ENTITY RANKING FOR DESCRIPTIVE QUERIES

Kai Hong[1], Pengjun Pei[2]†, Ye-Yi Wang[3], Dilek Hakkani-Tür[4]*

[1]University of Pennsylvania, Philadelphia, PA, USA
[2]Twitter, San Francisco, CA, USA
[3]Microsoft, Redmond, WA, USA
[4]Microsoft Research, Mountain View, CA, USA

ABSTRACT

We investigate the problem of entity ranking towards descriptive queries, that aims to match entities referred in user queries to entities of a large knowledge base (KB). Entity ranking faces the primary challenge of the sparseness of entity related data, such as various ways of referring to an entity. The lack of sufficient variations of entity referring expressions in KB makes it difficult to find entities referred in user queries, especially when the queries are descriptive. We tackle this problem by enriching KB entries using web documents and query click logs. First, we propose a novel method of injecting textual information from web documents to the KB on a large scale. Since the number of web documents can be large, we propose to use keyword extraction and summarization techniques for compactly representing entity-related information. Second, we mine web search query logs to link entities to existing queries. Experiments show significant improvements after the KB enrichment, compared with two competitive baselines. We also achieve further improvements by combining the data from these two resources.

Index Terms— entity ranking, descriptive user queries, web documents mapping, query click logs, keyword extraction, summarization

1. INTRODUCTION

While searching the web or using spoken dialog systems (SDS), oftentimes people are looking for one or more entities. Structured semantic knowledge graphs (e.g., Freebase [1]) provide an appropriate resource for responding to such user queries [2, 3] and for bootstrapping spoken language understanding (SLU) models [4, 5]. Given that simple queries such as *"who is Kobe Bryant's wife"* can be easily located in the knowledge base (KB), it is more difficult to return the correct entity for queries that include an entity description, but not an explicit referral, such as *"TV show with nerds in LA"* (referring to the *"Big Bang Theory"*). As while the KB includes entities and their relations, it doesn't necessarily include how people refer to these in natural interactions.

In this paper, we aim at finding the correct entities towards descriptive queries in a large structured KB. We investigate different approaches of performing data enrichment for the KB. We build different **profiles** for each entity in the KB, which we regard those **profiles** as unstructured text. The simplest profile is constructed by using the original entity data in the KB, that describes the basic properties of an entity. For example, a movie entity has directors, actors, release date and synopsis as its *entity attributes*. Moreover, popular

About Time

The time traveller's strife

After dominating the landscape of British middle-of-the-road romcoms, *Notting Hill* and *Four Weddings* writer Richard Curtis returns with just his third film as director.

Fortunately, *About Time* is more coherent and less stick-your-fingers-down-your-throat than either of its predecessors, *Love Actually* or *The Boat That Rocked*. Yes, it's a romance, but one that feels a little more soulful.

It's driven by Domhnall Gleeson as Tim, an aspiring lawyer who learns – at the age of 21 – a family secret from his father (Bill Nighy). Put simply, the men in their family have the ability to travel back in time. They can't rock up at any point in history, but they can return to various points in their own lives. All they need to do is disappear into a dark space – a wardrobe will do – and will themselves back.

Fig. 1: A review of the movie *"About time"*. The red lines highlight the terms useful for mapping the user query *"show me the movie where a man travels to the past using a wardrobe"* to the movie.

entities in the KB are often linked to several webpages (Wikipedia pages, official websites, etc.). Intuitively, one can use the textual information from those webpages to generate profiles for the entities in the KB. We regard these two profiles as baselines in our work.

Nevertheless, using values of the entity attributes and entity webpages are not enough in terms of coverage and variations. On the other hand, web documents on the World Wide Web include rich information expressed in different forms. Consider the query *"show me the movie where a man travels to the past using a wardrobe"*, where the correct answer is the movie *"About time"*. Given the term *wardrobe* does not appear amongst the KB entities or the webpages in the KB, it is mentioned in various websites which describe this movie (see Figure 1). For this reason, we propose a novel approach of making profiles by mapping a large number of web documents with the entities. We inject the textual information from the mined related web documents into the entity index. Experiments show that this profile is able to cover 93.3% of the unigrams (excluding function words) ever appeared in the descriptive user queries, much higher than the coverage of KB attributes (43.4%) and webpages in the KB (79.1%).

The web documents profile has high coverage of query terms. However, it takes great storage space. Therefore, we propose two novel methods of summarizing the web documents. First we extract keywords from the documents related to the entity. Second we employ multi-document summarization to generate summaries for the entities. We show that keyword extraction provides a good compromise between storage space and performance for entity ranking.

We also investigate to what extent mining query click logs (QCLs) is helpful for this task. A wide range of applications have benefited from the analysis of QCLs, including web document ranking [6, 7], slot filling [8], and spoken language understanding (SLU) [9].

*Work done when the author was an intern at Microsoft Research.
†Work done when the author was at Microsoft.

Queries	Movie name	Human evaluation results
General Form: (a/the) movie(s)/film(s) that/where/when *descriptions*	...	Excellent/Good/Fair/Bad
movie that **takes place** in **Walmart** with **Ashley Judd**	Where the heart is	Excellent
films that **holds record** for most **wrecked cars**	The Junkman	Excellent
movie where **guy** and **girl meet** at **Empire State**	When Harry met Sally	Excellent

Table 1: Samples of descriptive queries, targeted entities and human evaluation results

We collected the web search queries that clicked on the URLs related to an entity. We investigate two approaches of mining entity related URLs: one uses the URLs of the websites provided in the KB, the other employs the URLs linked with the entities derived from large scale web documents mapping. The two profiles mined from QCLs yield substantial improvement over all other profiles. To the best of our knowledge, this is the first work which applies QCLs for ranking entities towards verbose and descriptive queries.

The profiles we build take the form of free text. Hence, we employ a document retrieval formula derived from Okapi BM25 [10] and BM25+ [11] during the online entity ranking process. We experiment with a dataset consisting of descriptive queries from human-machine interactions in the movies domain. We demonstrate that our large-scale web document mapping and QCL mining approaches achieve significant improvement over the main webpages and KB attribute baselines. These two methods have similar performance for entities that are not frequently queried by web search users. We show that further improvement can be achieved by combining the output of different rankers.

2. PRIOR WORK

Web search community has long been interested in finding and ranking entities, for example the TREC Entity Track [12] and INEX Entity-Ranking Track [13]. Different from us, both TREC and INEX use webpages or Wikipedia XML collections to represent the entities. For TREC, their task is focused on relational queries, such as "What art galleries are located in *Bethesda, Maryland*?". The entities (e.g. *"Bethesda, Maryland"*) within the queries are also explicitly labeled. For INEX, they provide a query topic instead of a user query, including a keyword query, a description and a narrative (i.e., natural explanation of the information needed). Both tasks are different from ours, where we do not include any named entities explicitly labeled in our queries. Our queries are simply a description of the targeted entity from human-computer interactions.

Our work also takes the idea of entity linking in knowledge base population [14, 15]. Entity linking aims at mapping named entities in unstructured texts to entities in KBs [16]. Different from our task, the targeted entity appears explicitly in the documents for entity linking, and that their focus is to determine the correct entity in the reference KB.

Researchers have also looked into semantic interpretation towards queries in spoken dialog systems [4]. There, word sequences in the queries are classified into an in-domain semantic attribute such as a movie genre slot, and those attributes can be interpreted in the forms appearing in the KB for querying the KB.

Web text corpora have been used to extend KBs for improving semantic parsing and discovering relations. [3] combine the semantic information in the KB and syntactic information in dependency parsed documents to improve semantic parsing. [17] show that the coverage of relations in the KB can be extended by adding edges labeled by mining latent features from web documents. A more closely related work is [18], where they show entity ranking on Wikipedia can benefit from external links, using Wikipedia as a pivot. Different from [18], our approach mines related web documents on a larger scale rather than using the links from the original entity (Wikipedia page). Moreover, we focus on descriptive queries occurring in natural conversational interactions.

A number of studies focus on using QCLs for KB related applications. For example, [2] proposed a system for approximately matching web queries to KB by mining QCLs. [19] improved the quality of entity recommendation by leveraging general and vertical query search logs. More related to ours are the works of [20, 21], where query logs are directly applied to entity ranking. Different from our work, their queries there only include named entities (e.g. Milan, University of Milan) [21] or simple queries (e.g. fruit salad recipe, side salads) [20]. The focus of their work lies in better identifying user's preference towards existing possibilities, rather than populating the KB using existing queries towards difficult queries.

3. DATA AND EVALUATION METRIC

We gathered 525 descriptive queries in the movies domain from human-computer interactions. The queries are seven or more words, each including a description targeted at one or more movies. The average length of the user queries is 9.9 words. Sample queries are shown in Table 1. All of the queries are in English, however they can target non-English films. During entity ranking, we restrict our search space to all of the movie entities in our KB (about 800K), including non-English films. Note here that the number is larger than Freebase, where they have about 250K movie instances.

We collected the top 30 entities returned from our rankers, as well as the entities corresponded to the top 30 webpages returned from Bing, Google and Yahoo!. The webpages returned from search engines are manually mapped to the entity entries in the KB. Human evaluators then label the entities as *Excellent*, *Good*, *Fair* or *Bad* given the query. Among all 525 queries, 441 of them correspond to a single *Excellent* entity. Since the entities not labeled as *Excellent* do not correspond to a related query, we regard the excellent entity (entities) as our only correct answer(s), similar to [21]. The queries are split randomly, where 275 of them are used as development set, 250 are used as test set.

We employ Recall@K and Mean Reciprocal Rank (MRR) to evaluate the performance of the ranker, as 84% queries correspond to a single correct answer [21, 22]. The two evaluation metrics are defined as follows. Let Q denote a set of queries. Let L_k denote the top k entities returned corresponding to a query $q \in Q$. Recall@K is equal to 1 if the correct entity (or one of the correct entities) is retrieved within L_k, otherwise it is equal to 0. The final Recall@K for Q takes the mean of Recall@K for all $q \in Q$. Since Recall@K does not take position into account, we employ MRR as another metric. The reciprocal rank $RR(q)$ of a query q takes the multiplicative inverse of a rank for the first correct entity, except that we have $RR(q) = 0$ if the correct entity is not recalled within L_{30}.

For a set of queries Q, the MRR is defined as the mean of $RR(q)$ for all $q \in Q$.

4. GENERAL FRAMEWORK

We show our system framework in Figure 2. It consists of two components: offline KB enrichment and runtime entity ranking.

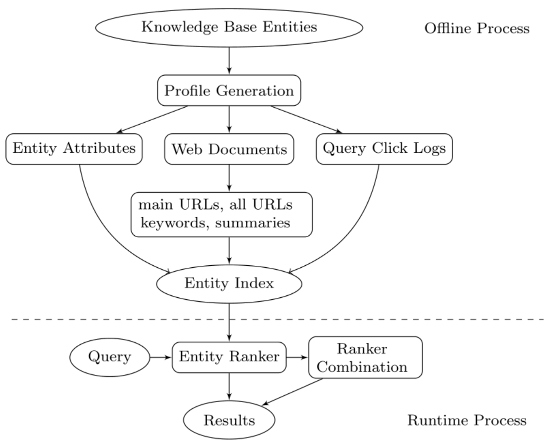

Fig. 2: General framework for entity ranking

Offline Knowledge Base Enrichment. We perform data enrichment for the KB by forming profiles from different knowledge sources. The simplest profile is constructed by directly using the values of attributes from the KB entities. We form four profiles from web documents: (1) main webpages using the URLs provided by the KB; (2) web documents mined on a large scale. (3) keywords of the set of web documents (4) summaries generated for the set of web documents. The QCLs profile is formed by mining related queries, where we take either queries that clicked on the main URLs or the queries that clicked on the URLs derived by web documents mapping. All profiles take the form of free text, without doing any analysis on the structure of the knowledge sources.

Runtime Entity Ranking. Given a user query, we use one of the profiles derived from the previous step as the background document for the entity. The ranking procedure retrieves the most relevant entity according to the relevance score between the selected profile and the user query. By using profiles, we transform the problem of entity ranking into document ranking. We employ a variation of BM25 and BM25+ as our approach of scoring the profiles corresponded to the entities. A MRR-motivated approach of combining the rankers can be applied at the end of the framework.

5. DATA ENRICHMENT PROCESS

We here introduce how we perform data enrichment from web documents and query click logs. Before that, we empirically show the challenge of current KBs when dealing with descriptive queries.

5.1. Query Unigram Recall in Knowledge Base

The motivation of our work is based on the assumption that large KBs lack coverage of descriptive information in user queries. To verify this, we compute the macro recall of the non-stopwords[1] from the 275 queries on the development set. We bold all of the non-stopwords we attempt to recall in Table 1. We directly check if the word has ever appeared in the values of attributes of the corresponding entity (entities) marked as an *excellent* match. Not surprisingly, only 43.8% of the query unigrams are recalled. This verifies the assumption that many details are missing in the KB. We use the profile constructed from KB attributes as our first baseline (ATTRIBUTES).

5.2. Enrichment from Web Documents

Main Web Documents. Entities in the KB provide URLs of its main webpages, such as entity webpages in Wikipedia. We use those main webpages as our first enrichment method of constructing profiles as well as a strong baseline. Indeed, Wikipedia pages themselves have been used as entities for many KB tasks, including entity ranking [13] and entity linking [15]. Compared to the structured information in KB, web documents offer richer descriptions in unstructured or semi-structured form. Among all 800K entities in the movie domain, 71.5% of them have at least one equivalent English webpage, mostly from Wikipedia, IMDB and Rotten Tomatoes. We directly use the main webpages as background document profiles for the entities (MAINURLS).[2] The query unigram recall can be improved to 79.1% from the main webpages.

Large Scale Web Documents Mapping. Main webpages are never enough for providing sufficient descriptions. The problem is even more severe for domains other than movies, as Wikipedia only features one-ninth of all the entities in Freebase [24]. Therefore, we attempt to find web documents related to entities on a larger scale. We take a dump of the English webpages on the WWW, and try to map all those webpages to the entities in the KB. The mapping approach is shown in Figure 3. We assert that a website is related to an entity when the following two standards are satisfied: (1) the entity name appears in the title of the webpage; (2) one of the value of the entity attribute appears in the body of the webpage.

Fig. 3: Finding webpages related to the entities

Our proposed approach of finding candidate web documents has the advantage of ensuring high coverage for potential information. We gathered at least one webpage for 83.0% of all entities, each entity has an average of 16.7 related web documents. Our large scale mapping approach greatly improves the query unigram recall to 93.4%. We name this profile as WEBDOCS.

[1] We use the stopword-list from the SMART system [23], augmented with the 200 most frequent words from the webpages we mined in Section 5.2. The augmentation aims at excluding common movie-related words within web documents, such as "film", "youtube".

[2] We only extract the English webpages in our work, since all of the queries are in English.

5.3. Summarizing the Web Documents

The WEBDOCS profile includes rich, sometimes redundant information for the KB entities. The number of documents related to an entity could be large, which makes it difficult to store the documents. There is also possibility that the redundant information would be too noisy for data mining. Therefore, we propose two approaches to summarize those web documents.

Keyword Extraction. We build profiles for the entities using keywords. Apart from the benefit of saving space, this approach also eliminates the long tail word-occurrences of an entity. We investigate two unsupervised approaches of extracting keywords: word frequency and TF*IDF. Here we remove all stopwords. Figure 4 depicts the query unigrams recall, changing with the number of keywords extracted. Most query unigrams can be covered within the top 3000 most frequent words. Employing word frequency also leads to consistently higher recall compared with TF*IDF. Therefore, we build our profile by extracting the top 3000 most frequent unigrams and bigrams (KEYWORD). We also keep the frequency of these keywords in the profile.

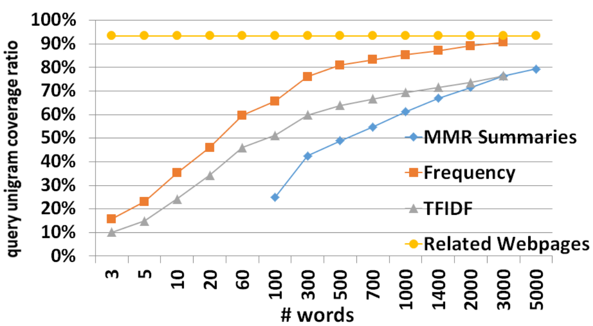

Fig. 4: Query unigram recall on the development set, changing with the number of keywords or size of the summary.

Generic Summarization. We make profiles by summarizing the web documents. Using summaries has the advantage of generating human-readable text. It also offers more flexibility of representing textual information compared with using keywords. Here we apply the classical Maximal Marginal Relevance (MMR) algorithm [25] for summarization. This algorithm greedily picks the sentence to be included in the summary until a predefined word limit. At each iteration, the greedy algorithm selects the sentence which is maximally relevant to the input set and minimally redundant to the sentences in the current summary. Let $D = \{s_1, s_2, ...s_n\}$ denote the document set and let C denote the current summary, the sentence score is:

$$Score(s_i) = \lambda \cdot Sim(s_i, D) - (1-\lambda) \cdot \max_{s_j \in C} Sim(s_i, s_j)$$

Following [25], we use cosine similarity as the similarity metric (Sim) and assign $\lambda = 0.7$, as tuned on the development set. For each entity, we produce a 5000-word summary as the summary profile (SUMMARY). Figure 4 shows the query unigrams recall, changing with the summary size. A 5000-word summary can cover 79.3% of the query terms, similar to the recall of main webpages (79.1%).

5.4. Enrichment from Query Click Logs

We construct profiles by injecting existing web search queries in the query click logs (QCLs) to the entities in the KB. To this end, we collected six months of web search sessions from January 1, 2013 to June 30, 2013. These sessions consist of QCLs based on IE browsing history. We build a tripartite graph G including entities nodes V, URL nodes (webpages) U and query nodes Q, similar to [8, 19]. A sample tripartite graph G is shown in Figure 5. The edges between the URLs (U) and queries (Q) are directly extracted from the QCLs. The weight $w_e(q, u)$ represents the number of times the user clicked the URL u after query q, $u \in U$, $q \in Q$.

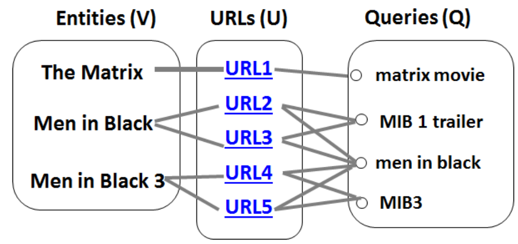

Fig. 5: Finding queries related to the entities from web search query click logs.

We investigate two approaches of connecting the URLs (U) to the entities (V) in the KB, similar to the methods in Section 5.2. For the first approach, an entity is linked with the main URLs in the KB. For the second approach, an entity is linked with the URLs mined by large scale web documents mapping (as shown in Figure 3). The second approach conducts more aggressive query mining. For both methods, the number of times an entity v ($v \in V$) related to a web search query q is denoted as:

$$freq(v, q) = \sum_{e(v,u) \in G} w_e(q, u)$$

The two profiles take the name of QCLS-MAIN (from main URLs) and QCLS-LARGE (from a large number of URLs). The query unigrams recall for the two profiles are 77.1% and 83.5%, respectively.

6. RUNTIME ENTITY RANKING

Here we describe the runtime entity ranking function used in our system, which is a variation of Okapi BM25 [10] and BM25+ [11].

The Okapi BM25 retrieval function, as presented in [22], scores a document d corresponded to a query q as follows:

$$F(q, d) = \sum_{t \in q} [\log \frac{N}{df_t}] \cdot \frac{(k+1)tf_{td}}{k((1-b) + b \cdot (\frac{L_d}{L_{ave}})) + tf_{td}}$$

Even if the algorithm has been proposed for two decades, it still remains the state-of-the-art for a wide variety of tasks [26]. Let t denote the terms in query q and let tf_{td} denote the frequency of term t in d. Let L_d denote the length of document d, let L_{ave} be the average document length for the whole collection. Within the IDF term $\log \frac{N}{df_t}$, N is the total number of documents, df_t represents the number of documents including the term t in the collection. k and b are the free parameters in this formula.

BM25+ [11] diagnoses the problem that the term frequency in BM25 is not properly lower-bounded. This approach has been shown appropriate for verbose queries in document ranking.

Two changes are made for our formula. First, we extract both unigrams and bigrams from the queries. Second, we do not include

Profiles	Attributes	MainURLs	WebDocs	Keywords	Summary	QCLs-main	QCLs-large
Query unigram recall %	43.4	79.1	93.4	89.8	79.3	77.1	83.5
Recall@1 %	32.0	42.8†	49.2†	46.4†	32.4	**67.6**†	**64.4**†
Recall@5 %	52.4	62.4†	**69.2**†	66.0†	46.0	**82.0**†	**78.0**†
Recall@10 %	56.8	67.2†	**74.8**†	71.2†	52.0	**84.8**†	**80.4**†
MRR %	40.9	52.1†	**58.1**†	55.2†	39.0	**74.2**†	**70.1**†
Parameters(L_b,k,b)	1, 1.4, 0.1	1, 1.4, 0.1	0, 6, 0.3	0, 6, 0.3	1, 1.4, 0.1	1, 1.4, 0.1	1, 1.4, 0.1
Storage size (inverted indexed)	1.0 GB	2.6 GB	62 GB	7.3 GB	2.5 GB	0.5 GB	1.3 GB

Table 2: Results of entity rankers from different profiles. Query unigrams recall on development set is shown in the first row. The Recall@k and MRR are reported on the test set. Statistically significant improvements ($p < 0.05$) over MAINURLS baseline are in **bold**. † denotes statistical improvements ($p < 0.05$) over the ATTRIBUTES baseline.

the IDF terms in our formula, with the observation that some frequent informative terms tend to be over punished. The formula employed in our entity ranker is below:

$$F(q,d) = \sum_{t \in q} c_t \cdot [\frac{(k+1)tf_{td}}{k((1-b)+b\cdot(\frac{L_d}{L_{ave}}))+tf_{td}} + L_b]$$

Here c_t is a parameter of balancing between unigram and bigram weighting. We fix $c_t = 1$ for unigrams, $c_t = 0.1$ for bigrams. The parameter L_b is used for switching between BM25 ($L_b = 0$) and BM25+ ($L_b = 1$). The selection of c_t, L_b, k and b are tuned based on the performance on the development set.

7. EXPERIMENTS

7.1. Comparison of Entity Profiles

Table 2 shows the performance using our seven profiles for the test set, measured by Recall@k (k = 1, 5, 10) and MRR. We also show the query unigram recall on the development set. As our two baselines, using KB attributes and main URLs as profiles achieve a M-RR of 41.8% and 52.9%, respectively. The WEBDOCS profile constructed by large scale mapping has the highest query unigram recall. It also gains a 6.0% improvement on MRR and at least 6.4% improvement on Recall@k over the MAINURLS baseline.

As for methods of summarizing the web documents, using keywords achieves a better performance than using main URLs, but worse than using all information mined from the web documents. Therefore, the process of removing infrequent words does not help the entity ranker. Most of the query terms are recalled within the SUMMARY profile (79.3%). However, our entity ranker only achieves a MRR of 39.0%, worse than using KB attributes. One possible reason is because this profile does not store the frequency of each term, which causes information loss. Another explanation is that many irrelevant information are included in the summaries. It is worth noting that for many summarization tasks, it is easy to outperform MMR. Thus there is still need to explore other approaches of summarizing the web documents for future work.

On the other hand, the profile constructed by mining QCLs perform remarkably, with a gain of 15.8% (QCLS-MAIN) on MRR compared with the WEBDOCS profile. The advantage of QCLS-MAIN and QCLS-LARGE profiles over the other approaches are significant on all evaluation metrics. Interestingly, using textual information from more web documents achieves better performance than using that of main URLs, however mining the queries related to those URLs does not improve over using the queries related to main URLs. The decrease in performance might be because some noisy queries have been collected from the clicks to other URLs.

We also examine the space required to store our profiles. It takes 181GB to store all webpages mined for the WEBDOCS profile, while the space needed is much smaller for the other profiles (less than 20GB). We further examine the storage space after inverted-index for the unigrams and bigrams for the profiles we formed, as shown in Table 2. Compared with the WEBDOC profile, keyword extraction saves the storage space by 88.2%, with a 2.9% decrease of performance evaluated by MRR. It gives a good compromise between the performance and the storage space.

7.2. Ranker Combination

We show that combining the result from different rankers can lead to further improvement. Here we use a heuristic approach motivated by mean reciprocal rank (MRR). We take the average of reciprocal rank from the rankers we want to combine as the new score of the entity. For example, if an entity is ranked as 1, 2 and 4 by three rankers, it would have an average score of $\frac{1}{3}(1 + 1/2 + 1/4) = 0.583$.

We investigate two groups of ranker combinations. We first combine our best ranker using QCLs (QCLS-MAIN) and best ranker using web documents (WEBDOCS). The combination of these two rankers outperforms all single rankers (See Table 3). The improvement over QCLS-MAIN is significant for Recall@5 (5.2%), Recall@10 (6.4%) and MMR (2.6%). The success of ranker combination suggests that the query click logs and web documents carry complementary information.

Since QCLs are strong indicators and oftentimes unavailable, we combine the five rankers not using QCLs. The combination leads to substantial improvement over WEBDOCS, our best ranker not using QCLs (see Table 3). Even though the profiles used (apart from ATTRIBUTES) are all subsets of the WEBDOCS profile, they have different ways of compactly representing knowledge. Combining the decision of these models is helpful for finding the correct entity.

	WebDocs+QCLs-main	All except QCLs
Recall@1 %	68.8†	52.0
Recall@5 %	**87.2**†	74.4†
Recall@10 %	**91.2**†	78.8
MRR %	**76.8**†	62.0†

Table 3: Performance of ranker combination. Statistically significant improvements ($p < 0.01$) over QCLS-MAIN are in **bold**. † denotes statistical improvements ($p < 0.01$) over WEBDOCS.

7.3. Performance Analysis by Entity Popularity

We analyze the performance of different profiles by the popularity of entities referred in the web search query logs. *Entity popularity* is defined as the total number of times the users clicked on URLs of the entity in the KB, computed from the QCLs. We split the queries in the test set into two bins of equal size according to the *entity popularity* the query refers to. For the queries which have more than one targeted entity, we use the one with a higher popularity count.

Figure 6 shows the MRR towards the entities of high and low popularity. For high popularity entities, the QCLS-MAIN profile outperforms all others. The advantage of WEBDOCS over MAINURLS is not high, which indicates the main webpages of popular entities include considerable details. However, for low popularity entities, the MRR of using QCLs drops to 64.9%. Using WEBDOCS achieves slightly worse performance (60.9%) than QCLS-MAIN, while outperforms using MAINURLS by about 10%. The advantage of profile combination can also be better pronounced for low popularity entities. Interestingly, the performance for WEBDOCS is higher among low popularity entities. This might be because a larger number of unrelated web documents get injected to the KB for the entities which are queried often. To sum up, the WEBDOCS profile and the ranker combination process can better demonstrate their capabilities for the entities which are not queried a lot in QCLs.

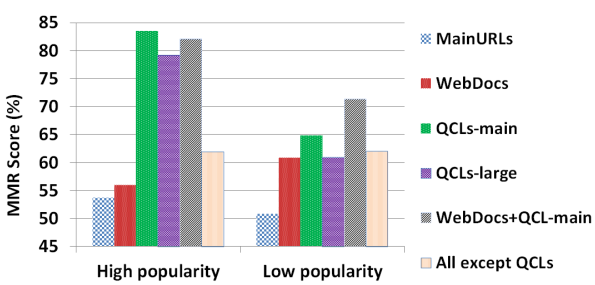

Fig. 6: Mean reciprocal rank (MRR) stratified by queries targeted at high and low popularity entities. Each group includes 125 queries.

8. CONCLUSION

This paper focuses on ranking entities towards descriptive user queries from human-computer interactions. We propose novel methods of performing data enrichment to deal with the sparseness of data in the knowledge graph. The entity profile constructed by large scale web document matching has a high coverage towards descriptive natural language queries. Performing data enrichment using large-scale webpage mapping and query click logs achieves significant improvement over two competitive baselines. We also investigate methods of summarizing the web documents of an entity. Moreover, we show further improvement can be achieved by combining the output of these rankers.

9. REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *SIGMOD*, 2008, pp. 1247–1250.

[2] Tao Cheng, Hady Wirawan Lauw, and Stelios Paparizos, "Fuzzy matching of web queries to structured data," in *ICDE*, 2010, pp. 713–716.

[3] Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen, "Reading the web with learned syntactic-semantic inference rules," in *EMNLP-CoNLL*, 2012, pp. 1017–1026.

[4] Asli Çelikyilmaz, Dilek Hakkani-Tür, and Gökhan Tür, "Statistical semantic interpretation modeling for spoken language understanding with enriched semantic features," in *SLT*, 2012, pp. 216–221.

[5] Larry Heck and Dilek Hakkani-Tur, "Exploiting the semantic web for unsupervised spoken language understanding," in *SLT*, pp. 228–233.

[6] Eugene Agichtein, Eric Brill, and Susan Dumais, "Improving web search ranking by incorporating user behavior information," in *SIGIR*, 2006, pp. 19–26.

[7] Thorsten Joachims, "Optimizing search engines using clickthrough data," in *KDD*. 2002, pp. 133–142, ACM.

[8] Gökhan Tür, Dilek Hakkani-tür, Dustin Hillard, and Asli Çelikyilmaz, "Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling," in *Proceedings of Interspeech*, 2011.

[9] Dilek Hakkani-Tür, Gökhan Tur, Larry P. Heck, Asli Çelikyilmaz, Ashley Fidler, Dustin Hillard, Rukmini Iyer, and Sarangarajan Parthasarathy, "Employing web search query click logs for multi-domain spoken language understanding.," in *ASRU*, 2011, pp. 419–424.

[10] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford, "Okapi at trec-3," 1995, pp. 109–109.

[11] Yuanhua Lv and ChengXiang Zhai, "Lower-bounding term frequency normalization," in *CIKM*, 2011, pp. 7–16.

[12] K. Balog, P. Serdyukov, and A. P. de Vries, "Overview of the TREC 2011 entity track," in *TREC 2011 Working Notes*. NIST.

[13] Gianluca Demartini, Tereza Iofciu, and Arjen P De Vries, "Overview of the inex 2009 entity ranking track," in *Focused Retrieval and Evaluation*. 2010, pp. 254–264, Springer.

[14] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis, "Overview of the TAC 2010 knowledge base population track," in *TAC*, 2010.

[15] Heng Ji and Hoa Trang Dang, "Overview of the TAC 2011 knowledge base population track," in *TAC*, 2011.

[16] Heng Ji and Ralph Grishman, "Knowledge base population: Successful approaches and challenges," in *ACL-HLT*, 2011, pp. 1148–1158.

[17] Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell, "Improving learning and inference in a large knowledge-base using latent syntactic cues," in *EMNLP*, 2013, pp. 833–838.

[18] Rianne Kaptein, Pavel Serdyukov, Arjen De Vries, and Jaap Kamps, "Entity ranking using wikipedia as a pivot," in *CIKM*, 2010, pp. 69–78.

[19] Patrick Pantel and Ariel Fuxman, "Jigs and lures: Associating web queries with structured entities," in *ACL-HLT*, 2011, pp. 83–92.

[20] Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel, "Ranking entities using web search query logs," in *ECDL*, Berlin, Heidelberg, 2010, pp. 273–281.

[21] Davide Mottin, Themis Palpanas, and Yannis Velegrakis, "Entity ranking using click-log information," *Intell. Data Anal.*, vol. 17, no. 5, pp. 837–856, 2013.

[22] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA, 2008.

[23] G. Salton, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[24] Hrushikesh Mohapatra, Siddhanth Jain, and Soumen Chakrabarti, "Joint bootstrapping of corpus annotations and entity types," in *EMNLP*, 2013, pp. 436–446.

[25] Jaime Carbonell and Jade Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *SIGIR*, 1998, pp. 335–336.

[26] Yuanhua Lv and ChengXiang Zhai, "When documents are very long, bm25 fails!," in *SIGIR*, New York, NY, USA, 2011, pp. 1103–1104, ACM.