Ranking Entities for Web Queries Through Text and Knowledge

Michael Schuhmacher, Laura Dietz and Simone Paolo Ponzetto
Data and Web Science Research Group
University of Mannheim, Germany
{michael,dietz,simone}@informatik.uni-mannheim.de

ABSTRACT

When humans explain complex topics, they naturally talk about involved entities, such as people, locations, or events. In this paper, we aim at automating this process by retrieving and ranking entities that are relevant to understand free-text web-style queries like Argentine British relations, which typically demand a set of heterogeneous entities with no specific target type like, for instance, `Falklands_War` or `Margaret_Thatcher`, as answer. Standard approaches to entity retrieval rely purely on features from the knowledge base. We approach the problem from the opposite direction, namely by analyzing web documents that are found to be query-relevant. Our approach hinges on entity linking technology that identifies entity mentions and links them to a knowledge base like Wikipedia. We use a learning-to-rank approach and study different features that use documents, entity mentions, and knowledge base entities – thus bridging document and entity retrieval. Since established benchmarks for this problem do not exist, we use TREC test collections for document ranking and collect custom relevance judgments for entities. Experiments on TREC Robust04 and TREC Web13/14 data show that: i) single entity features, like the frequency of occurrence within the top-ranked documents, or the query retrieval score against a knowledge base, perform generally well; ii) the best overall performance is achieved when combining different features that relate an entity to the query, its document mentions, and its knowledge base representation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.4 [**Artificial Intelligence**]: Semantic Networks

Keywords

Entities; Knowledge Bases; Information Retrieval

1. INTRODUCTION

Web search engine research shows an increasing interest in going beyond words, in particular by integrating structured knowledge into the retrieval process, which has recently led to large efforts aimed at building very large entity-centric knowledge bases of ever increasing coverage and complexity [42, 18]. These can be leveraged either as sources of information to be presented directly to the user or as background knowledge sources to be used by retrieval systems to improve the retrieval process itself [8, 14]. But despite the sophistication of existing search engines, the former task of satisfying the user information need by presenting an entity and its related description is still tackled in a very basic way. Common search engines display entity information only for entity-centric queries. For example a query for Falkland islands[1] returns a short description of this British overseas territory, together with information about its capital, currency, etc. However, entity information is missing for more complex queries like Falklands war or broad queries such as Argentine British relations. Consequently, in this work we develop a framework to satisfy the user's information need on any topic from an entity-centric viewpoint by accompanying the document result set with a ranked list of entities that are deemed as relevant for the query.

Entity ranking has recently attracted a lot of attention from researchers, especially in the context of the INEX and TREC initiatives [11, 2, 19]. This line of work is primarily aimed at retrieving entities from a structured knowledge repository based on keyword queries. As a result, the output of these entity ranking systems typically consist of a handful of homogeneous entities of the same type like, for instance, Royal Navy personnel who took part in the Falklands War in response to the query British Navy officer Falklands war. But while this can be adapted to cover many practical search scenarios, it still does not address a large amount of user queries issued as general-purpose search intent, namely those corresponding to complex information needs that could only be satisfied by a list of heterogeneous entities of arbitrary types, covering different aspects of the query. Consider, for instance, the query Argentine British relations. For this query, it is difficult to satisfy the user information need by means of a single entity. Instead, in this case the user probably would like to see a wide spectrum of entities of different types including events (`Falklands_War`), persons (`Margaret_Thatcher`) or even songs (`March_of_the_Malvinas`). Complex queries of this kind are far from being seldom issued. In previous work, Pound et al. [34] classified web-search query types in the context of entity retrieval using real-world querylog data, and showed that 40.6% of sampled queries are entity queries (CJ5 Jeep), 12.1% are type queries (cold medication), and 4.6% are attribute queries (zip code waterville maine). Accordingly, we can conclude that more than 40% of all remaining queries are open domain ones – namely, they are not limited to specific types of entities – which cannot be assigned to any of the previous classes.

CIKM'15 October 19 - 23, 2015, Melbourne, VIC, Australia

ACM 978-1-4503-3794-6/15/10 ...$15.00.
http://dx.doi.org/10.1145/2806416.2806480.

[1]In this paper, we use Sans Serif for words and terms, and `Monospace` for entities.

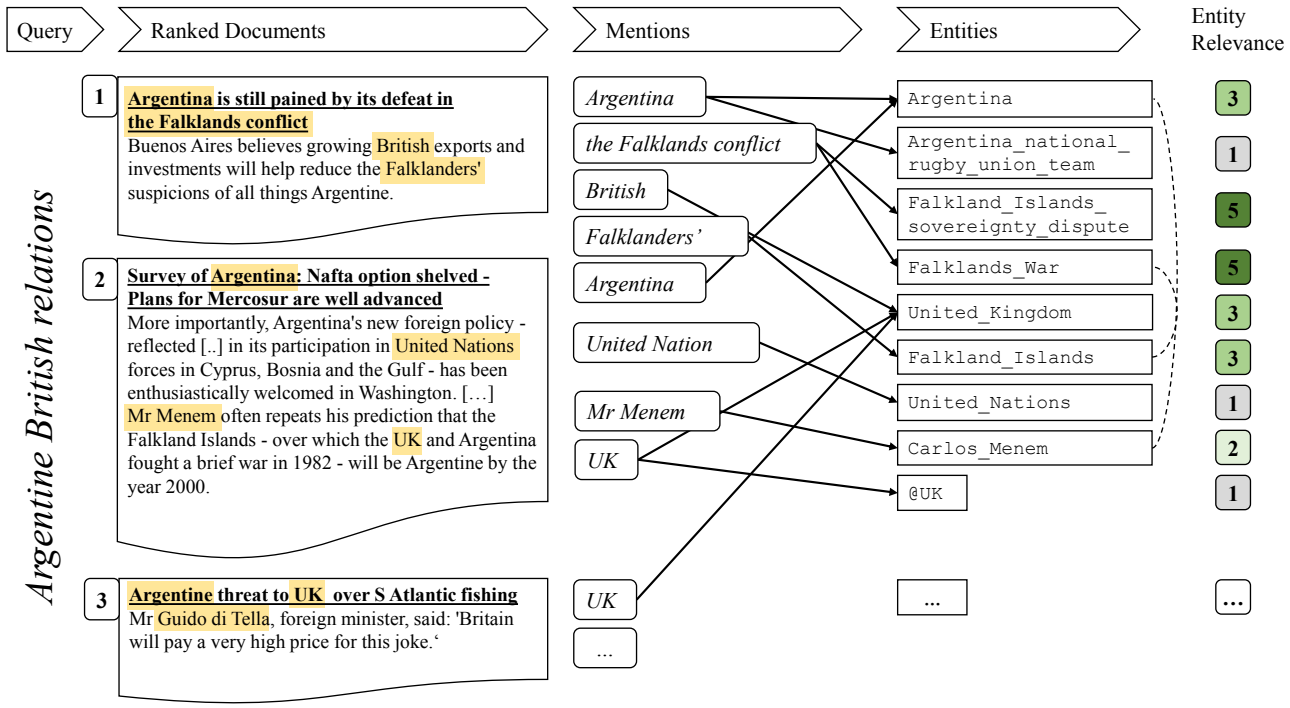


Figure 1: An example of ranking entities for a Web query. Solid arrows represent entity links, dotted lines knowledge base relations.

With this work we aim to fill this gap and specifically address open domain web queries with entity rankings. To this end, we build on a state-of-the-art document retrieval system and an entity linking tool. Entity-linked documents provide us with an initial pool of entity mentions in context. These entities are re-ranked with the goal of providing a ranking of entities according to their relevance for the user query. Key to our approach is the usage of entities that originate from the output of a document retrieval system, as opposed, for instance, to directly rank the entity-centric information stored within an underlying knowledge base. This provides us with a combination of documents and entities as a unified result to the search queries, which can be used later for a variety of higher-level applications like advanced entity-rich search engines [21]: that is, while we focus in this paper on entity ranking only, our method potentially enables the broader vision of a general-purpose search system displaying relevant entities alongside relevant documents or passages as the result to a user query [12], a feature that fits well with and extends existing search interface paradigms (we leave this exploration for future work).

The contributions of this paper are the following ones:

- New task: We formulate a new variant of ad-hoc entity search which is close in spirit to ad-hoc document search and complements existing entity retrieval challenges (e.g, INEX).
- New evaluation dataset: In order to foster a community around joint entity and document retrieval, we create an entity gold standard from queries of established test collections for ad-hoc document retrieval.
- Experimental study: We use ground truth judgments from our dataset to investigate and experiment with different approaches and features arising from the knowledge base and the document collection to evaluate initial solutions for this ranking problem.

To the best of our knowledge, none of the existing entity ranking benchmarks is designed for inter-operation with document retrieval. Therefore, we create a new evaluation set, which we make publicly available and whose details we present in Section 4. Thanks to our dataset, we are able to perform extensive experiments and show that high performance on this task can be achieved, by combining information from heterogeneous knowledge sources – e.g., co-occurrence statistics from large text corpora like ClueWeb, as well as structured knowledge bases such as DBpedia [4] – into a learning-to-rank framework. Our study includes two test sets with different corpus characteristics (news versus web), two document retrieval systems, namely the keyword-based Sequential Dependency Model (SDM, [31]) and the entity-aware EQFE [8], and entity links from the system KBBridge [7] and the FACC1 collection [17].

2. TASK DEFINITION

We tackle the problem of retrieving a ranking of entities in response to a web query. We do not restrict ourselves to address only a limited class of queries. In particular, we aim to retrieve lists of entities for *any* typical web query – even queries that are not posed with entity retrieval in mind. We define an entity as being relevant for a query if a human would mention this entity when providing an answer. Accordingly, the task is to rank entities based on their degree of usefulness for a user to understand what the query is about (i.e., its topic, main themes and involved actors). Ultimately, we aim for an approach that helps to understand queries about complicated topics, such as the Robust04 query Argentine British relations or the TREC Web 2014 query Pink Slime in Ground Beef. Therefore, we focus on informational queries about a general topic – i.e., we do not expect the query to ask for a particular fact or a specific answer [37]. We define our entity ranking task as follows:

- Given: a user-provided textual query (such as Argentine British relations).
- Provided: a background corpus with query-relevant documents that includes entity link annotations from mentions (e.g., the conflict of the Falklands) to entities (e.g., `Falklands_War`) in a background knowledge base.
- Goal: Rank the entities by relevance w.r.t. the query.

Example. We illustrate our problem by means of an example (Figure 1). Initially, we are given the query Argentine British relations and the retrieved query-relevant documents D_q from the document collection (in our experiments we used the TREC Robust04 corpus and the ClueWeb12 corpus, cf. Section 5). Text of these documents is processed by an entity linking system that connects entity mentions from the documents to their equivalent entity in the knowledge base (this work uses entities from DBpedia). The retrieved documents contain a wide spectrum of entities of different kinds such as countries, persons, organizations, and many others. The mentioned entities exhibit different degrees of relevance when compared against the input query. For instance, the `Falklands_War` is a defining event for the diplomatic relations between the United Kingdom and Argentina. However, the entity `United_Nations` does not provide us with very specific information related to the relations between these two countries – i.e., both countries belong to the UN, just like many other countries do. Consequently, our task is to automatically produce a ranking which best correlates with the gold-standard ranking obtained from a set of human judgments: in our case, a ranking which prefers the entities `Falklands_War`, `Argentina`, and `United_Kingdom`, which have been labeled as important for the given query. Note that, in this setting, the notion of 'importance' is not necessarily symmetric: While the `United_Kingdom` is important to understand the topic Argentine British relations, for the entity `United_Kingdom` itself the diplomatic relationships to Argentina are only a minor aspect.

3. ENTITY RANKING FOR WEB QUERIES

We develop a first solution for our entity ranking task that: i) combines heterogeneous knowledge from unstructured document text and structured entity-centric knowledge sources; ii) within a supervised learning approach. Our hunch is that complementary sources of query-relevance for entities can be extracted through entity mentions, their context, and explicit semantic information from a wide-coverage knowledge base through inspecting link structure and article text.

3.1 Learning to Rank Entities for Web Queries

Our study on entity-relevance indicators is performed within a learning-to-rank framework. For that we train different supervised feature-based models on labeled data, an approach that repeatedly demonstrated competitive performance for retrieval tasks [29]. There is a wide range of different learning-to-rank algorithms available, each with their own strengths and weaknesses. In order to reduce the influence of the learning algorithm on our conclusions, we employ different learning-to-rank algorithms in our study. This includes the Ranking Support Vector Machine (SVM, [24]), and an algorithm to optimize an underlying retrieval metric directly, as implemented in the RankLib package[2].

The Ranking Support Vector Machine views the ranking problem as a pairwise classification task and minimizes the number of discordant pairs in Kendall's τ. Therefore, it learns a ranking function that can always be represented as a linear combination of feature vectors, thus making it possible to use also non-linear kernels [24]. We use this to leverage a semantic kernel function that uses relations between entities as a similarity measure (Section 3.6) and also study an alternative linear kernel. In addition, we use a list-wise learning algorithm, here an implementation from RankLib to directly optimize Mean-Average Precision (MAP) and alternatively Normalized Discounted Cumulative Gain (NDCG), thus addressing the so-called metric divergence problem [30]. We use coordinate ascent as an optimization algorithm, since it has demonstrated good performance on low-dimensional feature spaces with limited training data.

This supervised learning setting allows us to study the effect of different document-based and knowledge-based relevance indicators for entity ranking.

3.2 Document Retrieval and Candidate Pool

We start by issuing the query to a document retrieval system and collect the top results. To analyze the impact of this step on the remainder of our approach, we study two retrieval models for two data sets: Robust04 and ClueWeb12.

For the Robust04 dataset, we use the document retrieval method EQFE, which is an entity-aware document retrieval method [8]. This system uses entity links within documents to produce its document ranking. These entity links are created with KBBridge [7], which we also use in our method. In the second experiment on ClueWeb12, we verify our findings by using a different, keyword-based retrieval method, the Sequential Dependency Model (SDM) [31], and a different, existing set of entity links, the FACC1 dataset [17].

In both cases, entity links in high ranked documents are used to build a pool of candidate entities. Consequently, our ranking problem is formulated as the task of comparing the query with the document mentions and associated entities.

3.3 Mention Features

The first feature is based on the number of entity mentions in retrieved query-relevant documents. To this end, count statistics over all the targets of all entity links are collected (notice, that some entity linkers retain multiple targets per link). While we study raw counts (MenFrq) for comparison, we use TF-IDF to weight mention counts across the document collection.

Mention Frequency (MenFrqIdf): The number of occurrences of each entity over all retrieved documents per query, weighted using TF-IDF as follows:

$$\text{MenFrqIdf}(e) = \text{tf}_q(e) \log \frac{N}{\text{df}(e)}$$

This feature is already a strong ranking method by itself (cf. Section 5), since query-relevant documents are likely to contain relevant entities. The downside of this method is that the connection between the query and the entity is only indirectly established – namely through the documents only.

3.4 Query–Mention Features

The second set of features compares entity links more directly with the query. We distinguish between the mention text, i.e. the surface form of the entity links (denoted "M"), and the context of up to ten terms surrounding the entity link (denoted "C").

We define an entity mention as the sequence of words that the entity linking system links to a target entity in the knowledge base. The entity mention is compared to any word in the query. In our example, this means that occurrences of the query term British within entity mentions are an indicator that the target of the links, i.e., entity `United_Kingdom`, is relevant. Likewise, if query terms

[2] `http://people.cs.umass.edu/~vdang/ranklib.html`

such as Relationship occur near entity links, this is also an indicator that the target of the entity link is relevant. We limit the search terms within a ten-term window surrounding the entity link.

We apply different word similarity methods to compare the surface forms or contexts to all query-terms. These similarities are averaged across all mentions for each entity.

String Edit Distance (SED): We compute the normalized Levenshtein String Edit Distance [28] between the query and the mention (context, respectively) as one string, in order to cover basic morpho-syntactic similarity. This is able to capture exact string matches, as well as approximate ones (e.g., due to spelling mistakes within queries and documents).

GloVe (Glo) / JoBimText (Jo): To generalize across different synonyms and word senses, we study the utility of distributional representations of words. The general idea of these distributional thesauri is to model word meanings based on their global frequency of co-occurrences in large text corpora. They can thus identify general semantic similarity between words, without the explicit need for exhaustive context. Here, we focus on two existing distributional semantics models, namely GloVe [33] and JoBimText [3]. As both frameworks associate single words with vectors, we aggregate all word-wise similarities for multi-word mentions and queries. As aggregation functions we study taking the average and the sum. In our setting, using the JoBimText framework can help us to identify, for instance, the high similarity between the query-term relationship and a contextual term like tension.

3.5 Query–Entity Features

The next set of features compares the knowledge base entities directly with those in the query. We achieve this in two ways: (a) we apply entity linking to query keywords and compare the set of linked query entities with the document entities; (b) we perform retrieval on text associated with an entity in the knowledge base with the query terms. For the first option, we leverage a structured knowledge base (DBpedia, Section 3.5.1), and for the second option a semi-structured textual knowledge resource (Wikipedia, Section 3.5.2).

3.5.1 Entity-set Comparison Using DBpedia

We first run the queries through an off-the-shelf entity linker (TagMe, [16]) to collect entities found within the query. This allows us to compare query entities to candidate entities. We incorporate direct matches and short paths between both sets.

a) **Direct Match (QEnt)**: Boolean feature indicating whether the candidate entity is contained in the query entities.

b) **Path (QEntEntSim)**: Whether the query and document entities are connected by some path along the DBpedia graph.

In our example from Figure 1, a direct match is the entity `United_Kingdom`, which is linked from the query term British and the document mention UK.

Besides, we tap into the structure of the background knowledge base by identifying relations between different, albeit related entities. For example `Carlos_Menem` is not mentioned in the query. However, he is related to Argentina, as he used to be its president. To this end, we build upon the work from [36], who extract weighted paths between entity pairs in DBpedia. Here, we consider entities that are involved in the same relation instance or share the same Wikipedia category. We consider all relations from the DBpedia Ontology (prefix `dbo:`), thereby staying agnostic towards particular predicates.

3.5.2 Knowledge Base Retrieval Using Wikipedia

Arbitrary queries can be rather ambiguous and hard to interpret for an entity linking system. Thus, as an alternative we compare the query keywords directly with text associated with entities in the knowledge base. Given the query keywords, we apply two types of information retrieval models on the English Wikipedia, and accordingly build two different kinds of features.

WikiBoolean: All entities returned by a basic standard Boolean retrieval model, based on a full text index over all Wikipedia articles. We bind query keywords with disjunctive operators. This approach tests if at least one query keyword is found within the Wikipedia article of the candidate entity.

WikiSDM: We further use a Galago[3] search index of an English Wikipedia dump. Using the Sequential Dependency Model (SDM, [31]) with collection-level Dirichlet Smoothing, we use the query to retrieve 1,000 Wikipedia articles. The retrieval score is used as a feature as a measure of relevance for the entity. While the WikiSDM feature is computed on the knowledge base only, it is used to re-rank entities that have a high mention frequency in retrieved, query-relevant documents. The effect of WikiSDM therefore includes both knowledge base and document information.

3.6 Entity–Entity Features

Finally, we leverage the relations between entities from the documents themselves. This is in contrast to Section 3.5.1, where we use direct relations and shared categories as a feature to capture similarity between entities from the query and the document.

At the heart of these features lies the idea that a ranking decision between two candidate entities should be influenced by how similar these two entities are to each other. This is a helpful signal for entity ranking, especially in the case of entities having very different features (as computed w.r.t. the query), but nevertheless being related to each other. Consider, for instance, the two document entities `Falkland_Islands` and `Falklands_War` in Figure 1. Both entities are strongly related due to the fact that the Falklands war took place on the Falkland Islands. This information is available from DBpedia through the relation `dbo:place`.

To capture the similarity between candidate entities, we use the same algorithm that gives rise to the feature QEntEntSim to compute a similarity matrix between all candidate entities. Since the value of the feature depends on the pair of entities that are ranked with respect to each other, we cannot derive a static feature for this. To include this feature into our learning problem, we consequently extend the ranking SVM and replace the linear kernel with a custom mixed kernel. This mixed kernel consists of a linear kernel for the standard features, combined with a semantic smoothing kernel [5] for entity–entity similarity.

The semantic kernel was originally proposed to cover semantic similarity between words: in this work, we use it instead to include the similarity between entities. Hence, with this kernel we can incorporate the information that `Falklands_War` has a strong relation to the islands it took place in, namely the `Falkland_Islands`, without changing any of the query-related features directly.

4. REWQ DATASETS

To the best of our knowledge, there exists no dataset for evaluating entity ranking for web queries; therefore we decided to create our own, the Ranking Entities for Web Queries dataset (REWQ, pronounced "rookie"). We study queries from well-known document

[3] `http://lemurproject.org/galago.php`

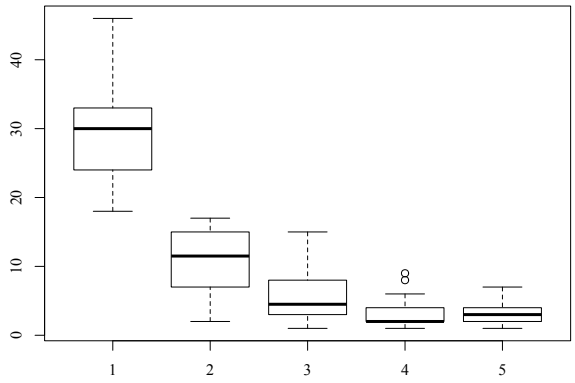


Figure 2: Boxplot for ground truth labels of REWQ-Robust (rounded to a 5-level scale) for 25 query with 50 entities each.

retrieval benchmarks, namely the TREC Robust04 data set [40] and the TREC Web 13/14 data[4], and use the respective corpora as a background corpus to provide a benchmark in two complementary experimental settings. Gold annotations for both datasets are available in the online appendix of this paper.[5]

REWQ-Robust: We aim at covering complex web queries, and accordingly build on the TREC Robust 2004 data set [40] as starting point. In order to study the interplay between document and entity retrieval, we start with a document set that has a high chance of including query relevant entities. Accordingly, we select the 25 top-performing queries of the EQFE system on the dataset (as measured by mean average precision) and collect for each of these the top 19 documents. The final dataset consist of the 50 entities with the highest mention frequency per query.

Entity relevance, defined as the degree of importance that entities play to explain the query (Section 2), was annotated separately by a pool of four different annotators, with each query being annotated by at least two annotators on a 5-level Likert scale ranging from 1 (non-relevant) to 5 (highly relevant). Annotation disagreement were resolved by a standard adjudication process. The final relevance score is obtained by taking the arithmetic mean across all annotations, leading to the final relevance metric $rel \in [1-5]_{\mathbb{R}}$.

We depict the distribution of the annotation scores in the box-plot in Figure 2. The relevance scores indicate that the absolute majority of entities is not relevant. Furthermore, there are only a few highly relevant entities in our dataset. This is a result of our annotation guidelines, which require entities to be marked as highly relevant only if they clearly satisfy the information need expressed in the query. As a result of this, the relevance judgments provided by humans annotators are rather strict. In the case of the query Argentine British relations, for instance, the entity `Falklands_War` receives a high relevance score (5). `Argentina`, instead is annotated only with a mildly relevant score (3.3) because, while being relevant w.r.t. query, it does not actually answer the question about the relationship between both countries.

REWQ-ClueWeb: We use a second dataset to benchmark our ranking problem from a different, yet complementary perspective. We build upon TREC Web 2013/2014 with the ClueWeb12 corpus, since it is an established dataset. It furthermore comes with a set of publicly available entity annotations, namely the FACC1 dataset [17], which allows us to evaluate our approach in another setting by using a entity linker other than KBBridge [7]. Instead of focusing, as in REWQ-Robust, on some specific subset of queries like the top-performing ones from EQFE [8], it is based on a random subset of 22 queries from TREC Web2013/2014. For each query the (entity-agnostic) Sequential Dependency Model [31], implemented in the Galago search toolkit, is used to retrieve the top 20 documents, thus leaving out the effects and potential gains given by EQFE's entity-linked documents. The final dataset consists of all entities per query – we heuristically filter out those entities occurring less than three times to remove many spurious entities from the data. This way we relax the REWQ-Robust's assumption of using only the top-*k* entities per query. Entity relevance is finally annotated in a standard (e.g, TREC-style) way using binary relevance judgments (again, we use the definition of relevance as in Section 2).

5. EXPERIMENTS

5.1 Evaluation Metrics

We use normalized discounted cumulative gain (NDCG) [23] as evaluation metric for both datasets, so as to capture the intuition that a higher relevance should be honored by higher rank. We additionally report Mean Average Precision (MAP) [41] when evaluating against binary relevance judgments using the REWQ-ClueWeb data. The values are computed with the TREC Evaluation tool[6] and reported in the following as the arithmetic mean over all queries.

5.2 Experimental Settings

We evaluate our approach using all features from Section 3 within two learning-to-rank methods: (a) the SVM^{rank} implementation from Joachims [25] and (b) the coordinate ascent methods as implemented in RankLib. Evaluation for both methods and datasets is performed with a linear 5-fold cross-validation. Parameter tuning for the SVM is done with an additional, random train-validation split, i.e. 3/5 training data, 1/5 parameter validation data, and 1/5 test data. For each fold, features are individually normalized with $x_{norm} = (x - \mu)/\sigma$, where mean μ and standard deviation σ are computed using only the training data folds. We compare the learned feature combinations against the following methods:

Mention Frequency (*MenFreqIdf*): A ranking consisting only of the idf-weighted mention frequency feature. This feature's individual performance comes primarily from the quality of the initial document retrieval: Relevant documents should contain relevant entities. In case the entity linker provides more than one entity per mention (as for the REWQ Robust04 dataset with KBBridge), we take this ranked list of candidate entities into account by replacing the mention frequency ($\text{tf}_q(e)$) with the total reciprocal rank (TRR):

$$\text{tf}_q(e) = \text{TRR}_q(e) = \sum_{d \in D(q)} \sum_{m} \frac{1}{rank_{e,m,d}}$$

Ranking by TRR combines the frequency of occurrence of the mentions with the entity linker's confidence scores on the linking of the mentions to their entities.

Wikipedia Fulltext Index (*WikiSDM*): A ranking based on the scores from a Sequential Dependency Model [31] retrieved from a retrieval index of Wikipedia text (using weight parameters from Dalton et al [8]). This baseline is closest in spirit to INEX-like entity retrieval from Wikipedia [26] and is the alternative to our approach of issuing the query against a document retrieval system and then

[4] `http://lemurproject.org/clueweb12/`

[5] `http://rewq.dswlab.de`

[6] Version 9.0; `http://trec.nist.gov/trec_eval/`

Table 1: Evaluation results for REWQ Robust dataset. We report differences w.r.t. the best performing reference method (here WikiSDM), statistically significant improvements are denoted with † (paired t-test p-value ≤ 0.05).

Method	ndcg	Δ%	ndcg10	Δ%
RankLib	0.936	†3.7	0.817	†11.6
SVM (w/ SK)	0.926	†2.6	0.804	†9.7
SVM (w/o SK)	0.923	2.2	0.796	†8.7
WikiSDM	0.903	0.0	0.733	0.0
MenFrqIdf	0.885	-2.0	0.694	-5.3
WikiPR	0.778	-13.8	0.440	-40.0

Table 2: Evaluation results for REWQ ClueWeb12 dataset. We report differences w.r.t. the best performing reference method (here MenFrqIdf), statistically significant improvements are denoted with † (paired t-test p-value ≤ 0.05).

	map	Δ%	ndcg	Δ%	ndcg10	Δ%
RankLib	0.328	†9.0	0.572	†3.4	0.710	†10.0
SVM (w/ SK)	0.278	-7.8	0.545	-1.6	0.646	0.1
SVM (w/o SK)	0.308	2.2	0.563	1.6	0.675	4.4
MenFrqIdf	0.301	0.0	0.554	0.0	0.646	0.0
WikiSDM	0.234	-22.3	0.515	-7.0	0.613	-5.1
WikiPR	0.075	-75.1	0.328	-40.8	0.126	-80.5

link the document to the knowledge base instead of querying the knowledge base directly.

Wikipedia PageRank (*WikiPR*): A ranking obtained by applying the (unpersonalized) PageRank algorithm to the link structure of Wikipedia, thus ranking entities by their global authoritativeness.

5.3 Results on the REWQ-Robust Dataset

We present our results in Table 1, where we compare three learning-to-rank models – namely, SVM-rank with (w/ SK) and without the Semantic Kernel (w/o SK) from Section 3.6, as well as the coordinate ascent model from RankLib. All reference methods (MenFrqIdf, WikiSDM, and WikiPR) achieve high NDCG scores, with WikiSDM performing best with slightly above 0.9. This indicates that they all provide hard-to-beat methods to benchmark our learning-to-rank approach against. The low performance of WikiPR, in contrast, suggests that authoritativeness correlates, in our setting, only marginally with entity relevance (0.096 Spearman's rank correlation coefficient). Error analysis reveals that entities ranked high by PageRank were very general ones linked to by many other entities, e.g. `Earth`, `United_States`, etc.. When comparing the reference methods, we observe that the best results are obtained when ranking using the Sequential Dependency Model (WikiSDM), an approach that has been shown to perform well in the context of the INEX competitions.

Finally, we observe that our learning-to-rank methods outperform all reference methods, thus reaching an overall NDCG score of 0.936 – the difference is statistically significant (according to a paired t-test, p-value $\leq \alpha = 0.05$). By re-ranking the entities with our method, we gain up to 3.7% in NDCG over the input ordering (MenFrqIdf), even though we have 'only' 50 entities per query. Among the different rankers, RankLib performs better than SVM-rank, with the semantic kernel (w/ SK) improving the SVM-rank results slightly (+.003 in NDCG).

When looking at the NDCG@10 scores, we observe the same trends on larger scale. That is, i) our supervised rankers beat all reference methods, which nevertheless achieve a very competitive performance, with WikiSDM ranking highest among them; ii) RankLib outperforms SVM-rank, which achieves better scores when using a semantic kernel. The larger relative improvements between baselines and supervised rankers suggests that our feature-based approach makes a difference in particular to move the relevant entities from the long tail into the top ten.

Narrative evaluation. To provide more insights into the actual method output, Table 3 shows the results obtained from RankLib for queries from REWQ-Robust. Queries are sorted by the average ground truth values (*gt*) for the top 3 entities, thus showing queries with meaningful entities at the top. Among the top queries we find e.g poliomyelitis and post polio, for which we are able to retrieve expected and relevant, but not surprising entities like `Poliomyelitis`, `Polio_vaccine` or `Jonas Salk`, resulting in an NDCG@10 score of 0.879. Another interesting query with a very high NDCG@10 of 0.931 is territorial waters dispute, for which not so well-known, yet relevant entities like `United_Nations_Convention_on_the_Law_of_the_Sea` are ranked high, as well as examples of specific water disputes taking place in the Mediterranean Sea (`Aegean_dispute`) and the Pacific Ocean (`Kuril_Islands_dispute`). The query on the bottom, agoraphobia, has a low *gt* value because the initial document retrieval in combination with the entity linking could not obtain any really useful entities besides `Charles_M._Schulz`.

Error analysis on the low-performing queries reveals that our method suffers from errors in the entity links. For the query Argentine British relations, for instance, the top retrieved entity is `Argentina_rugby_union_team`, which is actually an artifact of systematic errors from the entity linking system, which incorrectly links mentions like Argentine or Argentina to the national rugby team, and not to the country (`Argentina`). This suggests that a better entity linking could further boost our performance. Another source of errors comes from the retrieval system itself – e.g., low performance on the query agoraphobia comes from the rather noisy pool of documents we start with to collect potentially relevant entities. Finally, low performance on some queries are due to their degree of difficulty, as highlighted by fine-grained queries for very specific domains (e.g., hydroponics), where additional knowledge could potentially help.

5.4 Results on REWQ-ClueWeb Dataset

In Table 2 we report our results on the ClueWeb12 portion of the REWQ dataset. Similar to the Robust04 results, the single features perform quite well on their own. In contrast to the Robust dataset, the best single feature is the MenFrqIdf features. Again, our learning-to-rank approach outperforms all reference methods consistently across all measures, both when using RankLib and SVM-rank (up to +9.0% MAP, +3.4% NDCG, +10.0% NDCG@10).

Also in line with the Robust04 findings, the greater relative improvements of our method for the NDCG@10 value suggest that our features make a difference in particular for the top ranked entities. The performance of the SVM with Semantic Kernel (w/ SK) is worse in contrast, the MAP and NDCG scores are even below the MenFrqIdf feature. Because the NDCG@10 value is at par with the MenFrqIdf, we suspect that the knowledge base links between entities used by the Semantic Kernel are only helpful for the top entities - but fail when ranking within the long tail. Another factor is most likely the fact that this dataset has only binary annotations, and is thus not as fine grained as the 1-5 Likert scale of the REWQ Robust04 ground truth. All in all, we take these results to be additional evidence for our previous findings.

Table 3: List of all REWQ Robust04 queries sorted by average ground truth scores for top 3 entities (*gt*). Also showing the NDCG@10 score (*ndcg*) and the top-3 entities as retrieved by the RankLib coordinate ascent method (using the complete set of features). Ground truth values in brackets; abbreviated entities are denoted with *.

gt	ndcg	query	top-1 entity	top-2 entity	top-3 entity
5.0	.895	schengen agreement	Schengen_Agreement (5)	Schengen_Area (5)	Schengen_Inform._Sys. (5)
5.0	.894	magnetic levitation maglev	Maglev (5)	Shanghai_Maglev_Train (5)	Transrapid (5)
4.8	.931	territorial waters dispute	UN_Law_of_Sea* (4.3)	Aegean_dispute (5)	Kuril_Islands_dispute (5)
4.7	.865	el nino	El_Nino-Southern_Oscillation (5)	Pacific_Ocean (4)	La_Nina (5)
4.3	.942	ferry sinkings	MS_Estonia (5)	MS_Herald_of_Free_Enterprise (5)	Silja_Line (3)
4.3	.927	in vitro fertilization	In_vitro_fertilisation (5)	Fertility_clinic (5)	Shan_Ratnam (3)
4.3	.879	poliomyelitis and post polio	Poliomyelitis (4.5)	Polio_vaccine (4.5)	Jonas_Salk (4)
4.3	.787	osteoporosis	Hormone_therapy* (5)	Estrogen (5)	Nutrition (3)
4.0	.803	industrial espionage	Volkswagen (4)	General_Motors (4)	Opel (4)
3.8	.863	polygamy polyandry polygyny	Al-Arqam (4.5)	Code_of_Personal_Status_(Tunisia) (4.5)	Tunisia (2.5)
3.3	.769	amazon rain forest	Amazon_rainforest (5)	Manaus (2)	Amazon_Basin (3)
3.3	.748	argentine british relations	Foreign_relations_of_Argent. (4)	Argentina_national_rugby_team* (1)	Falklands_War (5)
3.2	.751	antarctica exploration	South_Pole (3.3)	Antarctica (4)	Antarctic_ecozone (2.3)
2.8	.860	supercritical fluids	Supercritical_fluid (5)	Euler_equations_(fluid_dyn)* (2.5)	Biodegradation (1)
2.8	.760	computer viruses	Michelangelo_(cmpt_virus)* (5)	Personal_computer (2)	Computer_industry (1.5)
2.7	.836	lyme disease	Lyme_disease (5)	Centers_for_Disease_Control* (2)	Old_Lyme,_Connecticut (1)
2.7	.669	falkland petroleum exploration	Falkland_Islands (4.3)	Falklands_War (2.7)	Stanley,_Falkland_Islands (1)
2.5	.739	hydroponics	NASA (4)	Jordan (1)	Mars (2.5)
2.3	.868	killer bee attacks	Africanized_bee (3)	Ceratitis_capitata (2)	San_Diego (2)
2.3	.831	implant dentistry	Dentistry (4)	Cochlear_implant (1)	Uni_of_Med_&_Dent_NJ* (2)
2.3	.749	agent orange exposure	Agent_Orange (5)	Agent_Orange_(band) (1)	Agent_Orange_(album) (1)
2.3	.718	king hussein peace	Hussein_of_Jordan (4.5)	Abdullah_II_of_Jordan (1)	Black_Septmbr_Jordan* (1.5)
2.0	.672	counterfeiting money	Counterfeit (3)	Los_Angeles (1)	Novosibirsk (2)
2.0	.582	unsolicited faxes	Fax (4)	Personal_computer (1)	ISDN* (1)
1.9	.966	agoraphobia	Charles_M._Schulz (3.3)	Snoopy (1.3)	UGM-27_Polaris (1)

5.5 Feature analysis

To better understand the importance of the different features within our model, we study the individual ranking performance of each feature, and perform a feature ablation study.

Single features as rankers. Analyzing the individual features in isolation, Figure 3 shows the NDCG@10 performance achieved by each feature individually. We find that the mention frequency (MenFrqIdf) and the Wikipedia fulltext search (WikiSDM) both perform individually well as ranking metric for both datasets. For the REWQ-Robust dataset, WikiSDM is the highest performing feature. Since we are only re-ranking the most frequent entity mentions in high-ranked documents, the WikiSDM method is filtered by a very effective whitelist. This confirms our intuition that entities which occur often in relevant documents are themselves relevant, but also that ranking entities based on their Wikipedia article according to WikiSDM is a non-negligible indicator. All context-based query-mention-features (indicated by prefix *C_*) perform worse than their no-context counterparts (indicated by prefix *M_*), e.g. C_GloSum vs. M_GloSum, thus letting us question their value for entity ranking.

The contribution of the other query-entity features, which are based on DBpedia, namely Qent and QEntEntSim, are in between – they perform worse than the strong WikiSDM, but better compared to some of the mention-based approaches. On both dataset, QEntEntSim as single feature performs better than the QEnt feature. Since QEntEntSim is leveraging knowledge base paths and ontological types between entities in the query and the documents, these provide a meaningful way to connect otherwise missing entities.

In summary, the high performance of the MenFrIdf features highlights that the candidate generation strategy already provides a useful approach on its own: this finding holds for both datasets despite using different document retrieval and entity linking methods.

Ablation study on Robust04. To further analyze the individual features, we perform a features ablation study: For each single feature, or set of features, we remove it from the set of features available to RankLib, re-train it with the same parameters and compare its performance against the full-feature setting. Results for the Robust04 are reported in Table 4 which is sorted by relative loss caused by removing a feature (group).

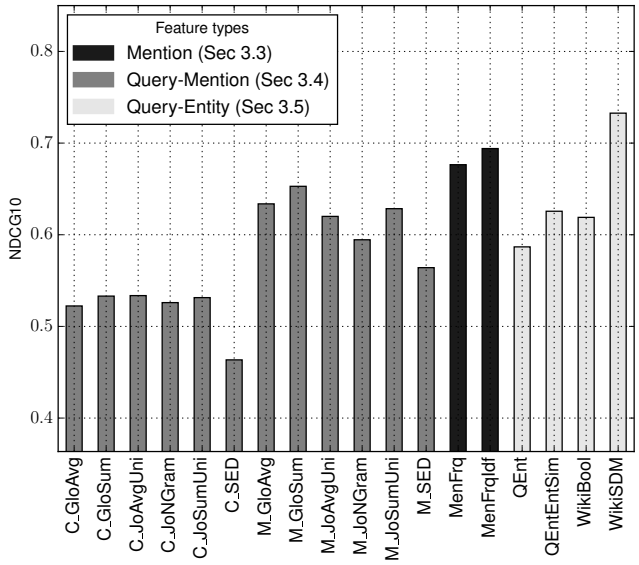
Surprisingly, we find that removing all mention-based features (i.e. SED, Jo, Glo for mention and mention context) actually improves the overall performance by 0.1% in NDCG (0.7% in NDCG@10) – however, the differences are small and not statistically significant. This finding might also results from the fact that the MenContext group combines features of different quality: While the string edit distance (SED) is helpful (-1.3% NDCG, -2.8% NDCG@10), we cannot confirm this for the JoBim text features (-0.2% and +0.3%).

The DBpedia-based features (DBpedia) seem to have a positive influence on the overall performance (-1.0% and -1.9%), even though not being statistically significant. Interestingly, removing any of the two DBpedia features QEntEntSim or Qent individually would let to a different conclusion.

The Wikipedia-based features show a strong and significant influence on the overall performance, removing them leads to a drop of -2.3% for NDCG and -5.1% for NDCG@10. The single most important feature is the mention frequency features (MenFrqIdf), thus supporting our assumption that a good initial document retrieval helps to obtain a good pool of relevant candidate entities.

Ablation study on ClueWeb12. The findings for the ClueWeb12 dataset in Table 5 confirm the findings from the Robust04 dataset. Again, leaving out all mention-based features actually improves the performances – but as above, the difference is not statistically significant. On the other end of the table, and also in line with above findings, the mention frequency is the single most important features with rather large differences between 19.7% (MAP) and 6.5% (NDCG).

(a) Robust04 Dataset, 1-5 annotations, KBBridge EL



(b) Clue12 Dataset, binary annotation, FACC1 EL

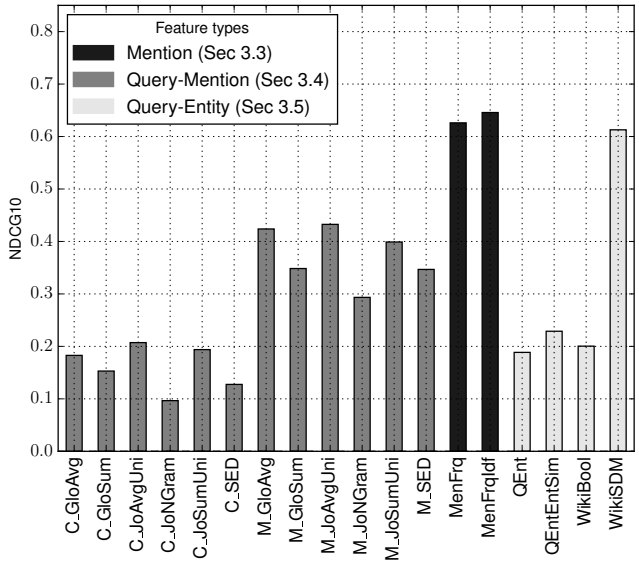


Figure 3: Feature-by-feature analysis for (a) the REWQ-Robust04 and (b) the REWQ ClueWeb12 dataset.

We can also confirm our finding that the simple SED is more effective than the Glove and Jo features. The role of the DBpedia features is slightly different, they seem to be even less helpful for the ClueWeb dataset than for the Robust04 dataset. A possible explanation is the the difference in the annotation method: The binary ClueWeb12 annotations are likely to not capture fine-grained differences between entity relevance levels, which might be expressed by knowledge-base links.

In summary, all findings for this dataset are in line with the findings for the Robust04 data, which is interesting because both datasets are rather different in nature, i.e., different ground truth labels (binary vs. 1-5 scale), different document retrieval (SDM vs. EQFE), and different entity linkers (FACC1 vs. KBBridge).

Table 4: Feature ablation study REWQ Robust04 dataset. Sorted by difference in NDCG value. P-values (p) from two-sided paired t-test ≤ 0.05 are denoted with †.

w/o	ndcg	$\Delta\%$	p	ndcg10	$\Delta\%$	p
RankLib All	0.936	-	-	0.817	-	-
MenContext	0.937	0.1	0.68	0.823	0.7	0.56
QEntEnt	0.935	-0.1	0.73	0.824	0.8	0.58
Qent	0.934	-0.2	0.58	0.825	0.9	0.44
Jo	0.934	-0.2	0.53	0.819	0.3	0.85
Context	0.933	-0.3	0.28	0.816	-0.1	0.89
Glo	0.928	-0.8	0.10	0.803	-1.7	0.26
DBpedia	0.927	-1.0	0.06	0.802	-1.9	0.21
WikiBool	0.926	-1.1	0.11	0.809	-1.0	0.56
SED	0.924	†-1.3	0.05	0.794	-2.8	0.09
WikiSdm	0.921	†-1.7	0.03	0.781	†-4.4	0.04
MenFrqIdf	0.917	†-2.1	0.04	0.774	†-5.4	0.05
Wikipedia	0.914	†-2.3	0.01	0.776	†-5.1	0.03

Table 5: Feature ablation study on REWQ ClueWeb12 dataset. Sorted by relative difference (Δ) in MAP value. P-values (p) from two-sided paired t-test ≤ 0.05 are denoted with †.

w/o	map	$\Delta\%$	p	ndcg	$\Delta\%$	p	ndcg10	$\Delta\%$	p
RankLib All	.328	-	-	.572	-	-	.711	-	-
MenContext	.333	1.4	.41	.574	0.3	.55	.714	0.5	.70
Jo	.332	1.0	.55	.573	0.2	.69	.716	0.8	.50
DBpedia	.329	0.1	.92	.572	0.0	.90	.701	-1.4	.26
QEntEnt	.327	-0.4	.48	.572	-0.1	.68	.708	-0.4	.64
Context	.326	-0.6	.49	.570	-0.3	.34	.698	-1.7	.06
Glo	.326	-0.7	.51	.571	-0.3	.46	.698	†-1.7	.05
Qent	.326	-0.8	.63	.571	-0.2	.75	.701	-1.4	.32
SED	.326	-0.8	.35	.571	-0.3	.46	.698	-1.8	.15
WikiSdm	.320	-2.6	.25	.566	-1.1	.26	.693	-2.5	.28
WikiBool	.313	†-4.6	.05	.565	-1.3	.08	.670	†-5.7	.01
Wikipedia	.303	†-7.7	.02	.556	†-2.9	.02	.650	†-8.5	.02
MenFrqIdf	.264	†-19.7	.00	.535	†-6.5	.01	.630	†-11.4	.03

6. RELATED WORK

In recent years, there has been a growing interest in research aimed at developing methods to search and rank entities on the web. Starting with the Initiative for the Evaluation of XML Retrieval (INEX) [11] and TREC initiatives [2], the task of entity ranking has gained momentum, and a variety of different approaches have been developed, ranging from natural language interface for XML retrieval and Question Answering over Linked Data ([15, 39, 43], *inter alia*) all the way through open-domain, document-based entity search [13].

Evaluation exercises on entity ranking systems within the INEX evaluation forum have taken place since 2006. Most tracks build upon a definition of entities based on Wikipedia, i.e., every Wikipedia article is a valid entity, which we also use. However, alternative or broader definitions are possible (see, e.g., [2, 13]). The focus on type queries led to the development of systems that leveraged Wikipedia category system – e.g., category overlap [32], or measures capturing the strength of association between terms and category labels [27]. Broadly similar in spirit to our WikiSDM approach, additional structured content from Wikipedia such as links can be also used to perform document retrieval [32, 10]. Wikipedia entities and their categories, in turn, can be used to improve INEX-style entity retrieval [26]. In our work, we also use Wikipedia to define our vocabulary of entities and develop models to rank these entities for keyword queries. However, we also aim to expand the pool of features from a fully-structured knowledge source, and accord-

ingly use facts from DBpedia. Our results indicate that, similarly to the case of a wide range of NLP and IR applications [22] further gains can be obtained by complementing knowledge from text with that from structured knowledge bases like Wikipedia and DBpedia. Knowledge of this kind has been recently exploited also within a feature-based approach to INEX entity ranking [35] and a hybrid approach that combines IR and structured search for Ad-hoc Object Retrieval (AOR) [38].

One of the methods closest to ours is the work of Kaptein and Kamps [26], who also address the problem of retrieving entities for web queries. They assume the availability of type information, either as an explicit category (given by the user) or a latent one that needs to be extracted from the query. Their method accordingly performs a latent estimation of the category, which in a second step is used to select and rank the entities before presentation to the user. In contrast, our goal is to retrieve entities for any (reasonable) web query, including those not posed with the goal of retrieving entities. That is, we crucially depart from the type-centric tasks formulated in the context of INEX and AOR, and embrace a notion of entity relevance which is based on whether a human would talk about this entity when explaining the topic.

The SemSets model [6] also addresses the problem of ranking entities w.r.t. a given keyword query. This approach focuses on type queries, following the entity query classification schema from Pound et al. [34]: in contrast, in this work we look primarily at web queries, which can also, and do in fact often cover one of the "remaining" query types, i.e. untyped relations and keyword queries. Arguably, entity retrieval for type queries has to put more emphasis on the correct interpretation of the query, since the answer crucially requires the intended target type to be correctly recognized. In contrast, we do not limit our approach to queries of specific types, and accordingly leverage instead type-agnostic methods such as, for instance, our DBpedia graph exploration method. Perhaps interestingly, Ciglan et al. mention in their work that for a type query, a "human user would probably enter such a query to a web search engine and inspect several top-k results and [...] search the text of the inspected documents to find the desired set of entities" [6, p. 131]. This is exactly the pipeline we created in this work to solve our entity ranking problem.

The question of ranking entities from documents, which we implicitly touch by ranking entities from the search result documents, is addressed in recent work from Dunietz and Gillick, who define the task of "entity salience [as] assigning a relevance score to each entity in a document" [13]. However, even though our method could also be used for entity ranking at the document level, we take here a query-centric view of the problem. That is, we rank entities from the whole collection of retrieved documents because we aim at ranking entities w.r.t. the query, and not w.r.t. the individual documents. In doing this, we follow the 'traditional' INEX-style Wikipedia-based definition of entities. This makes it possible for us to leverage a Wikipedia-based resource such as DBpedia in a straightforward way, and evaluate the contribution of a wide-coverage knowledge base for our problem. An 'open' definition of entities, like the one found in [13], instead, opens up new problems such as how to detect and include new entities into the background knowledge base: although recently there have been attempts to address this problem at web scale [20], we leave this issue for future work.

7. CONCLUSIONS

In this work, we addressed the problem of ranking entities for open-domain web queries, starting with the query-relevant documents retrieved by a state of-the-art document retrieval system. We investigated the performance of a variety of heterogeneous features, which were combined by two learning-to-rank methods. Our results indicate that query-relevant documents with entity links provide a complementary source of information to direct knowledge base (e.g., Wikipedia) retrieval, yielding an NDCG@10 score on Robust04 of over 0.82, compared to 0.73 for Wikipedia retrieval. Together with the frequency of entity links within the retrieved documents (0.68 NDCG@10), Wikipedia retrieval is one of the strongest individual features. For most other results of the ablation study we cannot find significant differences. For example, we cannot find a unique significant contribution of features based on distributional similarity for this task (JoBimText and GloVE). Likewise, incorporating relations between entities does not yield a measurable benefit (QEntEnt and Semantic Kernel). This finding is in spite of the benefit of directly matched query entities (QEnt), which is another strong signal. Nevertheless, combining all these signals together within a supervised learning framework is able to yield statistically significant improvements over ranking by single features, so as to yield competitive NDCG scores.

Our method relies on the performance of the underlying document retrieval and entity linking systems. While error analysis revealed that our ranking suffers from systematic errors from these two components, our NDCG scores on both datasets – which use different document retrieval and entity linking systems – indicate that our supervised approach is nevertheless able to cope with the noise in the input data.

We view our work as a first step that opens up new opportunities for the development of advanced entity-rich search engines. This is because, arguably, humans tend to think in terms of entities, like persons, places, or organizations when it comes to complex topics. Consequently, entity ranking for Web queries could play a key role in many high-end applications, including entity-centric search [9], topic page generation [1], as well as new approaches to browsable search interfaces [12] that aim at including relevant documents, entities and ontological types, together with textual evidence for how entities are connected to the query, in the search results.

Acknowledgments

This work was partially funded by the Deutsche Forschungsgemeinschaft within the JOIN-T project (research grant PO 1900/1-1). Part of the computational resources used for this work were provided by an Amazon AWS in Education Grant award and the bwUniCluster (funded by the bwHPC program of the state of Baden-Württemberg, Germany). We thank Christian Meilicke for his contributions and the anonymous reviewers for their helpful comments.

Downloads

In order to encourage further research on the topic of retrieving entities for open-domain queries, we make our gold standard annotation for the Robust04 and the TREC Web (ClueWeb12) datasets freely available at `http://rewq.dwslab.de` under a Creative Commons (by-nc-sa, 4.0) license.

8. REFERENCES

[1] N. Balasubramanian and S. Cucerzan. Beyond ranked lists in web search: Aggregating web content into topic pages. *International Journal of Semantic Computing*, 4(4):509–534, 2010.

[2] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *Proc. of TREC-09*, 2010.

[3] C. Biemann and M. Riedl. Text: Now in 2D! A framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1:55–95, 2013.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia – A Crystallization Point for the Web of Data. *Journal of Web Semantics*, 7(3), 2009.

[5] S. Bloehdorn, R. Basili, M. Cammisa, and A. Moschitti. Semantic kernels for text classification based on topological measures of feature similarity. In *Proc. of ICDM'06*, pages 808–812, 2006.

[6] M. Ciglan, K. Nørvåg, and L. Hluchý. The SemSets model for ad-hoc semantic list search. In *Proc. of WWW'12*, pages 131–140, 2012.

[7] J. Dalton and L. Dietz. A neighborhood relevance model for entity linking. In *Proc. of OAIR-13*, pages 149–156, 2013.

[8] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proc. of SIGIR-14*, pages 365–374, 2014.

[9] N. Dalvi, R. Kumar, B. Pang, R. Ramakrishnan, A. Tomkins, P. Bohannon, S. Keerthi, and S. Merugu. A web of concepts. In *Proc. of PODS '09*, pages 1–12, 2009.

[10] G. Demartini, C. S. Firan, T. Iofciu, R. Krestel, and W. Nejdl. Why finding entities in Wikipedia is difficult, sometimes. *Information Retrieval*, 13(5):534–567, 2010.

[11] G. Demartini, T. Iofciu, and A. P. de Vries. Overview of the INEX 2009 entity ranking track. In *Proc. of INEX*, pages 254–264, 2009.

[12] L. Dietz, M. Schuhmacher, and S. Ponzetto. Queripidia: Query-specific Wikipedia construction. In *Proc. of AKBC-14*, 2014.

[13] J. Dunietz and D. Gillick. A new entity salience task with millions of training examples. In *Proc. of EACL-14*, pages 205–209, 2014.

[14] O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-based information retrieval using Explicit Semantic Analysis. *ACM Transactions on Information Systems*, 29(2):8:1–8:34, 2011.

[15] S. Elbassuoni, M. Ramanath, R. Schenkel, M. Sydow, and G. Weikum. Language-model-based ranking for queries on RDF-graphs. In *Proc. of CIKM-09*, pages 977–986, 2009.

[16] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with Wikipedia pages. *IEEE Software*, 29(1):70–75, 2012.

[17] E. Gabrilovich, M. Ringgaard, and A. Subramanya. Facc1: Freebase annotation of ClueWeb corpora, version 1, 2013.

[18] R. Gupta, A. Halevy, X. Wang, S. Whang, and F. Wu. Biperpedia: An ontology for search applications. In *Proc. of PVLDB-14*, pages 505–516, 2014.

[19] S. Gurajada, J. Kamps, A. Mishra, R. Schenkel, M. Theobald, and Q. Wang. Overview of the INEX 2013 linked data track. In *Working Notes for CLEF 2013*, 2013.

[20] J. Hoffart, Y. Altun, and G. Weikum. Discovering emerging entities with ambiguous names. In *Proc. of WWW-14*, pages 385–396, 2014.

[21] J. Hoffart, D. Milchevski, and G. Weikum. STICS: Searching with Strings, Things, and Cats. In *Proc. of SIGIR-14*, pages 1247–1248, 2014.

[22] E. Hovy, R. Navigli, and S. P. Ponzetto. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27, 2013.

[23] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proc. of SIGIR-00*, pages 41–48, 2000.

[24] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of SIGKDD-02*, pages 133–142, 2002.

[25] T. Joachims. Training linear SVMs in linear time. In *Proc. of SIGKDD-06*, pages 217–226, 2006.

[26] R. Kaptein and J. Kamps. Exploiting the category structure of Wikipedia for entity ranking. *Artificial Intelligence*, 194:111–129, 2013.

[27] R. Kaptein, P. Serdyukov, A. P. de Vries, and J. Kamps. Entity ranking using Wikipedia as a pivot. In *Proc. of CIKM-10*, pages 69–78, 2010.

[28] V. I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, 1:8–17, 1965.

[29] T.-Y. Liu. *Learning to rank for information retrieval*. Springer-Verlag, Berlin, 2011.

[30] D. Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[31] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc. of SIGIR-05*, pages 472–479, 2005.

[32] J. Pehcevski, A.-M. Vercoustre, and J. A. Thom. Exploiting locality of Wikipedia links in entity ranking. In *Proc. of ECIR-08*, pages 258–269, 2008.

[33] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *Proc. of EMNLP-2014*, pages 1532–1543, 2014.

[34] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proc. of WWW-10*, pages 771–780, 2010.

[35] H. Raviv, D. Carmel, and O. Kurland. A ranking framework for entity oriented search using markov random fields. In *Proc. of JIWES '12*, pages 1–6, 2012.

[36] M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. In *Proc. of WSDM-14*, pages 543–552, 2014.

[37] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383, 2011.

[38] A. Tonon, G. Demartini, and P. Cudré-Mauroux. Combining inverted indices and structured search for ad-hoc object retrieval. In *Proc. of SIGIR-12*, pages 125–134. ACM, 2012.

[39] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over RDF data. In *Proc. of WWW-12*, pages 639–648, 2012.

[40] E. M. Voorhees. The TREC robust retrieval track. In *ACM SIGIR Forum*, volume 39, pages 11–20, 2005.

[41] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, 2005.

[42] W. Wu, H. Li, H. Wang, and K. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proc. of SIGMOD-12*, pages 481–492, 2012.

[43] N. Zhiltsov and E. Agichtein. Improving entity search over linked data by modeling latent semantics. In *Proc. of CIKM-13*, pages 1253–1256, 2013.