Web Semantics: Science, Services and Agents on the World Wide Web 75 (2023) 100759

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

LaSER: Language-specific event recommendation

Sara Abdollahi [a,*], Simon Gottschalk [a], Elena Demidova [b]

[a] L3S Research Center, Leibniz Universität Hannover, Germany
[b] Data Science & Intelligent Systems Group (DSIS), University of Bonn, Germany

ARTICLE INFO

Article history:
Received 30 November 2021
Received in revised form 16 May 2022
Accepted 13 September 2022
Available online 17 September 2022

Keywords:
Knowledge graphs
Event recommendation
Language-specific recommendation

ABSTRACT

While societal events often impact people worldwide, a significant fraction of events has a local focus that primarily affects specific language communities. Examples include national elections, the development of the Coronavirus pandemic in different countries, and local film festivals such as the *César Awards* in France and the *Moscow International Film Festival* in Russia. However, existing entity recommendation approaches do not sufficiently address the language context of recommendation. This article introduces the novel task of language-specific event recommendation, which aims to recommend events relevant to the user query in the language-specific context. This task can support essential information retrieval activities, including web navigation and exploratory search, considering the language context of user information needs. We propose *LaSER*, a novel approach toward language-specific event recommendation. *LaSER* blends the language-specific latent representations (embeddings) of entities and events and spatio-temporal event features in a learning to rank model. This model is trained on publicly available Wikipedia Clickstream data. The results of our user study demonstrate that *LaSER* outperforms state-of-the-art recommendation baselines by up to 33 percentage points in MAP@5 concerning the language-specific relevance of recommended events.

1. Introduction

Event and entity recommendation are critical tasks facilitating vital applications such as web navigation and exploratory research of a topic of user interest [1]. Finding relevant events is an increasingly difficult task in the global digital world, where event relevance is highly dependent on the language context of the users and their information needs. However, state-of-the-art event and entity recommendation approaches typically neglect this relevance dimension and provide results that do not adequately consider event-specific properties and language context.

Table 1 provides examples of events particularly relevant to the Coronavirus pandemic from the perspective of the German, Italian, and Spanish-speaking audience on Wikipedia. This example is created based on the number of clicks in the Wikipedia Clickstream dataset [2] that provides the click-through rates for the Wikipedia articles in the respective Wikipedia language edition. As we can observe, according to the clickstream, German Wikipedia users are mainly interested in the Coronavirus outbreaks in the German-speaking countries and the recession caused by the pandemic. Italian users are mainly interested in the pandemic in Italy and the SARS outbreak, a similar event in 2002, followed by the development in the US. The Spanish Wikipedia reflects user interests in several Spanish-speaking countries, such as Argentine, the US, and Colombia. These observations illustrate how event relevance varies based on the language-specific user context.

In this article, we present the new task of *language-specific event recommendation*. This task adds two critical dimensions to entity recommendation: (i) recommendation of *events* of societal importance, including political elections, military conflicts and sports events and (ii) the *language-specific context* of these events. These dimensions are essential in various application scenarios, including event recommendation in information retrieval, event analytics to understand cultural viewpoints [3] and perception of events in different cultures [4].

Language-specific event recommendations open up new web navigation and exploratory search opportunities and can assist users in researching events relevant to a topic in specific languages. Examples include historians researching the Second World War in different countries and journalists exploring the perception of the Coronavirus pandemic in different language communities. Language-specific event recommendations can help address such information needs and provide recommendations that better fit users' interests and linguistic backgrounds.

The most relevant task addressed in the literature in the context of this article is entity recommendation, typically defined as

* Corresponding author.
E-mail addresses: abdollahi@L3S.de (S. Abdollahi), gottschalk@L3S.de (S. Gottschalk), elena.demidova@cs.uni-bonn.de (E. Demidova).

https://doi.org/10.1016/j.websem.2022.100759

Table 1
Events with the highest number of clicks in the German, Italian and Spanish Wikipedia language editions starting from the article *Coronavirus pandemic* in April 2021 (#: The number of user clicks in the respective Wikipedia language edition).

Rank	German	#	Italian	#	Spanish	#
1	COVID-19 pandemic in Germany	3,775	COVID-19 pandemic in Italy	2,890	COVID-19 pandemic in Argentina	1,452
2	COVID-19 recession	1,852	2002–2004 SARS outbreak	780	COVID-19 pandemic in the United States	1,286
3	COVID-19 pandemic in Austria	1,072	COVID-19 pandemic in the United States	489	COVID-19 pandemic in Colombia	1,105

the problem of suggesting entities relevant in a particular context, mostly provided as an entity of interest [5]. Entity recommendation has been tackled from different perspectives, including time-aware entity recommendation [6] and personalized recommendation of social events [7]. Focusing on the language-specific event recommendation, we add novel dimensions to this task. We go beyond existing approaches that do not consider language-specific aspects and mainly optimize for entity popularity in general. Furthermore, in contrast to existing work, we train our model on publicly available data instead of relying on proprietary click or search logs typically used in the literature (e.g., [5,8]), enhancing the reproducibility of our results.

In this article, we present *LaSER* – a new method for **La**nguage-**S**pecific **E**vent **R**ecommendation. Given a query entity of user interest (e.g., a person like *Winston Churchill*, an event like *Coronavirus pandemic*, or a concept like *Film Festival*) and a language of interest, *LaSER* returns a list of events relevant to the query entity and the language community. With *LaSER*, we tackle two key challenges of the language-specific event recommendation:

(C1) The creation of methods for language-specific event recommendation requires consideration of the language context, including latent properties of events and their relations in this context, along with the spatial and temporal dimensions. To the best of our knowledge, language-specific context has not been considered in state-of-the-art entity recommendations.

(C2) Training and evaluation of the models for language-specific event recommendation require corpora reflecting events users consider relevant in the specific language context. However, existing corpora do not contain information regarding language-specific user needs. Furthermore, datasets used for training recommendation models are often proprietary (e.g., the Yahoo! search logs [5] and Baidu Web search engine logs [8]). Recommendation methods based on such proprietary corpora are barely reproducible.

To tackle these challenges in *LaSER*, we derive and utilize event-specific and language-specific characteristics and include them in a language-specific recommendation model (C1), use freely available datasets, and collect high-quality user relevance judgments (C2). *LaSER* is based on language-specific latent representations (embeddings) of entities and events in a language-specific knowledge graph representing the relevance of entity and event relations in different language contexts. We combine these latent representations with spatio-temporal event features and utilize them for training a learning to rank (LTR) model. Given a language of interest and a query entity, this model generates a ranked list of relevant events. We train the model using the publicly available Wikipedia Clickstream.

We evaluate the effectiveness of *LaSER* in two different setups. First, we evaluate the *LaSER* ability to predict language-specific clicks between entities and events in the Wikipedia Clickstream. The results demonstrate that our model outperforms link-based, embedding-based and graph attention network based ranking baselines by over 8 (nDCG@10) and 17 (MAP@10) percentage points on average. Second, we conduct a user study to evaluate the relevance of recommended events and analyze different relevance criteria. The results confirm that *LaSER* outperforms the baselines by up to 33 percentage points in MAP@5 concerning the language-specific relevance.

We make our source code and data publicly available to facilitate reproducibility of the results and their reuse by the research community.[1]

Contributions. In summary, our contributions presented in this article are as follows:

- We define the new task of language-specific event recommendation. This task is different from the existing recommendation tasks that focus on the individual user preferences, provide language-independent recommendations, and do not focus on the language-specific relevance and event characteristics.
- We represent the language-specific context through a set of novel features, including spatio-temporal event information, language-specific link data, and publicly available clickstream data that serve as target labels. We blend these features into an architecture for language-specific event recommendations. This architecture relies on the language-specific entity and event embeddings for candidate retrieval and an established learning to rank model.
- We propose novel language-specific embeddings where latent entity representations reflect their neighborhoods and relations in a language-specific knowledge graph and demonstrate that they are beneficial for the candidate generation.
- We conduct extensive experiments on real-world data and a user study and demonstrate that our approach outperforms state-of-the-art recommendation methods.

The remainder of this article is structured as follows: First, we define the task of language-specific event recommendation in Section 2. We present our proposed approach in Section 3 and introduce datasets used as background knowledge in Section 4. Following that, in Section 5 we describe our evaluation aims and setup. Sections 6–8 present the results of the ranking evaluation and a user study and discuss anecdotal results and application scenarios. Section 9 provides an overview of related work. Finally, we provide a conclusion in Section 10.

2. Problem statement

This section defines the notions of a language-specific knowledge graph, entities, events, and the task of language-specific event recommendation addressed in this article.

[1] Code: https://github.com/saraabdollahi/LaSER, Data: https://zenodo.org/record/5735580

To facilitate recommendation, we introduce a language-specific, which models entities, events, and relations in a language context.[2]

Definition 1. A **language-specific knowledge graph** is a directed graph $G = (E, R, L)$ whose nodes E represent a set of real-world entities (e.g., persons, places and events), connected via edges $R \subset E \times E$. L is a set of languages.

In the context of language-specific recommendations, relevant spatio-temporal features are locations and dates associated with entities.

Definition 2. An entity $e \in E$ can be assigned a start and end time $[e.t_s, e.t_e]$ as well as a set of coordinate pairs $e.C$, where each coordinate pair $c \in e.C$ consists of latitude and longitude: $c = (lat, lon)$, $lat \in \mathbb{R}$, $lon \in \mathbb{R}$.

For example, the *Summer Olympics 2012* happened from July 27 to August 12, 2012, and are assigned multiple coordinate pairs reflecting different sports venues in London. The entity representing *Winston Churchill* is assigned his birth and death dates (November 30, 1874, to January 24, 1965) and a set of coordinate pairs referring to essential places in his life (e.g., of the Blenheim Palace, his birthplace).

In the context of the language-specific knowledge graph, events are a subset of entities. Whereas many definitions of an event exist in the literature, in this work, we follow an event definition by J. Allan et al. proposed in the context of the event detection and tracking within news stories [9]:

Definition 3. An event $v \in \mathcal{V} \subset E$ is something that happened at a particular time and place.

Examples of events are the *Summer Olympics 2012*, the *fire at the Notre Dame* in 2020 and the *Coronavirus pandemic in Germany* starting in 2020. For ongoing events like the *Coronavirus pandemic*, the end date is not yet known.

Having introduced the entities, events, and their relations, we can now define the task of language-specific event recommendation.

In this article, given an entity of user interest referred to as a query entity, we address the new task of recommending relevant events for this entity in a specific language context. Note that a query entity can represent a real-world entity or an event.

Definition 4. Given a query entity $e \in E$, a language l and the language-specific knowledge graph $G = (E, R, L)$, the task of **language-specific event recommendation** is to create a ranked list $S_{e,l} = \langle v_1, \ldots v_n \rangle$ of events ($v_i \in \mathcal{V}, i \in 1, \ldots, n$). The events in $S_{e,l}$ are sorted in descending order regarding their relevance to the query entity e for the audience speaking the language l.

For example, consider the recommendation example in Table 1 created from the click counts on Wikipedia articles in specific Wikipedia language editions in April 2021. For the query entity *Coronavirus pandemic* (e) and the German language (l), this method returns a list $S_{e,l}$ of recommended events ⟨ *COVID-19 pandemic in Germany*, *COVID-19 recession*, *COVID-19 pandemic in Austria*⟩. Language-specific event recommendation generates a ranked list of events. The query entity may be any node in the language-specific knowledge graph.

[2] Note that in this work we follow a language-specific view, i.e., we do not further distinguish between different sub-communities speaking the same language (e.g., the different English-speaking sub-communities).

3. The *LaSER* approach

In this article, we present *LaSER*, a new method for language-specific event recommendation. Fig. 1 provides an overview of the *LaSER* components. *LaSER* consists of a training and a query phase. These phases rely on background knowledge that includes the language-specific knowledge graph and language-specific click data.

In the pre-processing training phase, we first create language-specific embeddings based on the language-specific knowledge graph. In addition, we train a learning to rank model that learns from language-specific click data. This model uses feature values extracted from the language-specific knowledge graph, i.e., event characteristics, as well as the relationships between events and entities.

In the query phase, given an input query entity $e \in E$ and a language $l \in L$, we use the embeddings and the trained LTR model to generate a ranked list of events.

Fig. 2 provides a concrete example of how a query is processed, with a specific focus on the feature extraction step. Here, World War II is the query entity e, with Russian as the query language l. From the candidate generation, we obtain two events[3]: "Events in Poland in September 1939" and "Battles in the Janowska forests". From the links and the spatio-temporal information in the language-specific knowledge graph, *LaSER* extracts feature values concerning the query entity, the candidate events and the language. Finally, *LaSER* ranks the candidate events in order of their language-specific relevance.

In this section, we describe the background knowledge and training and query phases of *LaSER* in more detail.

3.1. Background knowledge of LaSER

The *LaSER* approach relies on background knowledge, including the language-specific knowledge graph and language-specific click data.

3.1.1. Language-specific knowledge graph

Following Definition 1, the language-specific knowledge graph $G = (E, R, L)$ represents entities, their spatial and temporal characteristics, and relationships in the context of a specific language l.

3.1.2. Language-specific click data

The language-specific click data provides training labels for the ranking model. This data is extracted from the Wikipedia Clickstream and represents real user interactions with Wikipedia articles corresponding to the entities in the language-specific knowledge graph. From such user interactions, we infer the language-specific relevance scores for an entity $e \in E$ and an event $v \in \mathcal{V}$: $rel_{interaction}(e, v, l) \in [0, 1]$. Such values are derived by normalizing click counts in the Wikipedia Clickstream of a language $l \in L$ regarding click counts in all languages L. That way, the $rel_{interaction}$ scores reflect the language-specific relevance. We provide more details regarding this normalization in Section 4.

3.2. Training phase

The goal of the training phase is to create language-specific embeddings and train an event ranking model. The training phase is conducted as a pre-processing and does not impact the query efficiency. This phase consists of the following three steps:

[3] We only show two candidate events for brevity. More candidate events can be generated.

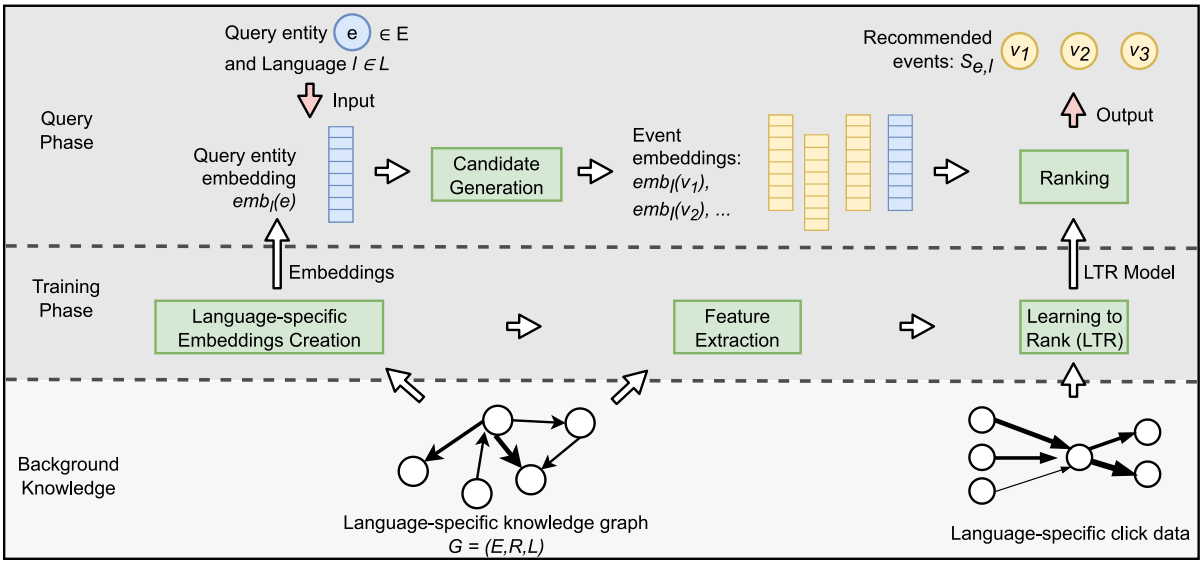


Figure 1. The *LaSER* overview includes three parts. (i) The background knowledge includes the language-specific knowledge graph and the language-specific click data. (ii) In the training pre-processing phase, the language-specific embeddings and the LTR event ranking model are trained based on this background knowledge. (iii) In the query phase, given a query entity e (e.g., *Coronavirus pandemic*) and a language l (e.g., German) as an input, the embeddings and the LTR ranking model are utilized to generate a language-specific ranked list of events $S_{e,l}$ (e.g., ⟨*COVID-19 pandemic in Germany*, *COVID-19 recession*, *COVID-19 pandemic in Austria*⟩).

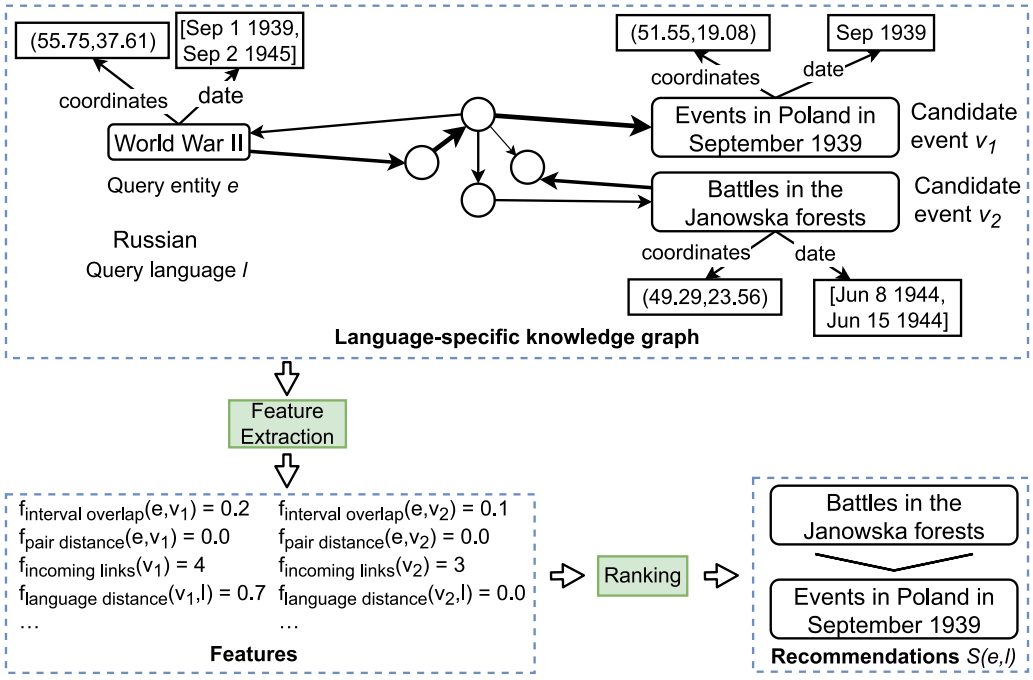


Figure 2. Example query looking for events relevant to *World War II* from the Russian perspective. We illustrate two candidate events with selected features for brevity. *LaSER* retrieves the candidates, extracts feature values, and ranks the candidates.

1. Language-specific embeddings creation: From the language-specific, we create language-specific embeddings of entities and events.
2. Feature extraction: For a pair of an entity and an event, we extract feature values representing different characteristics of the event and the pair. Example features are the event popularity, the spatial distance between the entity and the event and their embedding similarity.
3. Learning to rank: We incorporate the features to train an LTR model which ranks events regarding their relevance to the query entity.

In the following, we describe these steps in more detail.

3.2.1. Language-specific DeepWalk embeddings

To leverage the information of entities and the structure of the language-specific knowledge graph efficiently, we propose a language-specific embedding technique that learns continuous vector representation of entities representing their relations in a language l. This technique maps the entities to low-dimensional vectors, which are similar in cases where two entities appear close to each other in the language-specific context of l.

To create language-specific embeddings, we utilize DeepWalk [10], and follow a uniform random sampling approach: the next entity to visit in the random walk is chosen uniformly from all neighbors of the current entity.

After creating a set of random walks following the DeepWalk approach, we train a Word2Vec model. The resulting language-specific embeddings are utilized for (i) creating a candidate set of events relevant to the query entity and (ii) for measuring the language-specific relevance between the query entity and the event.

3.2.2. Feature extraction

To model event relevance to the query entity e in the context of language l, we extract 10 features from the event v and the entity e. This set of features F includes four groups, covering different entity aspects: spatial, temporal, link-based, and embedding-based features. Some of these features are computed for the event and the language only: $f(v, l) \mapsto \mathbb{R}, v \in \mathcal{V}, l \in L$. Other features are computed for the pair of a query entity and an event in the specific language: $f(v, l, e) \mapsto \mathbb{R}, v \in \mathcal{V}, l \in L, e \in E$ (language-dependent features) or irrespective of the language: $f(v, e) \mapsto \mathbb{R}, v \in \mathcal{V}, e \in E$.

Spatial Features ($Features_{spatial}$): As per challenge (C1), events have a spatio-temporal dimension. Consequently, spatial features are used to capture spatial dependencies between an event and the language, as well as between an event and the query entity.

- Language distance: This continuous feature denotes the spatial distance between the event and the set of countries $\mathfrak{C}_l = \{\mathfrak{c}_1, \mathfrak{c}_2, \ldots, \mathfrak{c}_i\}$ where l is an official language[4]:

$$f_{\text{language distance}}(v, l) = \min_{c \in v.C, \mathfrak{c} \in \mathfrak{C}_l} \text{distance}(c, \mathfrak{c}). \quad (1)$$

  An event can be assigned multiple coordinate pairs ($v.C$), and there can be multiple countries where l is an official language. This feature is based on the closest combination of event coordinate pairs and a country. *distance* represents the distance between a point and a polygon (i.e., a country) in kilometers (or 0, if the point is located inside the polygon). The intuition behind the language distance feature is that the distance can directly impact the interest of the language audience. For example, due to the location, the *2004 Summer Olympics* held in Athens are expected to be more relevant to the Greek language community than the *2008 Summer Olympics* located in Beijing.
- Pair distance: This continuous feature represents the spatial distance between the query entity e and the event v:

$$f_{\text{pair distance}}(v, e) = \min_{c_1 \in v.C, c_2 \in e.C} \text{distance}(c_1, c_2). \quad (2)$$

  This feature considers the minimum distance between any of the coordinate pairs of v and e. In contrast to the language distance in Eq. (1), here *distance* represents the spatial distance between two points. We assume that the lower their distance, the more relevant the event is to the query entity.

Temporal Features ($Features_{temporal}$): To take the temporal closeness of the query entity e and an event v into account, we employ temporal features.

The intuition behind the temporal features is that an event is expected to be more relevant regarding the query entity if they happened simultaneously. The extent of such temporal coincidence is computed through two different features which measure both the temporal overlap and the distance.

- Interval overlap: This feature indicates the overlap between the time intervals of the query entity e and an event v:

$$f_{\text{interval overlap}}(v, e) = \begin{cases} 0, & \text{if } v.t_s > e.t_e \text{ or } e.t_s > v.t_e \\ |[max(v.t_s, e.t_s), min(v.t_e, e.t_e)]|, & \text{else.} \end{cases} \quad (3)$$

- Begin time distance: The start time distance feature represents the time difference between the start times of the query entity e and an event v:

$$f_{\text{begin time distance}}(v, e) = |v.t_s - e.t_s|. \quad (4)$$

Temporal features represent overlap and distance based on the number of days and have discrete values.

Link-based Features ($Features_{links}$): The features in this category represent overall event importance and the similarity of the query entity e and an event v based on their language-specific knowledge graph neighborhoods. We assume that an event is more relevant to an entity if they appear in the same contexts, i.e., have a similar neighborhood in the graph. To measure such similarity of neighborhoods, we consider the number of incoming and outgoing links as well as the shared links between them. We obtain the link counts from the specific Wikipedia language editions. Given a set of links $W_l = E \times E$ in a language-specific Wikipedia edition in a language $l \in L$, there is a link from one entity $e \in E$ to another entity $e_n \in E$, if $(e, e_n) \in W_l$. To compute the overall importance of the event, we use the number of incoming links and outgoing links of v. To measure the similarity of the neighborhoods, we consider the number of shared incoming and outgoing links.

- Number of incoming links: We estimate an overall importance of an event in a language context based on its link count in the Wikipedia link set W_l of the language l:

$$f_{\text{incoming links}}(v, l) = |\{(e_i, v) \in W_l\}|. \quad (5)$$

- Number of outgoing links: In analogy to the number of incoming links, we also consider the number of outgoing links. This feature represents the general interaction of the event with other entities in W_l:

$$f_{\text{outgoing links}}(v, l) = |\{(v, e_i) \in W_l\}|. \quad (6)$$

- Number of shared incoming links: We estimate the similarity between the query entity e and an event v in terms of their interlinking with their neighbors in a specific language. A shared incoming link represents the situation where an entity $x \in E$ refers to both v and e in W_l:

$$f_{\text{shared incoming links}}(v, l, e) = \{x | (x, v) \in W_l \wedge (x, e) \in W_l\}. \quad (7)$$

- Number of shared outgoing links: We also consider shared outgoing links. This number represents the similarity between the query entity e and an event v in terms of their interaction with other entities in W_l. A shared outgoing link represents the situation where v and e refer to an entity $x \in E$ in the context of language l:

$$f_{\text{shared outgoing links}}(v, l, e) = \{x | (v, x) \in W_l \wedge (e, x) \in W_l\}. \quad (8)$$

All link-based features have discrete values.

In addition, we use the Milne–Witten relatedness score that is often used to estimate the semantic relatedness between Wikipedia articles [11].

- Milne–Witten relatedness: This feature value is computed using the Wikipedia link-based measure proposed by Milne

[4] We extract the set of countries where l is an official language from Wikidata using the *official language* property (https://www.wikidata.org/wiki/Property:P37). For example, we extract the countries Germany, Switzerland, Austria, and Liechtenstein for the German language.

and Witten [11].

$$f_{\text{Milne–Witten}}(e, v, l) = 1 - \frac{log(max(|In_e|, |In_v|)) - log(|In_e \cap In_v|)}{log(|E|) - log(min(|In_e, In_v|))}, \quad (9)$$

where $In_e = \{e|(e, e_i) \in W_l, e \in E\}$ and $In_v = \{v|(v, v_i) \in W_l, v \in \mathcal{V}\}$ are the sets of all incoming links to the query entity e and an event v in the Wikipedia link set W_l, respectively. The continuous Milne–Witten relatedness feature is bound between 0 and 1.

Embedding-based Features ($Features_{embeddings}$): The embedding-based features make use of the previously computed language-specific embeddings.

- Embedding similarity: We compute the cosine similarity between the language-specific embeddings of the query entity e and an event v:

$$f_{\text{embedding similarity}}(v, l, e) = cos(emb_l(v), emb_l(e)). \quad (10)$$

This continuous feature is bound between 0 and 1. We assume that v is relevant to e if their embedding vectors are close in the embedding space, as reflected by their cosine similarity.

3.2.3. Learning to rank

To rank the events relevant to the query entity, we train a learning to rank model that takes feature values as an input and is trained to predict the ranking inferred from the language-specific click data. In the context of the LTR model, the problem of language-specific event recommendation is defined as follows: Given a training set of language-specific relevance values between entities and events as well as their features, learn a scoring function that approximates the language-specific relevance $rel_{interaction}(e, v, l)$ for the query entity e and an event v in a language l.

We train a tree ensemble model to learn an optimal ranking of the language-specific relevance scores using LambdaMART [12]. LambdaMART is an LTR algorithm that uses gradient boosted decision trees with a cross-entropy cost function. In the literature, LambdaMART has been shown to outperform neural ranking models in information retrieval tasks [13]. Using LambdaMART, we perform a list-wise ranking where the normalized discounted cumulative gain (nDCG) is maximized.

3.3. Query phase

In the query phase, *LaSER* takes the query entity $e \in E$ and a language $l \in L$ given by the user as input and recommends a language-specific ranking of events $S_{e,l}$ as an output.[5]

The query phase consists of the following two steps:

1. Candidate generation: A set of candidate events is generated based on the language-specific embeddings.
2. Ranking: The candidate events are ranked by the previously trained LTR model.

3.3.1. Candidate generation

Due to numerous events in the language-specific knowledge graph, it is not feasible to compute the relevance scores of all events and to rank them. Therefore, we collect a set of candidate events that are likely to be among the recommended events for the query entity. More specifically, similar to the idea of [5], we select k events which are most similar to the query entity e based on the embedding similarity computed using Eq. (10). Such a candidate set can be obtained efficiently and reflects structural similarities.

3.3.2. Ranking

Finally, for each candidate event v, we compute its feature values as well as the feature values between the query entity e and v based on the language-specific knowledge graph. Given the input set of all candidate events and their feature values, we employ the LTR model trained in the training phase to estimate the language-specific relevance scores. The resulting scores are then used to sort the candidate events according to their relevance and create the set of recommendations $S_{e,l}$.

4. Extraction of background knowledge

LaSER requires a language-specific knowledge graph and language-specific click data described in Section 3.1 as background knowledge. In the following, we describe the extraction of both datasets in more detail.

4.1. Language-specific Knowledge Graph

Entities and their attributes (start and end time and coordinate pairs) in the language-specific knowledge graph are collected from the EventKG knowledge graph [14]. Note that although we are interested in the language-specific information, and thus conceptually speak about a language-specific knowledge graph, from the practical perspective we can also extract such information from multilingual sources, such as EventKG, directly.

4.2. Entities from EventKG

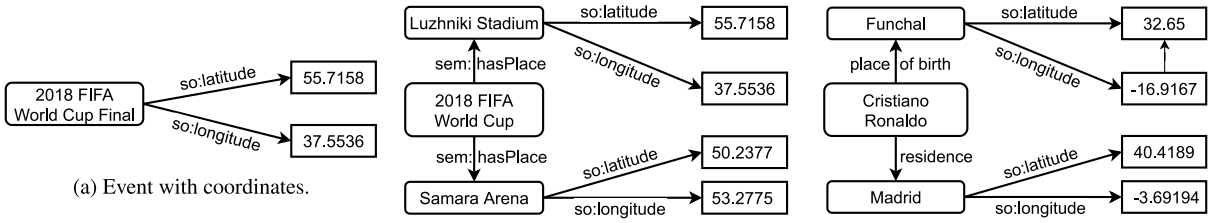
EventKG [14] is a multilingual knowledge graph that contains semantic information regarding events, their relations and temporal information about real-world entities. Such information builds the basis for the language-specific knowledge graph $G = (E, R, L)$ defined in Definition 1. Specifically, E represents the entities of EventKG.

The EventKG entities typed as `sem:Event`[6] make up the event set $\mathcal{V} \subset E$. To retrieve the start and end times of entities in E, we use EventKG's `sem:hasBeginTimeStamp` and `sem:hasEndTimeStamp` relations. The set of coordinate pairs for each entity is collected from a set of relations in EventKG. As shown in Fig. 3, there are different options to retrieve the coordinates from the `so:latitude`[7] and `so:`[longitude] triples. For events and locations, coordinates are often directly assigned to them (Fig. 3(a)). Some events are assigned coordinates via `sem:hasPlace` (Fig. 3(b)). Finally, the entities can be connected to locations via other properties (Fig. 3(c)). For each entity, we select coordinates from one of these options where coordinates are available. Following this process, 40% of entities have temporal and 48% spatial information. In terms of events, the numbers are 82% and 81% for temporal and spatial features, respectively.

[5] We assume that users can select a query entity from the language-specific knowledge graph, e.g. via its label.

[6] `sem`: http://semanticweb.cs.vu.nl/2009/11/sem/, Simple Event Model [15]

[7] `so`: http://schema.org/



(a) Event with coordinates. (b) Event with locations. (c) Entity with relations to locations.

Figure 3. Three examples of coordinate pairs for selected entities. For readability, we use entity and property labels instead of their URIs except for few selected properties.

Table 2
Statistics of the links extracted from three Wikipedia language editions.

	German	French	Russian
Entities (Nodes)	3,028,223	2,259,750	2,242,357
thereof Events	87,573	89,130	59,557
Links (Edges)	51,001,819	41,526,761	29,106,336

Table 3
Statistics of the Wikipedia Clickstream in 2020 in three languages. Click pairs are all recorded clicks between two Wikipedia articles in the respective language and time span.

	German	French	Russian
Entities	1, 222, 070	994,381	831,703
Clicked events	40,223	46,557	33,712
Click pairs	6, 862, 960	5, 192, 491	5, 264, 813

4.3. Link counts from Wikipedia

To compute the link features, we use Wikipedia. Wikipedia is a multilingual encyclopedia available in more than 300 languages.[8] Wikipedia is actively edited by volunteers worldwide, and thus reflects the cultural preferences of audiences in different language communities [16,17]. Each Wikipedia language edition covers a different set of interlinked articles. To profit from these language-specific link structures that potentially represent different linguistic points of view and language-specific information asymmetries, we use Wikipedia as an underlying resource to compute the link features. We extract the Wikipedia links W_l from the German, French and Russian Wikipedia language editions in our experiments.[9]

Table 2 provides statistics of the extracted Wikipedia links.

4.4. Language-specific Click Data

To train the LTR model for event recommendation, *LaSER* requires click data reflecting the language-specific user preferences. We adopt the EventKG+Click dataset created in our previous work [19] to obtain such information. EventKG+Click is a publicly available, cross-lingual dataset that reflects the language-specific relevance of events and their relations. This way, we facilitate the reproducibility of results addressing the challenge (C2).

EventKG+Click was created on top of two data sources: (i) the EventKG knowledge graph, which provides the set of events and (ii) the Wikipedia Clickstream dataset [2] that reflects how users explore articles in different language editions. Precisely, the Wikipedia Clickstream contains counts of how often users clicked on a specific link in a Wikipedia article. For example, the Wikipedia Clickstream contains the following information: in 2020, the links leading to the article *2020 United States presidential election* on the Wikipedia article about *Bernie Sanders* were clicked 2,196 times in the Russian Wikipedia. Another example was shown in Table 1. Table 3 provides statistics extracted from the Wikipedia Clickstream in 2020.

While the Wikipedia Clickstream provides information regarding entities and their relations for the specific Wikipedia language editions, the Clickstream does not put the different language editions in relation to each other. Thus, the clickstream does not fully reflect the language-specific relevance. In this work, we extend the EventKG+Click dataset to use it as a training corpus for the proposed *LaSER* approach as described in the following.

To compute language-specific relevance scores in EventKG+Click, we follow our previous work [19]. The idea behind the language-specific relevance scores is to take all languages available in the Wikipedia Clickstream ($\mathfrak{L}$) into account and normalize the click counts regarding the number of clicks in other languages. For example, while the articles regarding *Coronavirus pandemic* and *2021 German federal election* are often clicked in the German Wikipedia, *German federal election* has a higher language-specific relevance score. This score is higher because the relative number of clicks on *German federal election* is higher in the German Wikipedia than in any other Wikipedia language edition, highlighting the language-specific relevance of the event in the German context.

Formally, given a set of click counts from source to target entities (*clicks*), we first create *balanced click counts* between a source entity $e_s \in E$ and a target entity $e_t \in E$ as follows:

$$balanced\_clicks(e_s, e_t, l) = clicks(e_s, e_t, l) \cdot \frac{\sum_{l' \in \mathfrak{L}} \sum_{e'_s \in E} \sum_{e'_t \in E} clicks(e'_s, e'_t, l')}{\sum_{e'_s \in E} \sum_{e'_t \in E} clicks(e'_s, e'_t, l)}. \quad (11)$$

The language-specific relevance value between an entity $e \in E$ and an event $v \in \mathcal{V}$ introduced in Section 3.2.3 is then computed as follows:

$$rel_{interaction}(e, v, l) = \frac{balanced\_clicks(e, v, l)}{\sum_{l' \in \mathfrak{L}} balanced\_clicks(e, v, l')} \in [0, 1]. \quad (12)$$

Following this procedure, we created an extended version of EventKG+Click covering the whole year of 2020 in the languages German, French and Russian.[10] We make the extended version of EventKG+Click publicly available.[11] The statistics of this dataset are presented in Table 4.

[8] https://en.wikipedia.org/wiki/List_of_Wikipedias

[9] The German, French and Russian Wikipedia are among the top-6 most actively edited Wikipedia editions. English, the most active Wikipedia edition, is often edited by users of other languages and thus has a less clear language-specific focus [18].

[10] In comparison to the first version of EventKG+Click, the new version covers the entire year 2020 and considers all available languages ($\mathfrak{L}$) when computing the relevance scores.

[11] https://github.com/saraabdollahi/EventKG-Click

Table 4
Statistics of the language-specific click data obtained from EventKG+Click.

	German	French	Russian
Source entities	117,281	104,331	97,212
Events	40,223	46,557	33,712
Relevance pairs	304,564	271,243	254,910

5. Evaluation aims and setup

The evaluation aims to assess the performance of the main components of *LaSER*. First, we aim to assess the effectiveness of the proposed language-specific embedding method and its impact on the candidate generation step of *LaSER*. Second, we conduct an evaluation of the recommendations, where we assess *LaSER*'s results using the relevance labels obtained from the EventKG+Click dataset as a ground truth. Third, we analyze the effectiveness of the features adopted by the proposed *LaSER* approach. Fourth, we conduct a user study to assess the recommendation quality from the user perspective. Finally, we manually analyze anecdotal evaluation results and discuss potential application scenarios of the proposed *LaSER* approach.

This section introduces the ground truth for the recommendation evaluation, presents the embedding and recommendation baselines, describes the evaluation metrics, and provides the implementation details.

5.1. Ground truth creation

To train *LaSER* and evaluate the language-specific recommendation, we created a ground truth of language-specific event recommendations from EventKG+Click (see Section 4.4). For each considered language l, this ground truth GT_l contains query entities together with a ranked list of events and is composed as follows:

$$\begin{aligned} GT_l = \{&(e, \langle v_1, \ldots, v_n, v_1^-, \ldots, v_n^- \rangle) \\ &| rel_{interaction}(e, v_i, l) \geq rel_{interaction}(e, v_j, l) \\ &\forall 1 \leq i < j \leq n\}, \end{aligned} \tag{13}$$

where v_i^- denote negative examples, i.e., randomly chosen events not related to the query entity: $rel_{interaction}(e, v_i^-, l) = 0$. In other words, we select all entities in EventKG+Click for which events are provided and rank these events according to their $rel_{interaction}$ score. Each ranked event list is extended with randomly chosen negative examples of the same number as positive events.

We make the ground truth available online.[12]

5.2. Embedding methods

LaSER relies on node embeddings both for candidate generation and as a feature for ranking. We compare the following embedding methods in our evaluation.

5.2.1. DeepWalk embedding

As described in Section 3.2.1, DeepWalk [10] is an embedding approach that learns latent representations of nodes in a network and generalizes neural language models to process sets of randomly generated walks in analogy to sentence-based text embedding models. We compute the DeepWalk embeddings from the language-specific Wikipedia links in each language l.

5.2.2. Node2Vec embedding

Node2Vec [20] is an embedding approach that learns continuous feature representations for nodes in a network that maximizes the likelihood of preserving a network neighborhood of nodes. The authors of Node2Vec designed a biased random walk procedure, which efficiently explores diverse neighborhoods. Unlike DeepWalk [10] that creates embeddings on an unweighted graph using uniform random walks, Node2Vec considers edge weights to conduct a biased random walk. This biased random walk has the flexibility of exploring the network neighborhoods by a trade-off between breadth-first search and depth-first search. We set the Node2Vec parameters $p = 4$ and $q = 0.5$, following the experimental results in [20]. As edge weights, we take the average of shared incoming links and shared outgoing links from the language-specific Wikipedia links.

5.2.3. Wikipedia2Vec embedding

Wikipedia2Vec [21] jointly learns word and entity embeddings by applying the skip-gram model on the Wikipedia link graph, Wikipedia texts and the context terms of Wikipedia links. We use language-specific Wikipedia2vec embeddings pre-trained on the German, French, and Russian Wikipedia.[13]

5.2.4. TransE embedding

TransE [22] models relationships by interpreting them as translations operating on the low-dimensional entity embeddings. We use TransE embeddings[14] pre-trained on the Wikidata5 m dataset, which is not language-specific [23].

5.3. Recommendation baselines

To compare the proposed *LaSER* approach to the state-of-the-art recommendation baselines, we need to ensure that the baselines: (i) represent the state-of-the-art in the entity or event recommendation, (ii) can be applied to the novel task of language-specific event recommendation proposed in this work and (iii) are reproducible, i.e., do not depend on any proprietary data.

Therefore, following the evaluation procedure in [24], we evaluate *LaSER* against four recommendation baselines, which use publicly available data and have different facets: Milne–Witten, which models the relatedness of entities based on links, the embedding-based methods DeepWalk and Node2Vec, as well as SuperGAT, an attention-based graph neural network. We use each of the recommendation baselines to provide a relevance score between an entity $e \in E$ and an event $v \in \mathcal{V}$ in a language l, which is used for the event ranking in the language-specific event recommendation. In the following, we describe these baselines in more detail.

Based on the considerations above, we exclude methods that focus on highly specialized recommendation aspects, such as [6] due to their temporal focus. We also exclude entity recommendation approaches proposed in [5] which depends on the proprietary Yahoo! search logs for feature extraction, and [8] which heavily relies on proprietary click-through data and search logs of the Baidu Web search engine. As these datasets are not publicly available, recommendation methods that heavily depend on these datasets cannot be reproduced and compared to our approach. Note that these methods address the general entity recommendation task, as opposed to the language-specific event recommendation we address. In contrast, we select the baselines which can be applied to the language-specific knowledge graph to address the language-specific relevance.

12 https://zenodo.org/record/5735580

13 https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

14 https://graphvite.io/docs/latest/pretrained_model.html

5.3.1. Milne–Witten recommendation baseline

The Milne–Witten score measures the semantic relatedness between two entities based on the Wikipedia hyperlink structure [11]. For this baseline, we rank events based on their feature value $f_{\text{Milne–Witten}}(e, v, l)$ defined in Section 3.2.2.

5.3.2. DeepWalk recommendation baseline

As the DeepWalk recommendation baseline, we rank events regarding the cosine similarity between the DeepWalk embeddings of the query entity e and an event v.

5.3.3. Node2Vec recommendation baseline

As the Node2Vec recommendation baseline, we rank events regarding the cosine similarity between the Node2Vec embeddings of the query entity e and an event v.

5.3.4. SuperGAT recommendation baseline

We compare *LaSER* to SuperGAT, a state-of-the-art self-supervised graph attention network [25]. The architecture of the SuperGAT recommendation baseline consists of an encoder and a decoder component. In the encoder component, the nodes in the language-specific knowledge graph are embedded using the SuperGAT network. In the decoder component, pairs of negative and positive examples are created from the language-aware click data. Each such pair consists of one positive (query entity, event) example (e, v_1) and a negative (query entity, event) example (e, v_2), such that $rel_{interaction}(e, v_1, l) > rel_{interaction}(e, v_2, l)$ for the given language l. Based on such pairs, the objective is then to minimize the loss function. We adopt the margin ranking loss function typically used in recommender systems [26]. Here, the goal is to rank positive (query entity, event) examples above their corresponding negative (query entity, event) examples. We use three hidden layers with an embedding size of 64 and a learning rate of 0.01.

5.4. Evaluation metrics

For candidate generation, we report candidate recall, i.e., the fraction of events in the ground truth included in the candidate events.

To evaluate the recommendation quality, we use the normalized discounted cumulative gain (nDCG@10) and mean average precision (MAP@10). nDCG@10 compares the top-10 ranked events against the ideal ranking of the ground truth and rewards relevant events in the higher positions, where the ideal ranking achieves an nDCG@10 score of 1.0 [27]. MAP@10 averages over the average precision scores (AP@10) of each query entity in the ground truth, where AP@10 is the sum of precision scores (precision@k for $k = \{1, \ldots, 10\}$) divided by the total number of relevant events in the top-10 ranked results.

For evaluating the user study, we employ mean average precision (MAP@5) regarding the user's relevance judgments. When reporting user study results for selected events, we use average precision (AP@5).

5.5. Implementation details

LaSER and the baselines are implemented in Python 3.7. All experiments are conducted on a Linux machine with Intel(R) Xeon(R) Silver 4210 CPU@ 2.20 GHz and 1 TB memory. We train *LaSER* on EventKG+Click in each language separately. To train the LTR model, we have used the XGBoost library [28] which provides a regularizing gradient boosting framework. The SuperGAT baseline is implemented using PyTorch Geometric [29].

Table 5
Candidate recall achieved by the *LaSER* using different embedding methods.

Model	Candidate recall			
	German	French	Russian	Avg.
Deepwalk	**0.408**	**0.312**	**0.373**	**0.364**
Node2Vec	0.348	0.276	0.371	0.332
TransE	0.009	0.007	0.008	0.008
Wikipedia2Vec	0.017	0.017	0.018	0.017

6. Evaluation results

In this section, we first present the results of the candidate generation and the recommendation evaluation, where we assess the performance of *LaSER* based on the ground truth obtained from EventKG+Click. Then, in a feature analysis, we assess the impact of different feature groups on the *LaSER*'s performance. The queries considered in this section correspond to the source entities in the EventKG+Click dataset presented in Table 4.

6.1. Candidate generation evaluation

As illustrated in Fig. 1, the *LaSER* query phase consists of two main steps: candidate generation and ranking. In this experiment, we evaluate *LaSER*'s performance on the candidate generation task based on the ground truth described in Section 5.1, limited to those cases where a query entity has more than 10 clicked target events.

As described in Section 3.3.1, given a query entity e, the candidate generation step retrieves a set of candidate events regarding their embedding similarity towards e. To demonstrate the effectiveness of *LaSER*'s language-specific embeddings for candidate generation, we compare the performance of different embedding methods. Here, we use the embedding techniques introduced in Section 5.2.

For each query entity in the ground truth and each embedding technique, we retrieve the 200 most similar events as candidate events. Then, we compute the candidate recall per embedding technique, i.e., the fraction of events in the ground truth contained in the candidate events. The results are shown in Table 5. The DeepWalk and Node2Vec embeddings clearly outperform the other two embeddings, with the non-language-specific TransE embedding performing worst. This result demonstrates the benefit of creating language-specific random-walk-based embeddings for the language-specific event recommendation.

6.2. Recommendation evaluation

In this experiment, we evaluate *LaSER*'s performance on the recommendation task based on the ground truth described in Section 5.1. Given a query entity and a set of candidate events, the goal in this task is to rank the candidate events according to their relevance to the query entity for the audience speaking the language of interest, i.e., the language-specific relevance.

In this experiment, *LaSER* is trained via a 5–fold cross-validation on each language separately. The folds are created based on the set of query entities: in each run, we use 80% of the query entities and their events in the ground truth for training the LTR model, the remainder for testing. The results are averaged over the 5 runs.

Table 6 reports the nDCG@10 and MAP@10 scores of the recommendation evaluation for the four recommendation baselines and *LaSER* in three languages. As we can observe, in all the three languages, *LaSER* clearly outperforms the baselines. On average, across languages, with an nDCG@10 of 0.957 and MAP@10 of 0.97, *LaSER* outperforms the baselines by more than 8 and 17 percentage points, respectively. The *LaSER* performance is similar across languages.

Table 6
NDCG@10 and MAP@10 scores achieved by the *LaSER* approach and the recommendation baselines in three languages in the ranking study.

	nDCG@10 Score				MAP@10 Score			
	German	French	Russian	Avg.	German	French	Russian	Avg.
Milne–Witten	0.893	0.897	0.890	0.893	0.848	0.864	0.838	0.850
Node2Vec	0.860	0.841	0.885	0.862	0.729	0.679	0.803	0.737
DeepWalk	0.899	0.858	0.901	0.886	0.731	0.850	0.848	0.810
SuperGAT	0.853	0.884	0.879	0.872	0.824	0.806	0.780	0.803
LaSER	**0.957**	**0.958**	**0.956**	**0.957**	**0.969**	**0.970**	**0.971**	**0.970**

Table 7
Feature analysis: The results of *LaSER* by leaving out feature groups. We report the nDCG@10 scores in three languages.

Model	German	French	Russian
LaSER	0.957	0.958	0.956
- $Features_{spatial}$	0.956	0.952	0.955
- $Features_{temporal}$	0.956	0.957	0.956
- $Features_{links}$	0.911	0.946	0.909
- $Features_{embeddings}$	0.950	0.957	0.951

6.3. Feature analysis

To assess the effectiveness of specific feature groups in *LaSER*, we perform a feature analysis. To this extent, we leave out one feature group at a time and measure the resulting performance regarding nDCG@10. The results are presented in Table 7. As we can observe, each feature group contributes towards the *LaSER* overall performance. The link-based features ($Features_{links}$) provide the highest contribution among the four feature groups, while the temporal features ($Features_{temporal}$) have the lowest impact. We observe similar effects of feature groups across all languages.

A relatively low contribution of the spatial and temporal features can be explained through the non-availability of these features for a large proportion of entities, as reported in Section 4.2. Furthermore, whereas language-specific embeddings provide a substantial contribution in the candidate generation step, as discussed in Section 6.1, they have only a limited impact on the follow-up ranking step. An average embedding-based similarity in this step is 0.65 with a relatively low standard deviation of $\sigma = 0.18$. Thus, re-ranking candidates based on the embedding-based similarity is only possible to a limited extent. Overall, incorporating all the proposed feature groups leads to the best performance of the proposed approach.

Fig. 4 presents the correlation analysis of the features proposed in this article, computed using Pearson Correlation Coefficient (PCC). We report absolute correlation values. As we can observe, the highest correlation is obtained between the DeepWalk similarity and the number of links, as well as between the specific features in the time category. These correlations are expected. Overall, we observe that the general feature categories provide complementary information with low correlation scores across categories.

7. User study

The aim of the user study is to assess the recommendation quality from the user perspective. In this section, we describe the user study setup and discuss the results. Furthermore, we report the user agreement and provide insights into the user feedback.

7.1. User study setup

Existing datasets such as the Wikipedia Clickstream usually only cover a fraction of events potentially relevant to a query entity. Consequently, evaluation is typically performed via pooling [5], i.e., by judging the relevance of the top recommendations generated by the methods under consideration [6]. In the user study, we followed the pooling approach to obtain the relevance judgments.

To conduct the user study, we selected 10 popular query entities of various types, namely events, places, persons, art and religion. For each query entity, we generated event recommendations for German, French and Russian languages using the following methods:

- The link-based Milne–Witten recommendation baseline, the best performing baseline according to Table 6.
- The embedding-based DeepWalk recommendation baseline, the second best performing baseline according to Table 6.
- The proposed *LaSER* approach.

From each ranking, we select the top-5 highest ranked events and generate a set of 415 (query entity, event, language) triples.[15] To alleviate possible ranking-based bias in the judgments, we randomized the order of recommendations obtained by different methods for each query entity when presenting the recommendations to the study participants. Each triple was annotated by at least two study participants.

To gain more detailed insights into how the study participants perceive event relevance in a specific context, we break down the judgment into three *relevance criteria*: (i) relevance to the topic (i.e., query entity), (ii) relevance to the audience of a language community and (iii) relevance to the general audience.

Following the TREC annotation guidelines [30], we asked the study participants to assume they want to write a report about the given topic and provide their relevance judgments regarding that setting. More specifically, we provided the participants with the following instructions:

- Assume you want to write a report on the given topic (i.e. query entity).[16]
  - Relevance to the topic: To what extent do you find the recommended event relevant to the topic and worth mentioning in your report?
- Assume you want to write a report on the recommended event (independent of the given topic).
  - Relevance to the audience of a language community: To what extent do you find the event relevant to the audience that speaks the language?
  - Relevance to the general audience: To what extent do you find the event relevant to the general audience?

[15] As different approaches can recommend the same event, the total number of triples is less than 10 (query entities) ·5 (events per recommendation) ·3 (languages) ·3 (methods) = 450.

[16] For easier comprehension for the user study participants, we use the term "topic" in analogy to "query entity" in the user study interface.

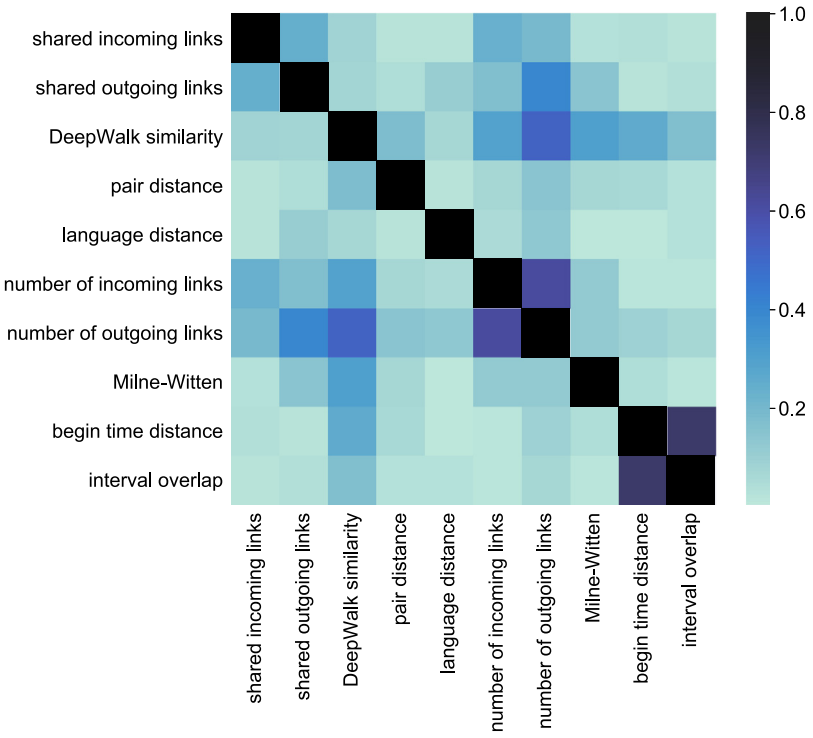


Figure 4. Feature correlation analysis using Pearson Correlation Coefficient (PCC). We report absolute correlation values.

For each recommended event and relevance criterion, the participants are asked to indicate whether the event is strongly relevant (3), partially relevant (2) or irrelevant (1). Alternatively, the participants can select the "I don't know" option. To determine whether an event is relevant to a query entity in a specific language, we average over the user judgment scores. Events that exceed an average of 1.5 are considered relevant when measuring precision of recommendations. Events judged as "I don't know" by any of the study participants are excluded from the evaluation.

A screenshot of the user study interface is presented in Fig. 5. The interface also provides links to the Wikipedia articles of the query entity and the recommended events, such that the participants can obtain additional information if required.

To collect feedback from the user study participants, they were given an option to leave a comment for each of their judgments. In addition, we conducted post-study interviews with selected participants.

As stated in Definition 4, we are interested in the relevance of an event to the query entity for the audience speaking the language of interest. Therefore, we derive language-specific relevance scores by requiring that both the relevance to the topic and the relevance to the language audience criteria are fulfilled.

- Language-specific relevance: An event is language-specifically relevant when it is relevant to the topic (i.e., query entity) and relevant to the audience of the language community.

17 post-graduate researchers in Computer Science and Digital Humanities participated in the study. In total, participants annotated 935 triples. Each participant annotated at least 9 triples and 55 triples on average. The ratings are available online.[17]

[17] https://zenodo.org/record/5735580

7.2. User study results

Table 8 reports the MAP@5 scores for all languages and relevance criteria. As we can observe, *LaSER* outperforms the recommendation baselines in the majority of the relevance criteria across the three languages. Most importantly, *LaSER* obtains the highest MAP@5 on relevance to the language audience in all languages (French: 0.95, German: 0.91 and Russian: 0.84) and outperforms the baselines by 10 (German), 17 (French) and 25 percentage points (Russian). For German and French, *LaSER* also achieves the highest relevance towards the general audience. Regarding the topic relevance, all approaches achieve MAP@5 scores larger than 0.9 and *LaSER* is slightly outperformed by the two baselines. This result is expected, as *LaSER* aims at language-specific rather than generic topical relevance.

To estimate *LaSER*'s ability to provide recommendations which cover relevance to the topic and the language community, we are particularly interested in the language-specific relevance. Here, *LaSER* outperforms both baselines in all languages. While the language-specific relevance for German is 0.81 which is similar to Milne–Witten, the language-specific relevance for French and Russian is 0.90 and 0.84 and clearly exceeds those of the baselines by up to 33 percentage points.

Table 9 presents *LaSER*'s performance for each of the query entities annotated in the user study separately. The highest AP@5 scores considering the relevance to the language community are achieved for query entities of type *Event* including sports and cultural events. Even in cases where the query entity has general engagement and impact, such as *Olympic Games* and *World War II*, our proposed approach can recommend specific events to each language community.

7.3. User agreement

To estimate the difficulty of providing relevance judgments for language-specific event recommendations, we measure the agreement between the user study participants using Fleiss' kappa

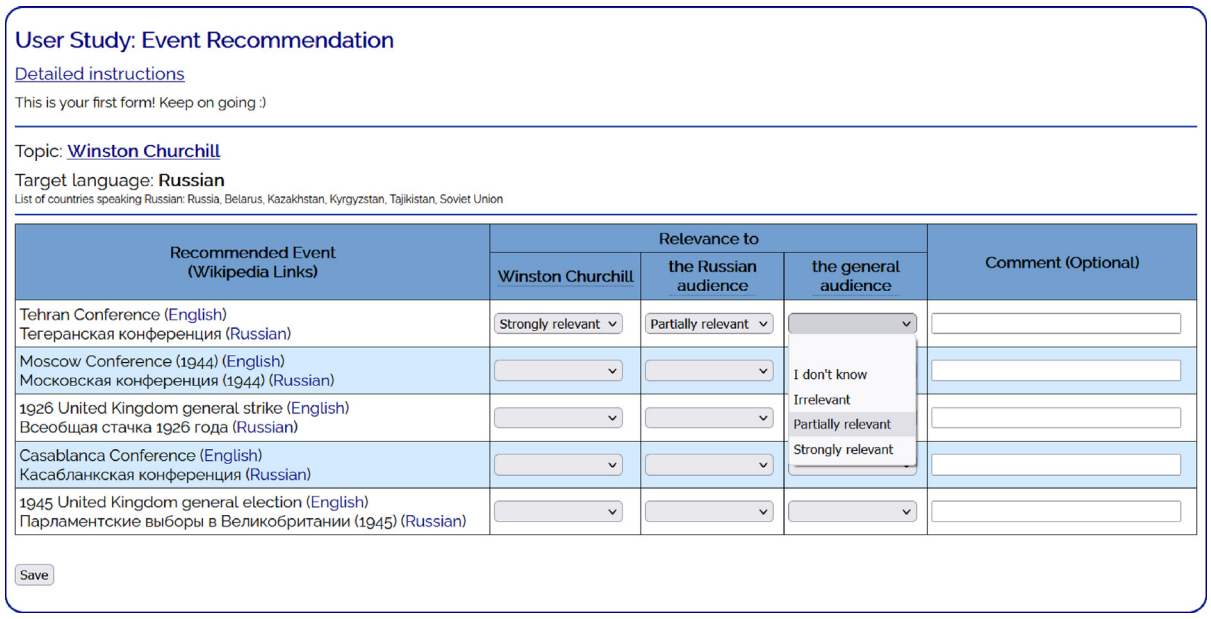


Figure 5. Screenshot of the user study. Given a topic, i.e., a query entity (here: *Winston Churchill*) and a target language (here: Russian), the participants are asked to judge the relevance of the recommended events. The participants assess each recommended event according to three relevance criteria: relevance to the topic, relevance to the audience of a specific language community, and relevance to the general audience. The relevance scores are as follows: strongly relevant, partially relevant, irrelevant, and unknown. An interface simultaneously presents five events (e.g., *Tehran Conference*, *Moscow Conference (1944)*, and others).

Table 8
User study results: MAP@5 of *LaSER* and two recommendation baselines in three languages regarding three relevance criteria judged in the user study and the overall language-specific relevance.

	Relevance (MAP@5)			Language-Specific relevance
	Topic	General audience	Language audience	
		German		
Milne–Witten	**1.00**	0.87	0.81	**0.81**
DeepWalk	0.98	0.77	0.70	0.70
LaSER	0.93	**0.89**	**0.91**	**0.81**
		French		
Milne–Witten	**1.00**	0.88	0.68	0.68
DeepWalk	**1.00**	0.88	0.78	0.77
LaSER	0.95	**0.94**	**0.95**	**0.90**
		Russian		
Milne–Witten	**1.00**	0.90	0.59	0.59
DeepWalk	**1.00**	**0.95**	0.51	0.51
LaSER	0.98	0.90	**0.84**	**0.84**

Table 9
User study: Detailed analysis of *LaSER* for all query entities annotated in the user study. For each query entity and a language, we report the average precision (AP@5) for the three relevance criteria (Topic: Relevance to the Query Entity, Lang.: Relevance to the audience of a language community, General: Relevance to the general audience).

Query entity	Type	AP@5 (German)			AP@5 (French)			AP@5 (Russian)		
		Topic	Lang.	General	Topic	Lang.	General	Topic	Lang.	General
Christianity	Religion	1	1	1	1	1	1	0.8	1	0.91
Film festival	Cultural event	1	1	1	1	1	1	1	1	1
Germany	Place	1	1	1	1	1	1	0.8	1	0.54
Olympic games	Sport event	1	1	1	1	1	1	1	1	1
Painting	Art	0.8	0.54	0.91	1	1	1	0.2	0.8	0.8
Social movement	Group action	1	0.91	1	1	1	0.84	0.8	1	0.84
UK	Place	0.6	0.55	0.61	1	0.50	0.54	0.8	1	0.96
Winston Churchill	Person	0.29	0.96	0.84	1	1	1	0.8	0.8	0.96
World War I	Event	0.87	0.96	0.25	0.45	1	1	1	1	1
World War II	Event	1	1	1	1	1	1	1	1	1

Table 10
Inter-rater agreement assessment using Fleiss' kappa.

Agreement on relevance to the	German	French	Russian	Avg.
Topic	0.62	0.79	0.89	0.78
Language audience	0.39	0.48	0.44	0.44
General audience	0.01	0.02	−0.05	− 0.01
General audience (binary)	0.15	0.10	0.23	0.16

statistic. This statistic measures agreement among any constant number of raters [31], where the values less than 0 indicate a poor agreement and 1 a perfect agreement.

To assess the user agreement, we conducted a second phase of the user study following the same setup, but under the following conditions:

- We considered *Film festival* as the query entity.
- We collected judgments from 5 users for each (query entity, event, language) triple.
- For each language, we added 5 negative examples randomly selected from the set of all events.

In total, we collected 270 annotations for 54 (query entity, event, language) triples in the user agreement study.

We compute the agreement among the study participants for evaluating the events in three languages, three different relevance criteria and three classes (partially relevant, strongly relevant and irrelevant). The resulting Fleiss' kappa (κ) values are shown in Table 10. According to these results, users achieved a substantial agreement when judging whether a recommended event is relevant to the query entity ($\kappa \geq 0.62$). We observe a moderate agreement for relevance to the language audience (average $\kappa = 0.44$). Interestingly, the users disagreed the most when judging the relevance of an event to the general audience (average $\kappa = -0.01$). To check if users at least agreed on the irrelevance of events, we also computed Fleiss' kappa considering only two classes ("partially or strongly relevant" and "irrelevant") for the relevance to the general audience, which slightly improved the measured agreement (average $\kappa = 0.16$).

The user agreement results confirm that the consideration of language-specificity is important, as this dimension can be captured more easily than the general relevance by the users.

There were cases where the same event was annotated with all grades of relevance (strongly, partially and irrelevant), specifically when judging relevance to the general audience. Examples include *46th Venice International Film Festival* and *All-Union Film Festival*. While some users see specific film festivals as generally relevant, others disagree. These examples demonstrate the subjectivity when providing relevance judgments. For example, a film fan might be more convinced to rate specific film festivals as generally relevant than others.

7.4. Participant feedback

To gain further insights into the relevance judgments provided by the study participants, we now look at their feedback, which was collected from the comments they could provide during the study (see Fig. 5) and in post-study interviews with selected participants. From this feedback, we identify the following challenges the study participants were facing:

- Lack of information: In some cases, study participants were unable to retrieve enough information about a recommended event to make a confident judgment. Examples include the triple (*World War I*, *Colmar Pocket*, French) (user comment: "I have heard nothing about that") and (*Winston Churchill*, *Litvinov Protocol*, Russian) (user comment: "I tried to find this word in the English Wikipedia, but I did not find it there".). This example also illustrates the information asymmetry across the Wikipedia language editions.
- Difficulty in judging the relevance to the general audience: In a post-study interview, one study participant described that it was difficult to decide on the relevance to the general audience. Regarding relevance to the language audience, the user identified more intuitive criteria, such as the location of the event or its participants. In contrast, the user struggled with finding particular criteria to measure the relevance to the general audience.
  This observation confirms our interpretation of the user agreement analysis, where we observed only a slight agreement regarding the relevance for the general audience.
- Wrong classification: In a few cases, the recommendation was not an event. An example is the triple (*Christianity*, *Constantinople*, French) (user comment: "Not an event"). This error type can be explained by wrong event type assignments in the underlying knowledge graph. In our evaluation, those cases are excluded if one of the users selected the "I don't know" option.

From the participant feedback, we learn that the distinction into three relevance scores helps guide the user study participants through their annotations, specifically in the cases where users do not feel confident with their judgment due to missing information.

8. Anecdotal results and application scenarios

In our final evaluation step, we analyze selected event recommendations of *LaSER* for two query entities annotated during the user study. In this section, we present anecdotal results and application scenarios to highlight the strengths of the proposed *LaSER* approach.

8.1. Anecdotal example 1: Film festival

As our first example query entity, we select *Film Festival* as a rather generic topic. The top-5 events recommended by *LaSER* for German, French and Russian are shown in Table 11. The recommended events clearly show a language-specific focus, i.e., important film festivals that happened in cities where the respective languages are spoken: for example, *LaSER* recommends *International Short Film Festival Oberhausen* and *Filmfest München* for the German audience, several César Awards and *Brest European Short Film Festival* for the French audience and *Moscow International Film Festival* as well as the *Окно в Европу*, a Russian film festival happening in the Russian city *Wyborg*, for the Russian audience.

8.2. Anecdotal example 2: World war II

As another query entity, we selected *World War II* as a concrete event for which we expect to retrieve a lot of regionally significant sub-events. The generated recommendations are shown in Table 12. As expected, these recommendations contain operations, battles, and conferences that happened in the immediate context of *World War II*. Again, the recommended events show a language-specific focus: For German, we get *Aktion Silberstreif*, *Battle of Loos* in the French city *Loos* for French and *Бои в Яновских лесах* (Battles in the Janowska forests) for Russian. Other recommended events such as *Vietnamese famine of 1945*, *Spa Conference of 1920* and *Death of Adolf Hitler* show a less language-specific focus. However, they represent important happenings during *World War II*.

In general, the two presented anecdotal examples further illustrate that *LaSER* can recommend events that differ across languages, reflecting language-specific relevance.

Table 11
Anecdotal example 1: Events recommended for the query entity *Film Festival* in three languages.

		Film festival
German	1	46th Venice International Film Festival
	2	International Short Film Festival Oberhausen
	3	KALIBER35 Munich International Short Film Festival
	4	Filmfest Hamburg
	5	Filmfest München
French	1	34th César Awards
	2	6th César Awards
	3	21st Lumières Awards
	4	17th César Awards
	5	Brest European Short Film Festival
Russian	1	All-Union Film Festival
	2	Окно в Европу (кинофестиваль) *Window to Europe (film festival)*
	3	Moscow International Film Festival
	4	Short Film
	5	Kinotavr

Table 12
Anecdotal example 2: Events recommended for the query entity *World War II* in three languages.

		World War II
German	1	Aktion Silberstreif
	2	Operation Jedburgh
	3	Vietnamese famine of 1945
	4	Einsatzgruppen trial
	5	Operation Felix
French	1	Spa Conference of 1920
	2	French war planning 1920–1940
	3	Locarno Treaties
	4	Battle of Loos
	5	Treaty of Neuilly-sur-Seine
Russian	1	Бои в Яновских лесах *Battles in the Janowska forests*
	2	События в Польше в сентябре 1939 года *Events in Poland in September 1939*
	3	Гибель авианосца «Глориес» *Sinking of the aircraft carrier Glorious*
	4	Defence of the Polish Post Office in Danzig
	5	Death of Adolf Hitler

8.3. Application scenarios

Motivated by the anecdotal examples, we envision the following application scenarios based on language-specific event recommendations:

- **Within-language exploration**: Exploration of events related to a topic in a specific language can strengthen the exploration focus. For example, a historian researching the course of *World War II* in France might be specifically interested in relevant happenings related to France, including the French war planning and *Battle of Loos* shown in Table 12. With *LaSER*, such historian could easily collect such events and use them as a basis for further research.
- **Cross-language exploration**: Language-specific event recommendation could also help to explore topics from a variety of viewpoints and thus widen horizons and minimize potential cultural biases which are prevalent on the Web [32]. A user can specify multiple languages and explore the respective events related to a topic. In our *Film Festival* example in Table 11, this procedure would result in a collection of important film festivals in different parts of the world. That way, one can even explore events such as *Окно в Европу* (Window to Europe), which are only described in a specific language but might be of interest for the cross-language exploration.
- **User language background**: The language background is often part of a user's profile. Thus, recommending events specifically relevant to the user language can potentially satisfy the user-specific information needs and increase user satisfaction in web navigation and exploratory search.

9. Related work

The *LaSER* approach presented in this article aims at recommending events by taking their language-specific relevance into account. This section describes related research areas including entity and event recommendation, learning to rank, graph embeddings, and cross-lingual research.

9.1. Recommendation

While the tasks of user–item and entity recommendation have been extensively studied in the literature, event recommendation has until now been mainly limited to social media events.

9.1.1. User–item recommendation

A typical recommendation task is that of user–item recommendation, where items (e.g., movies or points of interest) are recommended to an individual user [33], typically based on a network of users, items, and their interactions. The recommendation can be based on the user's past preferences, taking into account the preferences of similar users (collaborative filtering) or the similarity to other items (content-based filtering). Knowledge graphs have been used to serve as background knowledge for item–user recommendations where they provide additional information regarding the connections between items [34–37].

The task of language-specific event recommendation introduced in this article has different prerequisites than the user–item recommendation and is thus not comparable. Event recommendations are provided given a query entity, a query language, and a language-specific knowledge graph. However, there is no user–item network and consequently no possibility to incorporate the preferences of individual users.

9.1.2. Entity recommendation

Entity recommendation is the task of recommending a ranked list of entities to the user query. Blanco et al. [1] presented Spark, an entity recommendation system that, by using and combining several signals from a variety of data sources, provides a ranking of the entities related to the user query. Ni et al. [5] proposed a framework for recommending related Wikipedia entities using an architecture of multiple layered graphs, candidate generation via Doc2Vec embeddings, and ranking with an LTR model. In contrast to *LaSER*, this model is trained on proprietary search log data and is therefore difficult to reproduce. Other approaches focus on specific recommendation aspects: Zhang et al. [6] proposed a time-aware entity recommendation (TER), which allows users to restrict their interests in entities to a customized time range. Tran et al. [24] extended TER by incorporating topic and time and proposed contextual relatedness among entities using embedding techniques. The user interest and preference have also been studied by Bi et al. [38] who proposed "probabilistic Three-way Entity Model" (TEM) that provides personalized recommendations of related entities using user interactions from personal click logs. Huang et al. [8] studied serendipity to engage the interest of users while recommending entities.

Existing entity recommendation methods are focused on recommendation regardless of language preferences. Unlike these methods, our proposed *LaSER* approach takes languages into account and recommends relevant language-specific events.

9.1.3. Event recommendation

Events take an important role in a range of real-world applications, including news search [39], news linking [40] and event-centric user interfaces [41]. However, event recommendation has not been extensively studied and is primarily focused on social media events. Existing event recommendation approaches [42] focus on event-based social networks (ESRN) such as Meetup, where the goal is to recommend social events such as parties, concerts, and conferences to the users. Unlike the approaches mentioned above, we focus on events of societal importance, such as the Coronavirus pandemic and the Second World War. With the proposed *LaSER* approach, we leverage structured information from knowledge graphs and consider information needs and the specific context of language communities.

9.2. Learning to rank

The ranking is an essential step of many recommendation algorithms, typically following the candidate generation step. Given a set of objects, a ranking model calculates the score of each object and sorts them accordingly. The scores may represent the degrees of relevance, preference, or importance, depending on applications [43].

LambdaMART is a state-of-the-art LTR model that uses a boosted tree model, which is, according to [13], still "hard to beat for most neural ranking models based on raw texts". LambdaMART demonstrated superior performance when click-based data were used as features [44] and has been applied in many application domains, including recommendations [45], e-commerce click and search [46]. Our evaluation of *LaSER* confirms LambdaMART's superiority for the task of language-specific event recommendation.

9.3. Embedding methods

Graph embedding techniques have been recently adopted for recommendation tasks [47]. Graph embeddings aim to represent graph nodes by low-dimensional vectors, which preserve the graph structure. They are created using random-walk-based, deep-learning-based, and factorization-based methods. Random-walk-based methods [10,20] capture the node neighborhoods by creating random walks over the graph nodes fed into language models in analogy to sentences. Such methods are beneficial for large graphs when the graph is too large to cover in its entirety [48]. Well-known random-walk-based methods include DeepWalk [10] and Node2Vec [20] which learn continuous feature representations for nodes based on biased random walks that provide a trade-off between breadth-first and depth-first graph search. Graph embedding methods based on deep learning [49] typically learn auto-encoders to compress information about the local node neighborhood [50]. Factorization-based algorithms [51] represent the connections between nodes in the matrix form and factorize this matrix to obtain the embedding [48]. Due to the large size of the language-specific knowledge graph we employ random-walk-based graph embedding methods, which provide an effective solution for large graphs.

Knowledge graph embeddings specifically target the embedding of entities and relations in a knowledge graph and are used for knowledge graph completion, relation extraction, and other tasks. Translational distance models such as TransE [22] and its extensions exploit distance-based scoring functions. They measure the plausibility of a fact as the distance between the two entities, usually after a translation carried out by the relation [52]. Other knowledge graph embedding methods employ additional information such as entity types [53], relation paths [54], textual information [53] and hybrid information (e.g., Wikipedia2Vec [21]) in the embedding process. In Section 6.1, we discuss the impact of different knowledge graph embedding methods on *LaSER* and the benefits of using language-specific embeddings for the candidate generation.

9.4. Cross-lingual research

The emerging need to analyze multilingual information on the web has been targeted in a variety of studies, e.g., [55]. Wikipedia is an essential source for multilingual studies regarding the content, number of users, and language coverage. In this regard, several studies focused on investigating and exploring cross-lingual differences in Wikipedia [56]. By analyzing bias and linguistic points of view regarding a controversial event, R. Rogers [17] illustrates that Wikipedia articles varied in their titles and the content across the different language editions. Other works studied multilingualism in terms of user editing behavior [18] and reflection of cross-cultural similarities in the process of collective archiving knowledge on Wikipedia [57].

The *LaSER* approach presented in this article provides an intuitive way to explore language-specific events that can be beneficial for language-specific and cross-lingual studies.

10. Conclusion

In this article, we defined the novel task of language-specific event recommendation. We presented *LaSER*, a novel approach to tackle this task. *LaSER* recommends a list of events relevant to the query entity in a language-specific context. After the creation of language-specific entity embeddings, we train a learning to rank model that generalizes from language-specific click data using spatial, temporal, link-based and latent embedding-based features. We experimentally demonstrate the benefit of creating language-specific embeddings for the task of language-specific event recommendation. Furthermore, our experiments on a real-world dataset demonstrate the effectiveness of *LaSER* in ranking events compared to link-based, embedding-based and graph attention network-based recommendation baselines, outperforming them by more than 8 (nDCG@10) and 17 (MAP@10) percentage points, respectively. Moreover, we identified and analyzed different relevance criteria in a user study and demonstrated that *LaSER* effectively recommends events in a language-specific context. Regarding the language-specific relevance, *LaSER* outperforms the best performing baselines by up to 33 percentage points in MAP@5. Our evaluation demonstrates that language-specific context is an essential event recommendation criterion, together with topical and global event relevance.

CRediT authorship contribution statement

Sara Abdollahi: Conceptualization, Methodology, Software, Writing – original draft, Investigation. **Simon Gottschalk:** Validation, Formal analysis, Writing – review & editing, Supervision. **Elena Demidova:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 812997 ("Cleopatra") and by DFG, German Research Foundation under grant agreement no. 424985896 ("WorldKG").

References

[1] R. Blanco, B.B. Cambazoglu, P. Mika, N. Torzec, Entity recommendations in web search, in: International Semantic Web Conference, Springer, 2013, pp. 33–48, http://dx.doi.org/10.1007/978-3-642-41338-4_3.

[2] Wikimedia Analytics, Research:Wikipedia clickstream, 2021, https://meta.wikimedia.org/wiki/Research:Wikipedia_clickstream/. [last accessed on November 30th, 2021].

[3] S. Gottschalk, V. Bernacchi, R. Rogers, E. Demidova, Towards better understanding researcher strategies in cross-lingual event analytics, in: International Conference on Theory and Practice of Digital Libraries, Springer, 2018, pp. 139–151, http://dx.doi.org/10.1007/978-3-030-00066-0_12.

[4] J.H. Liu, R. Goldstein-Hawes, D. Hilton, L.-L. Huang, C. Gastardo-Conaco, E. Dresler-Hawke, F. Pittolo, Y.-Y. Hong, C. Ward, S. Abraham, et al., Social representations of events and people in world history across 12 cultures, J. Cross-Cultural Psychol. 36 (2) (2005) 171–191.

[5] C.-C. Ni, K. Sum Liu, N. Torzec, Layered graph embedding for entity recommendation using wikipedia in the yahoo! knowledge graph, in: Companion Proceedings of the Web Conference 2020, 2020, pp. 811–818, http://dx.doi.org/10.1145/3366424.3383570.

[6] L. Zhang, A. Rettinger, J. Zhang, A probabilistic model for time-aware entity recommendation, in: International Semantic Web Conference, Springer, 2016, pp. 598–614, http://dx.doi.org/10.1007/978-3-319-46523-4_36.

[7] H. Khrouf, R. Troncy, Hybrid event recommendation using linked data and user diversity, in: Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp. 185–192, http://dx.doi.org/10.1145/2507157.2507171.

[8] J. Huang, S. Ding, H. Wang, T. Liu, Learning to recommend related entities with serendipity for web search users, ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP) 17 (3) (2018) 1–22, http://dx.doi.org/10.1145/3185663.

[9] J. Allan, R. Papka, V. Lavrenko, On-line new event detection and tracking, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 37–45, http://dx.doi.org/10.1145/3209978.3210136.

[10] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710, http://dx.doi.org/10.1145/2623330.2623732.

[11] I.H. Witten, D.N. Milne, An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links, AAAI Press, 2008.

[12] C.J. Burges, From RankNet to LambdaRank to lambdamart: An overview, Learning 11 (23–581) (2010) 81.

[13] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W.B. Croft, X. Cheng, A deep look into neural ranking models for information retrieval, Inf. Process. Manage. 57 (6) (2020) 102067.

[14] S. Gottschalk, E. Demidova, EventKG–the hub of event knowledge on the web–and biographical timeline generation, Semantic Web 10 (6) (2019) 1039–1070, http://dx.doi.org/10.3233/SW-190355.

[15] W.R. Van Hage, V. Malaisé, R. Segers, L. Hollink, G. Schreiber, Design and use of the simple event model (SEM), J. Web Semant. 9 (2) (2011) 128–136, http://dx.doi.org/10.1016/j.websem.2011.03.003.

[16] B. Hecht, D. Gergle, The tower of Babel meets web 2.0: User-generated content and its applications in a multilingual context, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010, pp. 291–300, http://dx.doi.org/10.1145/1753326.1753370.

[17] R. Rogers, Digital Methods, MIT Press, 2013, http://dx.doi.org/10.1002/asi.23594.

[18] S.A. Hale, Multilinguals and wikipedia editing, in: Proceedings of the 2014 ACM Conference on Web Science, 2014, pp. 99–108, http://dx.doi.org/10.1145/2615569.2615684.

[19] S. Abdollahi, S. Gottschalk, E. Demidova, EventKG+Click: A dataset of language-specific event-centric user interaction traces, in: Proceedings of the 1st International Workshop on Cross-Lingual Event-Centric Open Analytics Co-Located with the 17th Extended Semantic Web Conference (ESWC 2020), in: CEUR Workshop Proceedings, vol. 2611, CEUR-WS.org, 2020, pp. 32–42.

[20] A. Grover, J. Leskovec, Node2Vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864, http://dx.doi.org/10.1145/2939672.2939754.

[21] I. Yamada, A. Asai, J. Sakuma, H. Shindo, H. Takeda, Y. Takefuji, Y. Matsumoto, Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 23–30.

[22] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Adv. Neural Inf. Process. Syst. 26 (2013).

[23] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, Trans. Assoc. Comput. Linguist. 9 (2021) 176–194.

[24] N.K. Tran, T. Tran, C. Niederée, Beyond time: Dynamic context-aware entity recommendation, in: European Semantic Web Conference, Springer, 2017, pp. 353–368, http://dx.doi.org/10.1007/978-3-319-58068-5_22.

[25] D. Kim, A.H. Oh, How to Find Your Friendly Neighborhood: Graph Attention Design with Self-Supervision, International Conference on Learning Representations (ICLR 2021), 2021.

[26] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, M. Volkovs, Hgcf: Hyperbolic graph convolution networks for collaborative filtering, in: Proceedings of the Web Conference 2021, 2021, pp. 593–601.

[27] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. (TOIS) 20 (4) (2002) 422–446, http://dx.doi.org/10.1145/582415.582418.

[28] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[29] M. Fey, J.E. Lenssen, Fast graph representation learning with PyTorch Geometric, in: ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[30] E.M. Voorhees, D. Harman, Overview of TREC 2002, in: TREC, 2002.

[31] J.L. Fleiss, Measuring nominal scale agreement among many raters, Psychol. Bull. 76 (5) (1971) 378.

[32] N. Nangia, C. Vania, R. Bhalerao, S. Bowman, CrowS-Pairs: A challenge dataset for measuring social biases in masked language models, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 1953–1967, http://dx.doi.org/10.18653/v1/2020.emnlp-main.154.

[33] F. Yu, A. Zeng, S. Gillard, M. Medo, Network-based recommendation algorithms: A review, Physica A 452 (2016) 192–208.

[34] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. (2020).

[35] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 950–958.

[36] P. Wang, Y. Fan, L. Xia, W.X. Zhao, S. Niu, J. Huang, KERL: A knowledge-guided reinforcement learning model for sequential recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 209–218.

[37] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, T.-S. Chua, Learning intents behind interactions with knowledge graph for recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 878–887.

[38] B. Bi, H. Ma, B.-J. Hsu, W. Chu, K. Wang, J. Cho, Learning to recommend related entities to search users, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 2015, pp. 139–148, http://dx.doi.org/10.1145/2684822.2685304.

[39] C. Rudnik, T. Ehrhart, O. Ferret, D. Teyssou, R. Troncy, X. Tannier, Searching news articles using an event knowledge graph leveraged by wikidata, in: Companion Proceedings of the 2019 World Wide Web Conference, 2019, pp. 1232–1239, http://dx.doi.org/10.1145/3308560.3316761.

[40] V. Setty, K. Hose, Event2Vec: Neural embeddings for news events, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 1013–1016.

[41] S. Gottschalk, E. Demidova, EventKG+ TL: Creating cross-lingual timelines from an event-centric knowledge graph, in: European Semantic Web Conference, Springer, 2018, pp. 164–169, http://dx.doi.org/10.1007/978-3-319-98192-5_31.

[42] L. Gao, J. Wu, Z. Qiao, C. Zhou, H. Yang, Y. Hu, Collaborative social group influence for event recommendation, in: Proceedings of the 25th ACM International Conference on Information and Knowledge Management, 2016, pp. 1941–1944, http://dx.doi.org/10.1145/2983323.2983879.

[43] T.-Y. Liu, et al., Learning to rank for information retrieval, Found. Trends® Inf. Retr. 3 (3) (2009) 225–331.

[44] L. Wu, D. Hu, L. Hong, H. Liu, Turning clicks into purchases: Revenue optimization for product search in e-commerce, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 365–374.

[45] E. Palumbo, G. Rizzo, R. Troncy, Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 32–36.

[46] R. Guo, X. Zhao, A. Henderson, L. Hong, H. Liu, Debiasing grid-based product search in e-commerce, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2852–2860.

[47] K. AlGhamdi, M. Shi, E. Simperl, Learning to recommend items to wikidata editors, in: International Semantic Web Conference, Springer, 2021, pp. 163–181, http://dx.doi.org/10.1007/978-3-030-88361-4_10.

[48] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: A survey, Knowl.-Based Syst. 151 (2018) 78–94, http://dx.doi.org/10.1016/j.knosys.2018.03.022.
[49] S. Cao, W. Lu, Q. Xu, Deep neural networks for learning graph representations, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, 2016.
[50] W.L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: Methods and applications, IEEE Data Eng. Bull. 40 (3) (2017) 52–74.
[51] M. Ou, P. Cui, J. Pei, Z. Zhang, W. Zhu, Asymmetric transitivity preserving graph embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1105–1114, http://dx.doi.org/10.1145/2939672.2939751.
[52] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (12) (2017) 2724–2743.
[53] R. Xie, Z. Liu, M. Sun, et al., Representation learning of knowledge graphs with hierarchical types, in: IJCAI, 2016, 2016, pp. 2965–2971.
[54] K. Toutanova, X.V. Lin, W.-t. Yih, H. Poon, C. Quirk, Compositional learning of embeddings for relation paths in knowledge base and text, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1434–1444.
[55] D. Roy, S. Bhatia, P. Jain, Information asymmetry in wikipedia across different languages: A statistical analysis, J. Assoc. Inf. Sci. Technol. (2021).
[56] S. Gottschalk, E. Demidova, MultiWiki: Interlingual text passage alignment in wikipedia, ACM Trans. Web (TWEB) 11 (1) (2017) 1–30, http://dx.doi.org/10.1145/3004296.
[57] A. Samoilenko, F. Karimi, D. Edler, J. Kunegis, M. Strohmaier, Linguistic neighbourhoods: Explaining cultural borders on Wikipedia through multilingual co-editing activity, EPJ Data Sci. 5 (1) (2016) 9, http://dx.doi.org/10.1140/epjds/s13688-016-0070-8.