *Absolute Learning Gain* and *MLG* is the *Maximum Learning Gain* (i.e., the maximum amount of *new* knowledge a user can acquire, given what they already know). As for *RPL*, it represents the fraction of knowledge the user acquired from the total knowledge they could obtain in their session. In this paper, we use *actual knowledge gain* interchangeably to refer to RPL and ALG.

4.3 Target knowledge

The topics used in the original user study came from the list of topics used in the CAR track from TREC 2018 [7]. In that track, each topic is the title of a Wikipedia article from a 2018 dump. Our experiment uses these Wikipedia texts, from the same 2018 dump as the original paper, as "reference documents" for generating the target knowledge state $\vec{t_{ks}}$. It is also worth mentioning that, in the user study which originated this dataset, the mentioned work filtered Wikipedia and clones of Wikipedia from the search results during the user study. Therefore, no document proposed to the user in the page results of the experiment came from Wikipedia or a similar page.

4.4 Implementation Details

Entity Recognition. We detect the *named entities* using the *Spacy* [8] Python library. We chose the English pipeline *en_core_web_sm*, which is trained on written web text (blogs, news, comments).

Entity Linking. We automatically annotate the texts with DBpedia resources using the *dbpedia_spotlight* tool [9].

Similarity Calculation and Comparison of results. As discussed in Section 3, the user's estimated knowledge gain $\widetilde{G}$ is calculated as the cosine similarity between the tracked knowledge $\vec{c_{ks}}$ and the target knowledge state $\vec{t_{ks}}$. To assess the validity of the framework (RQ1), we measure the correlation between the estimated gain $\widetilde{G}$ and the actual user's knowledge (*ALG* and *RPL*).

Baseline. We compare our approach against previous work [2], in which knowledge states were represented by keywords $\mathsf{RULK_{KW}}$ and large language models $\mathsf{RULK_{LM}}$. These two representations are briefly described above, in Section 2.2. The previous approaches

Method	ALG	RPL
RULK$_{KW}$	0.3022	0.3086
RULK$_{LM}$	0.2955	0.2923
RULK$_{KW+LM}$	**0.3164**	**0.3192**
RULK$_{NE}$	0.0931	0.1185
RULK$_{NE+KW}$	0.3184	0.3333
RULK$_{NE+LM}$	0.3228	0.3309
RULK$_{NE+LM+KW}$	**0.3378**	**0.3490**

Table 2: Pearson correlation between the estimated knowledge gain of a given RULK implementation and the actual knowledge gain. bold values indicate the best correlation against a learning metric.

achieved the best performance when using a combination of keywords and language models, RULK$_{KW+LM}$. The best correlation between the previously estimated gain and the actual gain was 0.3164 against *ALG* and 0.3192 against *RPL*.

RULK *mixed approaches.* We also test a combination of our proposed instantiation RULK$_{NE}$ with the previous approaches RULK$_{LM}$, RULK$_{KW}$, and RULK$_{KW+LM}$. In such combinations, each instantiation has the potential to contribute to the overall estimation of the knowledge gain by capturing some specific characteristics of the text documents. The mixed approach is defined by an interpolated estimator $\boldsymbol{\theta}$, parameterized by α and β, defined as follows:

$$\theta_{\text{RULK}_{\text{LM+KW+NE}}} = \alpha\widetilde{G}_{\text{RULK}_{\text{LM}}} + \beta\widetilde{G}_{\text{RULK}_{\text{KW}}} + (1-\alpha-\beta)\widetilde{G}_{\text{RULK}_{\text{NE}}}, \quad (4)$$

where $\widetilde{G}_{\text{RULK}}$ is the estimated knowledge gain according to the respective RULK implementation.

5 RESULTS AND DISCUSSION

To answer our first research question (RQ1), we test the validity of the RULK$_{NE}$ by computing the Pearson correlation between the *actual* knowledge gain of a user and its *estimated* knowledge gain $\widetilde{G}$, as measured by the estimator $\boldsymbol{\theta}$. We then answer the second research question (RQ2) by comparing the reported correlation to the ones of the other implementations. Table 2 shows a comparison between implementations involving *named entity* NE representations with the baseline.

The estimated knowledge gain resulting from the proposed approach RULK$_{NE}$ reported a correlation of 0.0931 with the ALG and 0.118 with RPL. While the results of NE alone may look disappointing, it is interesting to notice that all the combinations that include NE outperform all the ones that do not include it. This is clear evidence that the proposed representation is complementary to the others, i.e., capable of capturing something that the others miss. Indeed, the difference between keywords and entities such as those that are stored in a knowledge graph is that keywords in general correspond to individual words, whereas entities correspond to concepts or instances of concepts whose lexicalization may involve phrases. The word embedding produced by the language model too, like keywords, operates at the level of individual words, although, by taking context into account, it can be able to distinguish different meanings of the same word or merge different words in the same meaning. Entities, however, when they are successfully recognized and linked to background knowledge, are much more precise because they are capable of designating very specific concepts. Consider for example two texts like "the President of the United States was in Manchester" and "the President of Manchester United is in the States": after eliminating the stop words, the two texts contain the same keywords (Manchester, President, States, United); in terms of entities, however, they differ: [President of the United States] and [United States] are in the former, while [Manchester United (football team)] is in the latter.

		RULK$_{KW+LM}$	RULK$_{NE+LM}$	RULK$_{NE+KW}$	RULK$_{NE+LM+KW}$
ALG	α	0.44	0.82	-	0.44
	β	0.66	-	0.86	0.41
	$1-\alpha-\beta$	-	0.18	0.14	0.150
RPL	α	0.38	0.80	-	0.39
	β	0.62	-	0.84	0.44
	$1-\alpha-\beta$	-	0.20	0.16	0.169

Table 3: Comparing parameters with the different mixture models.

Table 3 shows the parameters of the mixed models, which are optimized according to the ALG and RPL metrics, respectively. It is intriguing that, while the combinations that involve NE are those that perform best, the weight of NE in those combinations is lower than the weight of the other two implementations, varying between 14% and 20%, meaning a lower contribution to the estimation of the user's knowledge gain.

6 CONCLUSION

We have proposed an implementation of the RULK framework using named entities to represent the user's knowledge state and the contents of the documents. The knowledge gain - also said learning outcome - was estimated as the similarity between the tracked user's knowledge state and a target knowledge state expressing the learning objective. The estimated knowledge gain resulting from this framework was compared to the actual user's knowledge gain reported in a search-as-learning study. Our experiments suggest that such representation alone does not lead to an accurate estimate of the knowledge gains. However, by combining this approach to previously proposed ones (namely keyword and language model based), the results outperformed the state-of-the-art baseline by 10%. We proved that named entities representations are complementary to the other representations.

A promising research direction is to extend this proposal to include relational knowledge in addition to entities, in an attempt to fully exploit the power of knowledge graphs.

REFERENCES

[1] Krisztian Balog and Tom Kenter. 2019. Personal knowledge graphs: A research agenda. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 217–220.

[2] Arthur Câmara, Dima El Zein, and Célia da Costa Pereira. [n. d.]. RULK: A Framework for Representing User Knowledge in Search-As-Learning. In *Third International Conference on Design of Experimental Search & Information REtrieval Systems (DESIRES'22)*.

[3] A. Câmara, Nirmal Roy, David Maxwell, and Claudia Hauff. 2021. Searching to Learn with Instructional Scaffolding. *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval* (2021).

[4] Jeffrey Dalton and Laura Dietz. 2013. Constructing Query-Specific Knowledge Bases. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction* (San Francisco, California, USA) *(AKBC '13)*. Association for Computing Machinery, New York, NY, USA, 55–60. https://doi.org/10.1145/2509558.2509568

[5] 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, 3 (2009), 365–398.

[6] Mariam Daoud, Lynda Tamine, and Mohand Boughanem. 2010. A personalized graph-based document ranking model using a semantic user profile. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 171–182.

[7] Laura Dietz, Ben Gamari, Jeff Dalton, and Nick Craswell. 2018. TREC Complex Answer Retrieval Overview. In *Trec (NIST Special Publication, Vol. 500-331)*. National Institute of Standards and Technology (NIST).

[8] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[9] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*. 1–8.

[10] Heather L. O'Brien, Andrea Kampen, Amelia W. Cole, and Kathleen Brennan. 2020. The Role of Domain Knowledge in Search as Learning. In *Chiir*. Acm, 313–317.

[11] Sindunuraga Rikarno Putra, Felipe Moraes, and Claudia Hauff. 2018. SearchX: Empowering Collaborative Search Research. In *Sigir*. Acm, 1265–1268.

[12] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2020. Knowledge Graphs: An Information Retrieval Perspective. Preprint. , 153 pages. url: https://staff.fnwi.uva.nl/m.derijke/wp-content/papercite-data/pdf/reinanda-2020-knowledge.pdf.

[13] Nirmal Roy, Felipe Moraes, and Claudia Hauff. 2020. Exploring Users' Learning Gains within Search Sessions. *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval* (2020).

[14] Sara Salimzadeh, Ujwal Gadiraju, Claudia Hauff, and Arie van Deursen. 2022. Exploring the Feasibility of Crowd-Powered Decomposition of Complex User Questions in Text-to-SQL Tasks. In *Ht*. Acm, 154–165.

[15] Katherine Anne Dougherty Stahl and Marco A. Bravo. 2010. Contemporary Classroom / Vocabulary Assessment / for Content Areas. *The Reading Teacher* 63 (2010), 566–578.

[16] Rohail Syed and Kevyn Collins-Thompson. 2017. Optimizing search results for human learning goals. *Information Retrieval Journal* 20, 5 (2017), 506–523.

[17] Marjorie Bingham Wesche and T. Sima Paribakht. 1996. Assessing Second Language Vocabulary Knowledge: Depth Versus Breadth. *Canadian Modern Language Review-revue Canadienne Des Langues Vivantes* 53 (1996), 13–40.

[18] An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. Personal knowledge base construction from text-based lifelogs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 185–194.