A Knowledge Base Approach to Cross-Lingual Keyword Query Interpretation

Lei Zhang(✉), Achim Rettinger, and Ji Zhang

Institute AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
l.zhang@kit.edu

Abstract. The amount of entities in large knowledge bases available on the Web has been increasing rapidly, making it possible to propose new ways of intelligent information access. In addition, there is an impending need for technologies that can enable cross-lingual information access. As a simple and intuitive way of specifying information needs, keyword queries enjoy widespread usage, but suffer from the challenges including *ambiguity*, *incompleteness* and *cross-linguality*. In this paper, we present a knowledge base approach to cross-lingual keyword query interpretation by transforming keyword queries in different languages to their semantic representation, which can facilitate query disambiguation and expansion, and also bridge language barriers. The experimental results show that our approach achieves both high efficiency and effectiveness and considerably outperforms the baselines.

1 Introduction

The ever-increasing quantities of entities in large knowledge bases (KBs), such as Wikipedia, DBpedia, Freebase and YAGO, pose new challenges but at the same time open up new opportunities of intelligent information access on the Web. In recent years, many research activities involving *entities* have emerged, such as entity tagging/extraction from texts and entity linking/disambiguation with KBs. Furthermore, there is an increasing portion of Web search queries involving entities. For example, through query log analysis, Pound et al. [1] found that more than half of Web queries are related to entities. In this regard, the exploitation of *entities and their relations* in information retrieval (IR) research beyond the term-based paradigm has become an area of particular interest. Recently, almost every major commercial Web search engine has announced their work on incorporating entity information from knowledge bases into its search process, including Google's Knowledge Graph, Yahoo!'s Web of Objects and Microsoft's Satori Graph / Bing Snapshots.

Within the context of globalization, *multilingual* and *cross-lingual* access to information has drawn increasing attention. Nowadays, more and more people from different countries are connecting to the Internet and many Web users are able to understand more than one language, e.g., more than half of the citizens in the European Union can speak at least one other language than their mother tongue. While the diversity of languages on the Web has been growing in recent

P. Groth et al. (Eds.): ISWC 2016, Part I, LNCS 9981, pp. 615–631, 2016.
DOI: 10.1007/978-3-319-46523-4_37

years, for most people there is still very little content in their native language. As a consequence of the ability to understand more than one language, users are also interested in Web content in other languages.

In addition, keyword search has proven to be a simple and intuitive paradigm for expressing information needs of users. However, traditional keyword search systems mainly suffer from the following challenges.

Ambiguity. Keyword queries are naturally ambiguous due to the fact that keywords could refer to different things in different contexts. In the multilingual and cross-lingual settings, this problem is more serious, e.g., "*WM*" could refer to the entity Windows_Mobile in English and FIFA_World_Cup in German[1].

Incompleteness. Keyword queries are often incomplete in the sense that instead of the full entity names, only the aliases, acronyms and misspellings are usually given in the queries. In addition, keyword queries might contain concept names representing a set of entities, e.g., "*Internet companies of China*".

Cross-linguality. Multilingual users probably formulate their information needs using native language. However, they are interested in relevant information in any language that they can understand. In some other cases, multilingual users could issue queries consisting of keywords in multiple languages. For example, Chinese users might represent a foreign company using its original name and a local company using its Chinese name, such as "*Google* 百度" with the aim of finding the relationship between Google and Baidu, the largest search engines for English and Chinese, respectively. In addition, specifying the query language should not be the burden of users, which poses new challenges since existing techniques for language detection, such as the well-known character n-gram probability language model, do not work well for short keyword queries [2].

In order to address these challenges, we present a knowledge base approach to cross-lingual keyword query interpretation. The goal is to find entity graphs in the KB matching the keyword query, called *query entity graphs* (QEG), which reflect different semantic interpretations of the keyword query. More specifically, our approach aims to eliminate the ambiguity of keyword queries by exploiting the semantic graph of the KB to generate the top-k QEGs. It supports keyword queries matching entities in their incomplete forms, such as aliases, acronyms and misspellings instead of the full names. In addition, the matching concepts in keyword queries are automatically expanded into sets of associated entities. To the best of our knowledge, this is the first work that allows users to issue keyword queries in any language, which can even contain keywords in multiple languages, for finding the query interpretations grounded in any other languages.

It is noteworthy that this work has been incorporated into XKnowSearch![2], a novel system to entity-based cross-lingual information retrieval (IR) [3]. With the help of the resulting QEGs, XKnowSearch! allows users to further explore entity relations to refine the queries. For bridging the language barriers between queries and documents, XKnowSearch! leverages the cross-lingual query interpretation

[1] *WM* is the abbreviation of *Weltmeisterschaft* in German, which means *World Cup*.
[2] http://km.aifb.kit.edu/sites/XKnowSearch/.

technique in this paper and a cross-lingual semantic annotation system [4] to construct semantic representation of keyword queries and documents in different languages, which are then used for document retrieval.

The main contributions of this paper are: (1) the introduction of a *knowledge base approach to cross-lingual query interpretation* by representing information needs of users as entity graphs to *address the challenges* of traditional keyword search; (2) a *scoring mechanism* for *effective query interpretation ranking* by exploiting various structures in the multilingual KB; (3) a new *top-k query graph exploration algorithm* aimed for *efficient query interpretation generation*; and (4) a *separate evaluation* of the ranking mechanism and the top-k graph exploration algorithm to show that both of them lead to a *considerable improvement* over the baseline methods on *effectiveness and efficiency*, respectively.

The rest of the paper is organized as follows. We firstly introduce the problem in Sect. 2 and provide an overview of our approach in Sect. 3. Details on the scoring mechanism and the top-k query graph exploration algorithm are then presented in Sects. 4 and 5, respectively. Experimental results are presented in Sect. 6. Finally, we survey the related work in Sect. 7 and conclude in Sect. 8.

2 Problem Definition

We deal with the scenarios where queries formulated by users are sets of keywords in any language or even in multiple languages, which are unknown in advance. Given such queries, we first introduce the concepts of *key term* and *key term set* and then define the *query entity graph* (QEG) as the interpretation of a query.

Definition 1 *(Key Term and Key Term Set). Given a query Q consisting of a sequence of keywords $\langle k_1, \cdots, k_n \rangle$, a* key term $t = \langle k_i, \cdots, k_j \rangle$ *is a subsequence of Q with the start index $start(t) = i$ and the end index $end(t) = j$, for which at least one matching entity or concept can be found in the knowledge base. A key term set $T = \{t_1, \cdots, t_m\}$ is a set of* non-overlapping *key terms resulting from Q such that for any t and t' in T either $start(t) \leq end(t')$ or $end(t) \geq start(t')$.*

For example, the keywords "*online companies of US*"could result in many key terms like *online*, *companies*, *online companies*, *US* and *online companies of US*, which could lead to different key term sets, such as {*online, companies, US*} and {*online companies of US*}. The key terms like *online* and *US* could refer to the entities Online_game and United_States, respectively, while *online companies of US* might refer to the concept Internet_companies_of_the_United_States, which has a list of associated entities belonging to it, such as Google, Yahoo! and EBay.

We consider the KB as a directed graph $G_{KB}(N, E)$, where each node $n \in N$ represents an entity and each edge $e(n_i, n_j) \in E$ denotes the relation between entities n_i and n_j. Given the key term sets resulting from a keyword query Q, the query interpretation of Q, i.e., the query entity graph, is defined as follows:

Definition 2 *(Query Entity Graph). A* query entity graph (QEG) *to a keyword query Q, denoted by $G_Q = (N_Q, E_Q)$, is a subgraph of $G_{KB}(N, E)$, which*

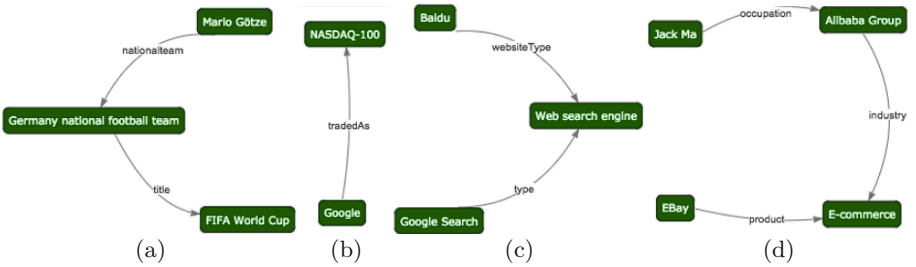

Fig. 1. Example QEGs generated by our system for the queries (a) *"WM Götze"*, (b) *"online companies of US NDX"*, (c) "*Google* 百度 and (d) *"eBay* 马云"

satisfies the following conditions: (1) there exists at least one key term set T *and for each key term* $t \in T$ *there is at least one entity* $n_t \in N$ *that matches* t*. The set of matching entities containing one for every* $t \in T$ *is* $N_T \subseteq N_Q$*; (2) for every possible pair* $n_i, n_j \in N_T$ *and* $n_i \neq n_j$*, there is a path* $n_i \rightsquigarrow n_j$*, i.e., an edge* $e(n_i, n_j) \in E$ *or a sequence of edges* $e(n_i, n_k) \ldots e(n_l, n_j)$ *in* E*, such that every* $n_i \in N_T$ *is connected to every other* $n_j \in N_T$*.*

Problem. We are concerned with the computation of QEGs from keywords in any language or even in multiple languages. Given a query Q, the goal is to find the top-k ranked QEGs, where the ranking is produced by the application of a scoring function $S : G_Q \rightarrow s$. For any given QEG G_Q, S assigns a score s that captures the degree to which G_Q matches the information need of users.

Some examples of the top-ranked QEGs generated by our system for different queries are shown in Fig. 1. To avoid the users' burden of specifying the query languages, our approach does not assume any input language given by users for all the queries. In the query "*WM Götze*", the keyword "*WM*", which could refer to 212 entities in German and 11 entities in English, has been disambiguated as FIFA_World_Cup based on the relation to Mario_Götze. Regarding the query "*online companies of US NDX*", the alias "*online companies of US*" referring to the concept Internet_companies_of_the_United_States has been resolved to the entity Google, which is listed in NASDAQ-100 referred to by the acronym "*NDX*". For the multilingual queries "*Google* 百度" and"*eBay* 马云", our approach can deal with them by supporting query keywords in multiple languages.

3 Overview of the Approach

In this section, we provide an overview of the off-line preprocessing and online computation required in our approach to cross-lingual query interpretation.

Preprocessing. In this work, we use DBpedia as the knowledge base, which is a crowd-sourced community effort to extract structured information from Wikipedia in different languages. In the following, we briefly introduce the offline

cross-lingual grounding extraction, where we construct the cross-lingual lexica[3] by exploiting multilingual Wikipedia to extract the cross-lingual groundings of DBpedia entities and concepts, which correspond to Wikipedia articles and categories, respectively. As Wikipedia provides several useful structures, such as titles of pages, redirect pages, disambiguation pages and link anchors, which associate entities and concepts in DBpedia with terms including words and phrases, also called *labels or surface forms*, all of them can be used to refer to the corresponding resources. In addition, Wikipedia pages in different languages that provide information about the equivalent resources are often connected through the cross-language links. Based on the above sources, for each DBpedia entity or concept grounded in one language we extract its possible surface forms in different languages. More details can be found in our previous work [5,6]. The cross-lingual lexica and the knowledge extracted from DBpedia are indexed for online computation. Based on such indexed data, we are concerned with ranking the query interpretations effectively and propose a scoring mechanism for it, which will be discussed in Sect. 4.

Query Interpretation Computation. In order to compute the QEGs as query interpretations for a keyword query Q, all the key terms are first extracted from Q based on the cross-lingual lexica, which has been also used for finding the matching entities n_t for each key term t, where either t can be used to refer to n_t directly or n_t belongs to a concept that can be referred to by t. Such key terms then result in different key term sets, each of which reflects one possible information need of users. For each key term set T and all the matching entities of its key terms, the exploration of the knowledge graph G_{KB} starts from each matching entity n_t of a key term $t \in T$ to find a connecting element, denoted by n_c, namely an entity that connects at least one starting entity n_t for all $t \in T$. Once a connecting element n_c is found, a QEG can be constructed from a set of paths that start at each n_t and meet at n_c. This process of exploration continues until the top-k QEGs have been achieved. In this paper, we are concerned with performing this query interpretation computation efficiently and propose a new top-k graph exploration algorithm, which will be discussed in Sect. 5.

4 Query Graph Scoring

A keyword query could result in many QEGs all corresponding to possible query interpretations. This section introduces a scoring mechanism that aims to assess the relevance of QEGs for *effective query interpretation ranking*.

4.1 Key Term Set Score

Our approach supports query keywords in multiple languages and we assume that the languages of keywords in a query Q are unknown, such that key terms extracted from Q could be entity/concept names in any language. Therefore,

[3] http://km.aifb.kit.edu/sites/xlid-lexica/.

for each language L, we define the probability $P(t_L)$ that the key term t in L, denoted by t_L[4], is an entity name or a concept name as

$$P(t_L) = \frac{count_{link}(t_L)}{count_{link}(t_L) + count_{text}(t_L)} \tag{1}$$

where $count_{link}(t_L)$ denotes the number of links using t as anchor text and $count_{text}(t_L)$ denotes the frequency of t mentioned in plain text without links in Wikipedia of language L. This estimation is further smoothed by the Laplace smoothing method for the zero probability problem. As the languages of query keywords are not specified, we define the probability $P(t)$ that the key term t refers to an entity or a concept for a set of supported languages $\mathcal{L}$ as

$$P(t) = \max_{L \in \mathcal{L}} P(t_L) \tag{2}$$

All the possible key terms might result in many key term sets that reflect different information needs. Therefore, we define the score of each key term set in the following. Given a keyword query Q, for each resulting key term set T, we take into account both its *importance* and *informativeness*. In general, the more often a key term t is selected as anchor text for the corresponding resources, i.e., t has larger $P(t)$, the more likely that t is important. In addition, the more keywords in Q are covered by all key terms $t \in T$, the more likely that T is informative, since it can reflect more aspects of the initial keyword query. Based on the above observation, we calculate the score of T as

$$\mathrm{S}(T) = \frac{\sum_{t \in T} P(t) \cdot \sum_{t \in T} |t|}{|T|} \tag{3}$$

where $|t|$ is the number of keywords in t and $|T|$ is the number of key terms in T. While $\sum_{t \in T} P(t)$ reflects the *importance* of T, $\sum_{t \in T} |t|$ captures its *informativeness*. The denominator $|T|$ is a normalization factor used to reduce the advantage of T with more key terms. For example, {*online, companies, US*} might result in a larger numerator compared with {*online companies of US*}.

4.2 Entity Matching Score

For each key term t, there might be many entities that can be referred to by t. Assuming that t is in language L, denoted by t_L, we define the probability $P(n_{L'}|t_L)$ that t_L refers to the entity $n_{L'}$ grounded in the target language L' as

$$P(n_{L'}|t_L) = \frac{count_{link}(n_L, t_L) \cdot \tau(n_L, n_{L'})}{\sum_{n_L \in N_L} count_{link}(n_L, t_L)} \tag{4}$$

where $count_{link}(n_L, t_L)$ denotes the number of links using t_L as anchor text pointing to n_L in Wikipedia of language L and N_L is the set of entities that

[4] We use t for a term whose language is not observed and t_L for the same term t whose language is considered as L.

have name t_L. The language mapping function $\tau(n_L, n_{L'})$ is defined as

$$\tau(n_L, n_{L'}) = \begin{cases} 1 & \text{if } n_L \overset{\text{LL}}{\leftrightarrow} n_{L'} \text{ or } n_L = n_{L'}, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where n_L and $n_{L'}$ are considered to be an equivalent entity if they are connected by cross-language links in Wikipedia, denoted by $n_L \overset{\text{LL}}{\leftrightarrow} n_{L'}$. Given a key term t, for which the language is not specified, we calculate the matching score of entity $n_{L'}$ based on the maximal probability $P(n_{L'}|t_L)$ as

$$\mathrm{S_m}(n_{L'}, t) = \max_{L \in \mathcal{L}} P(n_{L'}|t_L) \tag{6}$$

In addition, for each key term t_L in language L that could be a concept name, we first map t_L to the matching concepts C_L in the same language L and then expand each C_L into a set of associated entities in the target language L', denoted by $N_{L'}^{t_L}$, based on the associations between entities and concepts as well as the cross-language links between entities available in the KB (see more details about concept matching and expansion in our TR [7]). Let $|N_{L'}^{t_L}|$ denote the number of entities in $N_{L'}^{t_L}$. For each entity $n_{L'} \in N_{L'}^{t_L}$, we calculate its score based on a uniform distribution over all entities in $N_{L'}^{t_L}$, Similarly, the matching score of entity $n_{L'}$ is calculated based on the maximal score w.r.t. t_L as

$$\mathrm{S_m}(n_{L'}, t) = \max_{L \in \mathcal{L}} \frac{1}{|N_{L'}^{t_L}|} \tag{7}$$

4.3 Query Entity Graph Score

Given a key term set T extracted from a keyword query Q and the set of matching entities N_T containing one for each key term $t \in T$, each QEG, denoted by G_Q^T, is constructed from a set of paths that start at each $n_s \in N_T$ matching a key term $t \in T$ and meet at a *connecting element* n_c. Based on that, we introduce a scoring function to assess the relevance of QEGs as follows

$$\mathrm{S}(G_Q^T) = \sum_{n_s \in N_T} \mathrm{S}(T) \cdot \mathrm{S_m}(n_s, t) \cdot \mathrm{S}(P_{n_s \rightsquigarrow n_c}) \tag{8}$$

where $\mathrm{S}(T)$ is the score of key term set T defined in Eq. 3, $\mathrm{S_m}(n_s, t)$ is the matching score of entity n_s defined in Eqs. 6 and 7, and $\mathrm{S}(P_{n_s \rightsquigarrow n_c})$ captures the score of edges $\langle n_i, n_j \rangle$ along the path $P_{n_s \rightsquigarrow n_c}$ from n_s to n_c, defined as

$$\mathrm{S}(P_{n_s \rightsquigarrow n_c}) = \prod_{\langle n_i, n_j \rangle \in P_{n_s \rightsquigarrow n_c}} \frac{\mathrm{S_r}(n_i, n_j) \cdot (\mathrm{S_p}(n_i) + \mathrm{S_p}(n_j))}{2} \tag{9}$$

where $\mathrm{S_r}(n_i, n_j)$ measures the relatedness between entities n_i and n_j, and $S_p(n)$ reflects the popularity of entity n.

For each pair of entities n_i and n_j, we adopt the Wikipedia link-based measure described in [8] to calculate their relatedness score as follows

$$\mathrm{S_r}(n_i, n_j) = 1 - \frac{\log(\max(|N_i|, |N_j|)) - \log(|N_i \cap N_j|)}{\log(|N|) - \log(min(|N_i|, |N_j|))} \tag{10}$$

where N_i and N_j are the sets of entities that link to n_i and n_j respectively, and N is the set of all entities in the KB.

To measure entity popularity, we exploit both Wikipedia link structure and page view statistics. The second source captures the number of times Wikipedia pages are requested and can be treated as a query log of entities. By leveraging the two sources, we calculate the frequency of entity n as

$$freq(n) = freq_{link}(n) + \beta \cdot freq_{view}(n) \tag{11}$$

where $freq_{link}(n)$ denotes the number of links pointing to n in Wikipedia and $freq_{view}(n)$ denotes the average number of page view requests on n per day. While $freq_{link}(n)$ represents the prior popularity of n in the KB, $freq_{view}(n)$ captures the popularity of n based on user interests. Due to the different scales between Wikipedia link frequency and page view request frequency, $freq_{view}(n)$ is adjusted by a balance parameter $\beta = \frac{\text{total number of links in Wikipedia}}{\text{average number of page views per day}}$, which accounts for the difference in frequencies of Wikipeida links and per-day page view requests. Then the popularity score of each entity $n \in N$ is calculated as

$$\mathrm{S_p(n)} = \frac{freq(n)}{\sum_{n_i \in N} freq(n_i)} \tag{12}$$

5 Top-K Query Graph Exploration

In this section, we present the top-k query graph exploration for *efficient query interpretation generation*. The goal is to find top-k QEGs that connect at least one entity for each key term in a key term set. For pragmatic reasons, existing solutions [9–11] use a maximal path length d_{max}, such that only paths of length d_{max} or less between entities n_i and n_j, denoted by $n_i \leftrightsquigarrow^{d_{max}} n_j$, will be taken into account. Such restriction has also been applied to graph exploration in this work, where d_{max} is set as 6. The algorithm is shown in Algorithm 1.

Input and Data Structures. The input to the algorithm comprises the list of top-m key term sets $LT = \{T_1, \cdots, T_m\}$ and the list $LN = \{N_{t_1}, \cdots, N_{t_n}\}$, where each N_{t_i} is a set of entities matching key term t_i. And d_{max} is the maximal path length applied to the graph exploration. For each entity n, we keep track of the information of paths from an entity n_{start} matching $t_j^i \in T_i$[5] to n, where $n.S_{t_j^i}$ is used to store each pair of the starting entity n_{start} and the score $s_{n_{start}}$ of the path from n_{start} to n, $n.s_{t_j^i}$ and $n.d_{t_j^i}$ are employed to store the maximal score

[5] We use t_j^i to denote a key term t_j belonging to a specific key term set T_i, while t_j represents the same key term without considering the key term sets it belongs to.

Algorithm 1. Top-k Exploration of QEGs

Input: $LT = \{T_1, \cdots, T_m\}$; $LN = \{N_{t_1}, \cdots, N_{t_n}\}$; d_{max}.
Data: $n.S_{t_j^i} = \{\langle n_1, s_{n_1}\rangle, \cdots, \langle n_l, s_{n_l}\rangle\}$; $n.s_{t_j^i}$; $n.d_{t_j^i}$; $LQ_{T_i} = \{NQ_{t_1^i}, \cdots, NQ_{t^i_{|T_i|}}\}$;
$UB_{T_i} = \{ub_{t_1^i}, \cdots, ub_{t^i_{|T_i|}}\}$; $\overline{S(G_Q^{T_i})}$; R; θ.
Result: the top-k optimal QEGs.

foreach $T_i \in LT$ **do**
  foreach $t_j^i \in T_i$ **do**
    foreach $n_{start} \in N_{t_j}$ **do**
      if $\forall t^i_{k\neq j} \in T_i$, $\exists n'_{start} \in N_{t_k}$: $n_{start} \leftrightsquigarrow^{d_{max}} n'_{start}$ **then**
        $s_{n_{start}} \leftarrow \mathrm{S}(T_i) \cdot \mathrm{S_m}(n_{start})$;
        $n_{start}.S_{t_j^i}$.add($\langle n_{start}, s_{n_{start}}\rangle$);
        $n_{start}.s_{t_j^i} \leftarrow s_{n_{start}}$;
        $n_{start}.d_{t_j^i} \leftarrow 0$;
        $NQ_{t_j^i}$.add(n_{start});
      end
    end
    $ub_{t_j^i} \leftarrow \max_{n\in NQ_{t_j^i}} n.s_{t_j^i}$;
  end
  $\overline{S(G_Q^{T_i})} \leftarrow \sum_{ub_{t_j^i}\in UB_{T_i}} ub_{t_j^i}$;
end
while *not all* $NQ \in LQ$ *are empty* **do**
  $T_i \leftarrow \arg\max_{T_i\in LT} \overline{S(G_Q^{T_i})}$;
  $t_j^i \leftarrow \arg\max_{t_j^i\in T_i} ub_{t_j^i}$;
  $n \leftarrow NQ_{t_j^i}.pop()$;
  foreach $n' \in n.neighbors()$ **do**
    $n'.d_{t_j^i} \leftarrow n.d_{t_j^i} + 1$;
    if $n'.d_{t_j^i} < d_{max}$ *and* $\forall t^i_{k\neq j} \in T_i$, $\exists n'_{start} \in N_{t_k}$: $n' \leftrightsquigarrow^{d_{max}-n'.d_{t_j^i}} n'_{start}$ **then**
      foreach $\langle n_{start}, s_{n_{start}}\rangle \in n.S_{t_j^i}$ **do**
        $s'_{n_{start}} \leftarrow s_{n_{start}} \cdot \frac{\mathrm{S_r}(n,n')\cdot(\mathrm{S_p}(n)+\mathrm{S_p}(n'))}{2}$;
        $n'.S_{t_j^i}$.add($\langle n_{start}, s'_{n_{start}}\rangle$);
      end
      $n'.s_{t_j^i} \leftarrow n'.S_{t_j^i}.maxScore()$;
      $NQ_{t_j^i}$.add(n');
      $ub_{t_j^i} \leftarrow \max_{n\in NQ_{t_j^i}} n.s_{t_j^i}$;
      $\overline{S(G_Q^{T_i})} \leftarrow \sum_{ub_{t_j^i}\in UB_{T_i}} ub_{t_j^i}$;
      if $\forall t_j^i \in T_i : n'.S_{t_j^i}$ *is not empty* **then**
        R.add(newQEGsByMergingPath(n'));
        if $R.size() \geq k$ *and* $\max_{T_i\in LT} \overline{S(G_Q^{T_i})} < \theta$ **then**
          return Top-k(R);
        end
      end
    end
  end
end
return Top-k(R);

extracted from $n.S_{t_j^i}$ and the length of shortest path from entities matching t_j^i to n, respectively. For each T_i, LQ_{T_i} is a list of $NQ_{t_j^i}$, each of which is a priority queue of entities on the paths starting at entities matching t_j^i and UB_{T_i} is a list of upper bound scores $ub_{t_j^i}$ for paths starting at entities matching all $t_j^i \in T_i$. For supporting top-k, R is used to keep track of the obtained candidate QEGs during graph exploration and θ denotes the lowest top-k score of the QEG in R.

Initialization. Instead of starting at entities matching each query keyword as described in [9–12], our exploration starts with each matching entity $n_{start} \in N_{t_j}$ for a *key term* $t_j^i \in T_i$ (Lines 1–3). For each starting entity n_{start}, we first check its connectivity (Line 4) to avoid unproductive exploration, which will be discussed later. When the connectivity condition is satisfied, we initialize the score $s_{n_{start}}$ stored in $n_{start}.S_{t_j^i}$, the maximal score $n_{start}.s_{t_j^i}$ and the distance $n_{start}.d_{t_j^i}$ (Lines 5–8). Such starting entities n_{start} are then added into the respective queue $NQ_{t_j^i} \in LQ_{T_i}$ (Line 9) and the upper bound score $ub_{t_j^i}$ for each t_j^i is initialized as the maximal score for all $n_{start} \in NQ_{t_j^i}$ (Line 12).

Connectivity Checking. The aim of checking the connectivity (Lines 4 and 22) is to predict whether an entity n could participate in any QEGs. Given an entity n with path of length $n.d_{t_j^i}$ from n_{start} matching $t_j^i \in T_i$ to n, if it cannot reach some entities n'_{start} matching $t_k^i \in T_i$ $(k \neq j)$ within distance $d_{max} - n.d_{t_j^i}$, it is guaranteed not to be a connecting element and thus the exploration involving n can be avoided. For efficient entity connectivity indexing, we model paths between entities in G_{KB} with length no larger than d as a boolean matrix M_{KB}^d, where each entry m_{ij}^d is 1, if there is a path between entities n_i and n_j of length no larger than d; otherwise, m_{ij}^d is 0. The matrix $M_{KB}^{d_{max}}$ is constructed iteratively using the formula $M_{KB}^{d_{max}} = M_{KB}^{d_{max}-1} \times M_{KB}^1$.

Upper Bound Principle. The upper bound principle captures the goal of exploring only necessary entities for generating the top-k QEGs. The key is to effectively bound the ultimate score of potential QEGs based on the currently explored paths. Since the score of each edge $\langle n_i, n_j \rangle$ defined in Eq. 9 is less than 1, the score of paths satisfy the subset monotonic property, namely $S(P_{n_{start} \rightsquigarrow n}) \geq S(P_{n_{start} \rightsquigarrow n'})$ if $P_{n_{start} \rightsquigarrow n} \subseteq P_{n_{start} \rightsquigarrow n'}$. This implies that the score of a path cannot increase after path expansion during graph exploration and thus the score of all paths starting at entities matching t_j^i can be upper bounded by the maximal score for all $n \in NQ_{t_j^i}$. i.e., $ub_{t_j^i} = \max_{n \in NQ_{t_j^i}} n.s_{t_j^i}$, where $n.s_{t_j^i} = n.S_{t_j^i}.maxScore()$. These upper bound scores indicate the best the potential QEGs resulting from T_i, denoted by $G_Q^{T_i}$, can eventually achieve, such that we define the maximal possible score for all $G_Q^{T_i}$ as $\overline{S(G_Q^{T_i})} = \sum_{ub_{t_j^i} \in UB_{T_i}} ub_{t_j^i}$, which will guide our graph exploration and help with early termination.

Graph Exploration. The graph exploration starts with entities in $NQ \in LQ$ (Line 16). To avoid the unnecessary exploration, our algorithm prioritizes the

entity by the maximal possible score of the potential QEGs. At each iteration, the most promising T_i that could result in the optimal QEG and the key term $t_j^i \in T_i$ with the largest upper bound score $ub_{t_j^i}$ are selected (Lines 17–18). Then the entity n achieving the maximal score of paths from entities matching t_j^i to n is taken from $NQ_{t_j^i}$ (Line 19) and the algorithm continues to explore the neighborhood of n, i.e., all adjacent entities n'. In case that the distance $n'.d_{t_j^i}$ does not exceed d_{max} and the connectivity condition is satisfied (Line 22), we expand the path from each n_{start} to n by adding n', and the score $s'_{n_{start}}$ of each expanded path is calculated and added into $n'.S_{t_j^i}$ (Lines 24–25), where the maximal score $n'.s_{t_j^i}$ is extracted (Line 27). All newly explored entities n' are then added into $NQ_{t_j^i}$ for further exploration (Line 28). Since the maximal score of paths from entities matching t_j^i might change after expansion, the upper bound score $ub_{t_j^i}$ and the maximal possible score $\overline{S(G_Q^{T_i})}$ of potential QEGs are updated accordingly (Lines 29–30). If n' is verified to be an connecting element, i.e., for all $t_j^i \in T_i$, there exists a path from n_{start} matching t_j^i to n' (Line 31), the new QEGs generated by merging paths resulted from n' are added into R (Line 32). Finally, we check whether the exploration can terminate to retrieve the top-k QEGs (Lines 33–35), which will be discussed in the following.

Early Termination. The exploration terminates when one of the following conditions is satisfied: (1) all possible entities have been explored such that there are no further entities in any $NQ \in LQ$ or (2) the top-k QEGs are guaranteed to be obtained. With the goal of retrieving the top-k QEGs, all entities have to be considered as connecting element in order to keep track of all possible QEGs. However, the upper bound principle deals with the requirement of early termination. The maximal possible score $\overline{S(G_Q^{T_i})}$ for all T_i indicates the best the potential QEGs can achieve and the lowest top-k score of the obtained QEGs captures the threshold θ such that only the QEGs with score higher than or equal to θ have a chance to make into the top-k. To conclude that the current k top-ranked QEGs in R are guaranteed to qualify for the final top-k and thus the exploration can terminate, there should be at least k QEGs in R and $\overline{S(G_Q^{T_i})}$ for all T_i must be below θ, i.e., $\max_{T_i \in LT} \overline{S(G_Q^{T_i})} < \theta$ (Lines 33–35).

6 Experimental Results

The experiments were conducted on a virtual machine with 8 Cores at 2.0 GHz and 40GB memory and our system is implemented in Java 8. To assess both effectiveness and efficiency of our approach addressed by Sects. 4 and 5 respectively, we asked volunteers to provide keyword queries along with the underlying information needs. It results in 21 English queries, 10 German queries, 5 Chinese queries and 14 multilingual queries[6], where the query length ranges from 2 to 7

[6] It is a realistic phenomenon that queries consist of keywords in different languages, especially for Chinese users, which is also reflected in the 14 multilingual queries in our experiments, where only English and Chinese keywords are contained.

with an average of 3.24. We assume that the language of each keyword query is unknown and the target language of query interpretations is English[7].

6.1 Effectiveness Evaluation

For evaluating the effectiveness of query interpretation ranking, which is mainly addressed by Sect. 4, we consider the normalized Discounted Cumulative Gain at rank k, denoted by nDCG@k, as quality criteria, which measures the goodness of a retrieval model based on the graded relevance of the top-k results. According to our query interpretation problem, the results are judged by the volunteers who provide the keyword queries on 0–5 relevance scale based on the criteria such as relevance, completeness and correctness w.r.t. the underlying information needs.

For a comparative analysis, we conducted the experiments with the following approaches: (1) the baseline using an online *machine translation* service[8] and a *keyword-based scoring* function described in [11], denoted by *MT+KS*; (2) the baseline using our *cross-lingual lexica* for keyword-to-entity mapping and the *keyword-based scoring* same as (1), denoted by *CL+KS*; (3) the baseline using the *machine translation* service same as (1) and an adaption of our query entity *graph scoring* based on *key term* sets, denoted by *MT+GS+KT*; (4) our approach using the *cross-lingual lexica* for entity matching and the query entity *graph scoring* based on *key term* sets as discussed in Sect. 4, denoted by *CL+GS+KT*.

Figure 2(a) illustrates the nDCG@20 of different approaches for the individual queries (Q1-Q50). Our approach CL+GS+KT achieves the best results for 38 queries, while MT+KS, CL+KS and MT+GS+KT perform the best for 9, 16 and

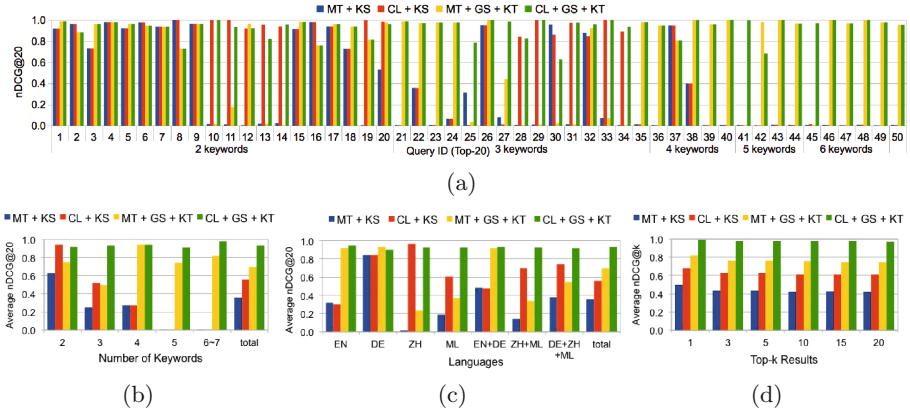

Fig. 2. Experimental results of query interpretation effectiveness

[7] In our experiments, we use English as the target language of query interpretations, but it can be easily extended to other languages.

[8] In our experiments, we used Google Translate for translating queries in different languages to English by selecting the input language option as "Detect language".

28 queries, respectively. Comparing the two methods with keyword-based scoring function, i.e., MT+KS and CL+KS, it is observed that using our cross-lingual lexica (CL) performs better than the machine translation service (MT) in most cases (e.g., Q10-Q14). There is a similar conclusion for the approaches based on our query entity graph scoring, i.e., MT+GS+KT and CL+GS+KT (e.g., Q27-Q31). Based on the further comparison between MT+KS and MT+GS+KT as well as CL+KS and CL+GS+KT, our query entity graph scoring based on key term sets (GS+KT) considerably outperforms the keyword-based scoring (KS) (e.g., Q38-Q50). By taking advantage of both CL and GS+KT compared with MT and KS, CL+GS+KT apparently achieves the best results in most cases.

Figure 2(b) illustrates the impact of query length l, i.e., the number of keywords, on query interpretation effectiveness. While our approach CL+GS+KT is stable for different l, the results of other approaches change considerably when l varies. More specifically, the performance of the approaches using keyword-based scoring (KS), i.e., MT+KS and CL+KS, decreases rapidly when l increases. This is due to the fact that when l is larger, the query entities are usually expressed by more than one keyword such that the keyword-to-entity mapping doesn't work well.

The impact of languages on query interpretation is shown in Fig. 2(c). For English queries (EN), by comparing MT+KS with CL+KS and MT+GS+KT with CL+GS+KT, MT and CL exhibit only minor differences because no cross-lingual mapping is needed when the input and target languages are both English. However, MT+GS+KT and CL+GS+KT still considerably outperform MT+KS and CL+KS respectively, because GS+KT has a clear advantage over KS. For German queries (DE), all approaches achieve comparable results for two reasons: (1) the entities in German queries are usually expressed by compound keywords or their abbreviations, e.g., "*Fußball-Weltmeisterschaft*" or "*WM*" corresponding to "*FIFA World Cup*", such that the keyword-based scoring yields a similar performance to ours; (2) the machine translation service works well when translating from German to English. For Chinese queries (ZH), CL+KS and CL+GS+KT considerably outperform MT+KS and MT+GS+KT because the machine translation service (MT) doesn't work well for translating entity names from Chinese to English compared with our cross-lingual lexica (CL). In addition, in Chinese queries each entity is usually split by users as one compound keyword such that CL+KS even yields slightly better results than CL+GS+KT. Obviously, CL+GS+KT achieves the best results for multilingual queries (ML), where MT+KS and MT+GS+KT perform the worst because the machine translation service (MT) cannot deal with the keywords in multiple languages simultaneously. The experimental results for different combinations of the query languages are also shown in Fig. 2(c), where our approach CL+GS+KT achieves the best results (with nDCG@20 > 0.9) for most cases.

Figure 2(d) illustrates the results of nDCG@k for different k. We observe that the performance of all approaches decreases slightly when k becomes larger. Among these approaches, CL+GS+KT achieves the most stable performance, e.g., MT+KS, CL+KS, MT+GS+KT and CL+GS+KT yield 15 %, 10 %, 8 % and 2 % performance degradation respectively, when k varies from 1 to 20.

6.2 Efficiency Study

For assessing the efficiency of query interpretation generation, which is mainly addressed by Sect. 5, we conducted the experiments with several approaches: (1) the *keyword-based exploration* from each keyword matching entity [12], denoted by *KE*; (2) the *top-k algorithm* on top of the *keyword-based exploration* [11], denoted by *KE+Top-k*; (3) our key term *set-based exploration* starting from the entities matching the extracted key terms, denoted by *SE*; (4) our graph exploration incorporating only *connectivity checking*, denoted by *SE+CC*; (5) our graph exploration incorporating only *early termination*, denoted by *SE+ET*; (6) our approach incorporating both *connectivity checking* and *early termination* into the graph exploration as discussed in Sect. 5, denoted by *SE+CC+ET*.

We start with a comparison between different approaches for the individual queries. The experimental results for computing the top-20 query interpretations for Q21-Q50 with query length from 3 to 7 are illustrated in Fig. 3(a). For the sake of space, we omit the results for Q1-Q20 with query length 2, where individual times do not exhibit significant differences. Clearly, SE outperforms KE for the long queries (e.g., Q36-Q50), where 42 % performance improvement has been achieved on average, while the performance of SE for short queries is slightly better than KE (e.g., Q21-Q35) or similar to KE (e.g., Q1-Q20). Such differences are primarily due to the number of starting entities for the graph exploration as shown in Fig. 3(b). While both connectivity checking (CC) and early termination (ET) contribute to the performance improvement individually, the incorporation of both of them into SE yields the best results. Compared with the baselines KE and KE+Top-k, our approach SE+CC+ET achieves a considerable performance improvement in most cases.

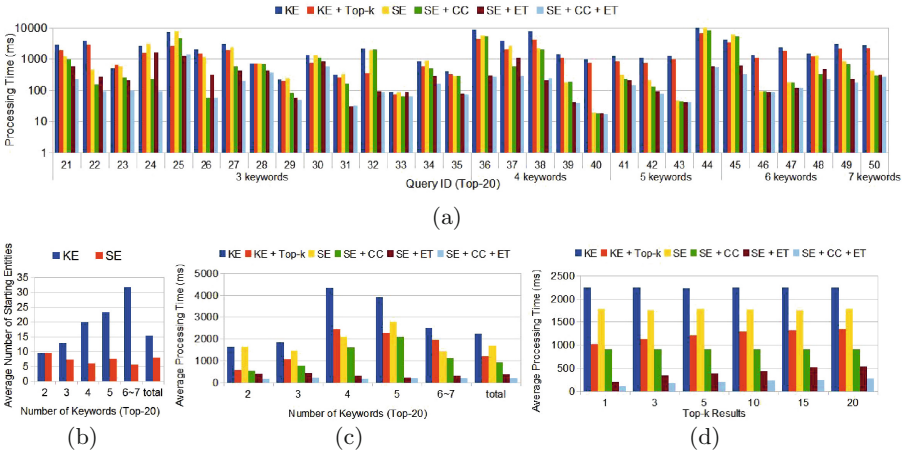

Fig. 3. Experimental results of query interpretation efficiency

We have investigated the impact of query length l on the performance of different approaches. Figure 3(c) shows the average processing time for different l. Compared with KE, the processing time for SE is relatively stable. The reason might be the number of starting entities generated by SE is less sensitive to l as shown in Fig. 3(b). Furthermore, our approaches SE+ET and SE+CC+ET are not sensitive to l due to the application of early termination (ET), while the performance of other approaches changes with varying l.

Figure 3(d) shows the average time for computing top-k query interpretations for different k. The time needed by KE+Top-k, SE+ET and SE+CC+ET decreases rapidly when k becomes smaller. For example, KE+Top-k, SE+ET and SE+CC+ET yield 24 %, 61 % and 62 % time reduction respectively, when k varies from 20 to 1, while the performance of other approaches doesn't change with k since they have to process all results no matter what the value of k is. In total, our approach SE+CC+ET outperforms KE by one order of magnitude and is 5 times faster than KE+Top-k when $k = 20$, and it achieves even more considerable performance improvement for smaller k, e.g., 22 times and 10 times faster than KE and KE+Top-k respectively, when $k = 1$.

7 Related Work

We firstly present the related work to keyword query interpretation and then review some existing work on cross-lingual and concept-based IR.

Keyword Query Interpretation. The main challenges in dealing with keyword queries are their *ambiguity* and *incompleteness*. The use of thesauri to deal with the ambiguity of keywords has a long history. Most commonly, WordNet thesaurus has been found beneficial in disambiguating keywords and in choosing their senses [13]. There are also proposals for mapping keyword queries to elements in an ontology [14], where the resulting semantics provides the basis for identifying the search intents of users. In addition, graph-based approaches [9,11,12] have been widely used to find substructures in structured data, including relational, XML and RDF data. The recent work [15] also aimed to boost the scalability of interactive query construction over large scale data from the perspective of both user interaction cost and performance.

While existing methods only deal with individual keywords in the query, our approach relies on the extracted key terms referring to entities in KBs, which helps to improve both efficiency and effectiveness as shown in our experiments. In addition, most existing methods only focus on the *ambiguity* of keywords. The *cross-linguality* issue has not been studied in the previous work.

Cross-lingual and Concept-based IR. Traditional IR is normally based on the bag-of-words (BOW) models, which have the limitation of retrieving only the syntactically relevant but not the semantically relevant documents. Meanwhile, they suffer from the vocabulary mismatch problem, i.e., queries and documents, which are semantically very related, might contain only few terms in common. This problem is more serious in cross-lingual IR due to the fact that queries

and documents in different languages rarely share common terms. In order to address the problem, different concept-based solutions [16–19] and their cross-lingual extensions [20,21] have been proposed. Instead of the BOW models used in the classic IR, the goal is to capture queries and documents as concepts, such that the relevance can be estimated in the concept space even in the presence of vocabulary gap, especially for cross-lingual IR.

Unlike the previous studies, we developed XKnowSearch!, a novel system to entity-based cross-lingual IR by exploiting multilingual knowledge bases. Based on our cross-lingual query interpretation, XKnowSearch!, to the best of our knowledge, is the first entity-centric system to cross-lingual IR, where users can issue keyword queries in any language (even in multiple languages), for retrieving documents related to the query entities in any other languages.

8 Conclusions and Future Work

We present a knowledge base approach to cross-lingual query interpretation by transforming keywords in different languages to their semantic representation. As the main contributions of this work, we propose a scoring mechanism for effective query interpretation ranking and a top-k graph exploration algorithm for efficient query interpretation generation. A separate evaluation on each of these two aspects has been performed and it shows that our approach achieves promising results w.r.t. both effectiveness and efficiency. In addition, this work has been integrated into XKnowSearch!, a novel system for entity-based cross-lingual IR. As future work, we would like to extend our approach by taking into account entity relations expressed in keyword queries to construct the QEGs. And it is essential to perform further evaluation to show the promising results of our query interpretation can carry over to the performance of cross-lingual IR.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 611346.

References

1. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: WWW, pp. 771–780 (2010)
2. Baldwin, T., Lui, M.: Language identification: the long and the short of the matter. In: HLT-NAACL, pp. 229–237 (2010)
3. Zhang, L., Färber, M., Rettinger, A.: XKnowSearch! exploiting knowledge bases for entity-based cross-lingual information retrieval. In: CIKM (2016)
4. Zhang, L., Rettinger, A.: X-LiSA: cross-lingual semantic annotation. PVLDB **7**(13), 1693–1696 (2014)
5. Zhang, L., Färber, M., Rettinger, A.: xLiD-Lexica: cross-lingual linked data lexica. In: LREC, pp. 2101–2105 (2014)
6. Zhang, L., Rettinger, A., Thoma, S.: Bridging the gap between cross-lingual NLP and DBpedia by exploiting Wikipedia. In: NLP&DBpedia Workshop (2014)

7. Zhang, L., Rettinger, A.: Exploiting knowledge bases for entity-based multilingual and cross-lingual information retrieval. Technical report. http://km.aifb.kit.edu/sites/XKnowSearch/TR.pdf
8. Witten, I., Milne, D.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: WIKIAI, pp. 25–30 (2008)
9. He, H., Wang, H., Yang, J., Yu, P.S.: BLINKS: ranked keyword searches on graphs. In: SIGMOD, pp. 305–316 (2007)
10. Li, G., Ooi, B.C., Feng, J., Wang, J., Zhou, L.: Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In: SIGMOD, pp. 903–914(2008)
11. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query candidates for efficient keyword search on graph-shaped (RDF) data. In: ICDE, pp. 405–416 (2009)
12. Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Desai, R., Karambelkar, H.: Bidirectional expansion for keyword search on graph databases. In: VLDB, pp. 505–516 (2005)
13. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: SIGIR, pp. 61–69 (1994)
14. Tran, T., Cimiano, P., Rudolph, S., Studer, R.: Ontology-based interpretation of keywords for semantic search. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 523–536. Springer, Heidelberg (2007)
15. Demidova, E., Zhou, X., Nejdl, W.: Efficient query construction for large scale data. In: SIGIR, pp. 573–582 (2013)
16. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: SIGIR, pp. 178–185 (2006)
17. Bendersky, M., Croft, W.B.: Discovering key concepts in verbose queries. In: SIGIR, pp. 491–498(2008)
18. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: IJCAI, pp. 1606–1611 (2007)
19. Egozi, O., Markovitch, S., Gabrilovich, E.: Concept-based information retrieval using explicit semantic analysis. ACM Trans. Inf. Syst. **29**(2), 8 (2011)
20. Sorg, P., Cimiano, P.: Cross-language information retrieval with explicit semantic analysis. In: CLEF (Working Notes) (2008)
21. Potthast, M., Stein, B., Anderka, M.: A Wikipedia-based multilingual retrieval model. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 522–530. Springer, Heidelberg (2008)