Finding Images of Difficult Entities in the Long Tail

Bilyana Taneva
Max-Planck Institute for Informatics
Saarbrücken, Germany
btaneva@mpi-inf.mpg.de

Mouna Kacimi
Free University of Bozen-Bolzano
Italy
Mouna.Kacimi@unibz.it

Gerhard Weikum
Max-Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

While images of famous people and places are abundant on the Internet, they are much harder to retrieve for less popular entities such as notable computer scientists or regionally interesting churches. Querying the entity names in image search engines yields large candidate lists, but they often have low precision and unsatisfactory recall. In this paper, we propose a principled model for finding images of rare or ambiguous named entities. We propose a set of efficient, light-weight algorithms for identifying entity-specific keyphrases from a given textual description of the entity, which we then use to score candidate images based on the matches of keyphrases in the underlying Web pages. Our experiments show the high precision-recall quality of our approach.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

1. INTRODUCTION

1.1 Motivation

The digital information world is getting more and more organized. Knowledge bases such as DBpedia [1] or Freebase (`freebase.com`) systematically organize millions of entities and billions of facts into a formal representation based on the RDF data model. However, despite these advances in moving from raw data to value-added knowledge, there are still major shortcomings in organizing multimedia information such as images of named entities. For example, out of the 735 articles in the Wikipedia category *2010 FIFA World Cup players*, many articles do not have an image of the football (soccer) player. The same problems hold for scientists, artists, and politicians in the long tail of entities. Even if Wikipedia contains a picture, users may be interested in obtaining a wide variety of pictures at different occasions or different ages. Likewise, for geographic or cultural landmarks (mountains, temples, etc.), users may want to see different perspectives, weather/light conditions, etc.

It is often a tedious task to find good images using search engines. Even when the top-20 results contain some true matches, the user may have to look at the actual Web pages to figure out which image shows which entity (unless the user was already familiar with the requested person, waterfall, cathedral, etc.). Ideally, we would like to have a knowledge base, perhaps as an extension of Wikipedia or DBpedia, that contains a wide variety of different pictures for all named entities. This collection should be automatically constructed and maintained as new images appear on the Web. This paper addresses this very topic. While projects like `image-net.org` are collecting large amounts of images for general concepts (e.g., sunsets, cats, kiwis), there is no counterpart for individual entities (e.g., the Bridge of Sighs in Venice, as opposed to any kind of bridge).

The outlined endeavor is challenging for the following reasons. Names can be highly ambiguous, and search engines do not always favor the interpretation that the user is interested in. For example, assume you want to find pictures of the economist David Gale. Searching with "David Gale" yields results that are dominated by the actor Kevin Spacey who acted in the movie "The Life of David Gale" (totally unrelated to the economist). Entities in the long tail may be rare on the Web, despite being well worthy of inclusion in a universal knowledge base. For example, the top-100 search results for Carsten Lund, who has received the Goedel prize (the most prestigious award for theoretical computer science), contain only few correct images at low ranks. Unfortunately, the names themselves do not give any cues if an entity is a difficult case in terms of rarity or ambiguity.

A practically viable solution should be able to operate with minimum human supervision. The main prior work on this problem [16] highly depends on explicitly labeled training samples for each class of entities, and performs computationally expensive query expansions and aggregation steps.

Our goal is not just finding one image on the first result page (ideally rank 1), but to find as many (ideally different) images of the entity on high ranks. Thus we aim at a high value of the area under the precision-recall curve.

1.2 Contribution

Our approach of finding images for ambiguous or rare named entities is illustrated in Figure 1. It consists of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.
Copyright 2011 ACM 978-1-4503-0717-8/11/10 ...$10.00.

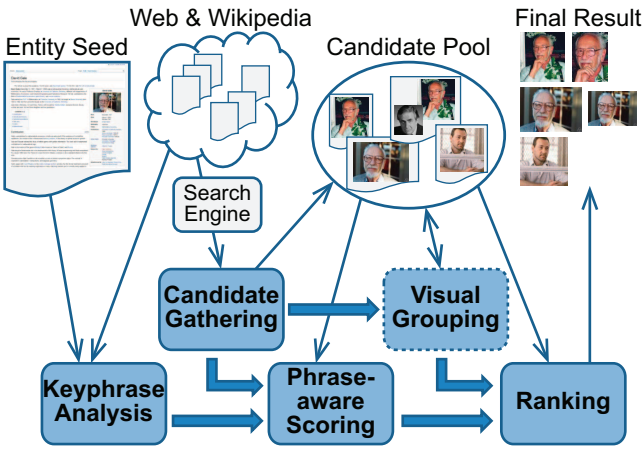

Figure 1: System Architecture. Rectangles are system components. *Visual grouping* is optional.

following steps. For a given entity of interest, we start from a salient *seed page* (or ask the user for it, find it in a knowledge base, etc.). This could be the Wikipedia article for the entity, but we can handle arbitrary seed pages on the Web such as people's home pages or short descriptions. The only requirement is that the user herself can uniquely identify the entity from solely seeing the seed page. If there is no other information about the entity but its name, the task becomes ill-defined for the machine and the only possible output can be a mixture of results for different entities with the same name. The *keyphrase analysis* component then automatically extracts from the seed page a ranked list of *keyphrases* that are characteristic for the entity. The *candidate gathering* component sends a keyword query using only the entity name to image search engines and obtains a pool of candidate images fetched with their underlying Web pages. Then the *phrase-aware scoring* component uses a new model for *re-ranking* the results in the candidate pool, based on the entity-characteristic keyphrases found earlier. For each image in the pool it identifies *full or partial matches* of keyphrases in the Web page containing the image, and computes a new form of relevance score used for re-ranking. Optionally, the *visual grouping* component groups visually similar images aiming at a diversified final list of results.

The paper makes the following novel contributions: 1) a principled model for re-ranking of images for rare or ambiguous named entities in the long tail; 2) a phrase-aware scoring model for image candidates based on partial keyphrase matches in an image's underlying Web page; 3) a robustness test for entity difficulty that allows us to selectively apply our ranking model only when it is likely to improve the result list; 4) a comprehensive experimental evaluation with a variety of entity categories, demonstrating the high precision-recall quality of our approach, and the improvements over various baseline methods including the original image-search result list and a language-model-based ranking method that directly uses the seed page of an entity.

2. RELATED WORK

A number of recent projects have aimed at enhancing the *semantic organization of image collections*. Prominent examples are TinyImage [19] and LabelMe [15]. TinyImage is a dataset of low resolution images collected from the Internet by sending all nouns in WordNet [4] as queries to several image search engines. It uses the hypernymy relation of WordNet in conjunction with nearest-neighbor methods to automatically classify the retrieved images. LabelMe is a large collection of images with ground truth labels to be used for object detection and recognition research. It aims at object class recognition (e.g., bridge) as opposed to instance recognition (e.g., Golden Gate Bridge).

A few projects tackle the more specific problem of *integrating images into knowledge bases* [3, 16]. ImageNet [3] builds a large-scale labeled image collection based on the taxonomic hierarchy of WordNet. To this end, it exploits the hypernymy relation between entity classes and nearest-neighbor-based classification with visual features. While ImageNet focuses on finding images of semantic classes such as towers, churches, etc., our work addresses photos of *individual entities* like the Blue Mosque in Istanbul. Closest to our paper is the work of [16], which aims to populate a knowledge base of individual entities with their images. The latter harnesses relational facts about entities for generating expanded queries posed to image search engines. The approach retrieves all result lists from the generated expanded queries, merges the lists, and ranks the individual images by weighted voting procedure. Weights are dependent on the type of entity (e.g., scientist vs. politician) and computed from training entities for each type. This approach achieved very good experimental results but had significant limitations: dependence on ontological facts which are not always available, the need for training samples for each entity type which is a bottleneck, and the high overhead caused by query expansions resulting in a large number of search-engine requests. In our work, we propose a very different, more lightweight technique that overcomes these limitations.

Keyphrase extraction is one of the components in our system. There are both supervised [5, 8] and unsupervised [7, 11] approaches. Supervised methods crucially depend on the availability of manually labeled training data, while unsupervised methods do not need labeled samples and are domain-independent. They typically use IR measures like tf-idf, consider linguistic features, and harness document structure such as XML tags. We adopt an unsupervised approach for keyphrase extraction to avoid training bottlenecks and for domain-independence. We use noun phrases ranked by Mutual Information, as described in Section 3.

3. KEYPHRASE MINING & WEIGHTING

Finding good images of entities is not always straightforward, especially when the user is not familiar with the (look of the) requested entity. Given a list of image results, the user sometimes has to look at the Web pages that contain the image results to figure out which image shows which entity. To automate this challenging task, we exploit characteristic phrases of entities to select good matches of images from the result pool that we obtain from querying image search engines with entity names.

For a given entity, we start from a salient *seed page*. We assume that the page has enough information so that a human user can uniquely identify the entity and there is no confusion about other entities with the same name. We then automatically extract from the seed page a *ranked list of keyphrases* that are characteristic for the entity. These keyphrases are later used to re-rank images.

Keyphrase extraction. On first thought, a good method for extracting keyphrases would be to identify all noun phrases in the seed page. We use the OpenNLP tool [12] for this purpose. For example, from the seed page of the economist David Gale (`en.wikipedia.org/wiki/David_Gale`), we gather phrases like "economist", "Professor Emeritus", "partner Sandra Gilbert", "poet", "daughters", etc. Some of them are characteristic for the entity, but others dilute the focus by being either too broad or misleading (e.g., the phrase "poet" actually refers to Gale's partner). To overcome these issues while keeping the approach computationally efficient (e.g., avoiding deep natural-language parsing), we introduce a notion of *focused keyphrases* that are truly characteristic for an entity. For David Gale, we prefer phrases like "University of California, Berkeley", "economist", "game theory", etc.

Depending on whether the entity seed page is a Wikipedia article or an arbitrary Web page, we use two different strategies to select focused keyphrases. Given a Wikipedia seed page, we extract from the article's text part all outgoing links that point to other Wikipedia articles. Then, we select the anchor text of these links as focused keyphrases. We use the WikiPrep tool [6] for this purpose. For an arbitrary Web page, we select all noun phrases that are titles of Wikipedia articles, including redirects. This way, we restrict the vocabulary of keyphrases to named entities and informative nouns.

Keyphrase weigthing. For each selected keyphrase of a given entity, we also compute and assign a weight, which measures how well the keyphrase characterizes the entity. We use the standard *Mutual Information* measure (*MI*) for this purpose, but other measures can be applied as well. The *MI* of a given keyphrase and an entity indicates how much information the keyphrase contains about the entity. The higher the *MI* is, the more dependent they are. More formally, for each entity we have two possible classes of pages: one for pages about the entity (c), and one for other pages ($\overline{c}$). The *MI* of a keyphrase and an entity is then given by:

$$MI(X;Y) = \sum_{x_k \in \{1,0\}} \sum_{y_c \in \{1,0\}} P_{XY}(x_k, y_c) \log_2 \frac{P_{XY}(x_k, y_c)}{P_X(x_k) P_Y(y_c)}$$

where X is a random variable that takes values 1 if the page contains the keyphrase and 0 otherwise, and Y is a random variable that takes values 1 if the page is in class c and 0 if the page is in class $\overline{c}$. In our implementation we typically have one seed page per entity. Thus, the class c contains only this page, and all other pages in the corpus (e.g., all other Wikipedia articles) belong to class $\overline{c}$.

Note that keyphrases often consist of multiple words. We compute the *MI* weight for the entire keyphrase and also for each of its constituent words. The usage of the weights of individual words is described in Section 4.

Our model can also be specialized to use individual words only, for example, all words that constitute the keyphrases of an entity. In this special case, referred to as the *words-aware model* (as opposed to *phrase-aware model*), words lose their phrase context but can still be good cues for an entity, especially with our weighting method. For example, David Gale would be characterized by single words like "economist", "university", "Berkeley", "game", etc.

4. PHRASE-AWARE SCORING

Assume that for an entity of interest e we are given its candidate pool of images with their underlying Web pages, and its set of characteristic keyphrases ranked by *MI* as described in Section 3. We denote the keyphrases of e by $k_1(e), \ldots, k_m(e)$, or $k_1, \ldots, k_m$ if the entity is given by the context. Then for each image/page p in the candidate pool we compute a *phrase-aware score* $s(p)$, which is later used to rank the images in the pool:

$$s(p) = \sum_{i=1}^{m} w(k_i)\ \mathcal{S}(k_i, p)$$

Here $w(k_i)$ is the *MI* weight of keyphrase k_i and by $\mathcal{S}(k_i, p)$ we denote a *keyphrase score* for phrase k_i and image/page p. The keyphrase score $\mathcal{S}(k_i, p)$ is estimated by identifying matches or partial matches of a phrase k_i in a page p. Note that in the special case of the words-aware model, $\mathcal{S}(k_i, p)$ is either 0 or 1, as a single word is either in the page or not.

The best image pages for a given entity would ideally match exactly the entity's keyphrases. However, partial matches of keyphrases can still be good cues for the entity. For example, if "University of California, Berkeley" is a keyphrase, we are still interested in pages that contain pieces and variants such as "Berkeley University" or "UC Berkeley". In such cases, a good image page should contain as many of the keyphrase words as possible within close distance.

We compute keyphrase scores using a *Minimum Cover model*. We also explored two further alternative models, Büttcher's and Spans scoring models, but they achieved slightly inferior results to the minimum-cover-based model (see [17] for more details). All these models are extensions of prior work on proximity-aware scoring. The original models aimed at enhancing the scoring for standard keyword search by considering the proximity of the query keywords in a result candidate. In contrast, we apply and adapt these kinds of models to entity-specific keyphrases, not queries.

Scoring based on Minimum Cover. The Minimum Cover [18, 2] of a set of words in a text sequence is defined as the length of the shortest subsequence that contains all words at least once. We introduce an extension of this model to compute the keyphrase score for given entity keyphrase k and image page p:

$$\mathcal{S}(k, p) = \frac{|k \cap p|}{mincover(k \cap p, p)} \left(\frac{\sum_{t \in k \cap p} w(t)}{\sum_{t \in k} w(t)} \right)^{\lambda}$$

Here $k \cap p$ denotes the set of words from a keyphrase k that are matched in page p, and $mincover(k \cap p, p)$ returns the length of the shortest text segment of p where all words in $k \cap p$ appear at least once. We use the reciprocal of $mincover(k \cap p, p)$ to obtain high scores for short text segments and low scores for long segments. To capture how many keyphrase words are reflected by the *mincover* score, we multiply the reciprocal of the *mincover* by the number of matched keyphrase words $|k \cap p|$. In this way, we distinguish pages with comparable *mincover* scores but with different number of matched keyphrase words. The first factor in the formula ranges from 0 to 1. It is equal to 1 if there is an exact match of the words in $k \cap p$ in p, and to 0 if $|k \cap p| = 0$.

The original Minimum Cover model of [18, 2] for improved result ranking of standard text queries would consider only the first factor in the formula (with adaptation to its re-

spective setting). However, this would still favor pages with fewer matched keyphrase words. For example, consider a keyphrase k with 5 words, and two pages p and q. Assume, $|k \cap p| = 2$ and $mincover(k \cap p, p) = 2$, and $|k \cap q| = 4$ and $mincover(k \cap q, q) = 4$. In this case, both p and q would have score 1 for the first factor in the formula, even though they match different number of keyphrase words. To solve this inconsistency, we introduce the second factor of the formula. It captures how many keyphrase words are missing from the page and how characteristic they are for the keyphrase. This is expressed by the weighted fraction of keyphrase words that appear in the page, where words are weighted by *MI* (see Section 3). This factor ranges from 0 to 1. It is equal to 1 if $|k \cap p| = |k|$, and to 0 if $|k \cap p| = 0$.

We adjust the influence of the two factors in the formula using a parameter λ. To favor pages containing more phrase words with relatively low *mincover*, we set $\lambda > 1$ (e.g., 2).

5. ENTITY DIFFICULTY

For some entities the image search engines perform already very good, with perfect precision for the first result page. In such cases we want to keep the original ranking of images and should not apply our re-ranking model. For deciding whether to re-rank the search engine's results or not, we perform a *robustness test* for *entity difficulty*.

The robustness test uses the top-15 results retrieved from image search engines by querying with the entity name only. We cluster the set of Web pages that contain the image results using a simple density-based method, which produces a variable number of clusters depending on a threshold for intra-cluster similarity. If an entity's results produce many clusters (e.g., ≥ 4), we conclude that the entity is difficult (i.e., ambiguous, rare, or both). Only then we apply our re-ranking; otherwise the entity is considered easy and we keep the original ranking.

The clustering method processes the pages in their original ranking order. For each page we find its first sufficiently similar neighbor from the already processed pages. If such a page exists, we assign the current page to the cluster of that previous page; otherwise we create a new cluster. As a similarity measure between two pages, we use the cosine similarity of their *tf-idf* vector representations. The *tf* value of a given word is based on the frequency of the word in the page, and the *idf* value – on the full Wikipedia corpus.

6. VISUAL GROUPING OF IMAGES

In addition to exploring the text of the pages containing the candidate images for the entity-specific keyphrases, we can optionally consider the visual content of the images. Our approach groups visually similar images and assigns to each group a representative image. Having already assigned to each candidate image in the pool its phrase-aware score (see Section 4), every representative image of a visual group is given a *group score* by summing over all phrase-aware scores of the images in its visual group. Finally, in the result list of images only the representatives are included, ranked by their group scores. This way we obtain visually diverse images.

To consider the visual content of the images we use visual features like *SIFT* feature descriptors [9] and *MPEG-7* features [10]. For testing pairwise similarity between images we applied similar techniques as in [16].

	Keyphrases
Entity: Peter Naur Category: Turing Award laureates	1) Backus-Naur form 2) ALGOL 60 3) ACM A.M. Turing Award 4) Niels Bohr Institute 5) Regnecentralen
Entity: Wapta Falls Category: Waterfalls of British Columbia	1) BC Geographical Names Information System 2) Yoho National Park 3) Kicking Horse River 4) waterfall 5) British Columbia

Table 1: Keyphrases from Wikipedia seed pages.

7. EXPERIMENTS

7.1 Setup

Methodology. We evaluated our method using entity collections such as waterfalls or Turing award winners. We focused on difficult entities in the long tail, and did not consider prominent entities such as "Niagara Falls". To decide whether an entity is difficult or not, we used our robustness test for entity difficulty presented in Section 5. We also disregarded extreme cases like "Basalt Falls" (located in BC, Canada), for which we could not find a single good result in the top-50 results returned by image search engines.

For each entity we used its seed page to extract (focused) keyphrases, for which we computed *MI* measures, as described in Section 3. Table 1 shows two entities and their best focused keyphrases ranked by *MI*. To collect a candidate pool of images for each test entity, we posed a query with the entity name to `images.google.com` and retrieved the top-50 image results and their underlying Web pages.

We manually assessed the candidate pictures for each test entity by assigning one of three possible labels: relevant, not relevant, undefined. The last label was assigned to pictures, for which we could not decide whether they are relevant or not (e.g., if a person was possibly shown in a group, but the photo quality was too poor to truly tell). The undefined results were not considered in our experiments.

We performed two types of experiments: one based on Wikipedia seed pages and one based on seed pages which were not Wikipedia articles, but simple Web pages varying in text length and quality of entity description.

Test data. Our test data is based on Wikipedia categories of named entities. We used 2 Wikipedia categories with specific themes, which we perceived as typical for the long tail of entities: "Turing Award laureates" with 56 entities, out of which 34 are difficult, as concluded by the test for entity difficulty, and "Waterfalls of British Columbia" with 20 entities, 14 of which are difficult. We also used 2 entity lists with broader but heterogeneous themes: "Economists" with 589 entities and "Ruins" with 788 entities. We completely assessed the image results for all entities in the first two categories. For the other two categories we randomly sampled 25 entities from each, excluding extremely prominent entities with perfect precision for the first result page. We applied the entity difficulty test on the two samples of 25 entities: there were 23 difficult entities from the "Economists" category and 17 from the "Ruins" category.

Methods under comparison. We compared five methods: 1) our new ranking method based on the minimum-

cover matching of (focused) keyphrases (**Phr**); 2) the words-aware model as a special case of our method (**Word**); 3) the original search engine, as a main baseline (**G**); 4) the original search engine with query expansion, by including the highest-*MI* keyphrase in the entity query (**GQE**); 5) a language-model-based ranking, using the Kullback-Leibler divergence $KL(LM(e)|LM(p))$ between a result page p and the entity seed page e (in the role of a query), with Dirichlet smoothing for p using the entire Wikipedia as a background corpus. This baseline represents state-of-the-art IR methods for document retrieval [13, 20] (**KL**).

Quality measures. We used four quality measures: Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Precision at k (P@k), and Mean Reciprocal Rank (MRR). Our main measures of interest are MAP and NDCG, as we are interested in the entire precision-recall curve. We include P@k and MRR for completeness. We compute MAP and NDCG similarly to [14] and refer to [17] for more details.

7.2 Results

Ranking based on Wikipedia seed pages. The results for the ranking models using Wikipedia seed pages are shown in Table 2. The phrase-aware model almost always improves all measures in comparison to the original search engine and the search engine with query expansion. The original search engine is better than the phrase-aware model only in terms of our secondary measures P@k and MRR for "Turing Award winners" and "Waterfalls of BC" respectfully. The gains of the phrase-based model depend on the category with highest gains on the "Ruins" category.

The words-aware model and the KL-divergence-based model perform amazingly well. They perform worse than the search engine baseline for the waterfalls category, but outperform the baseline on all other categories. The phrase-aware model almost always outperforms the words-aware and the KL-divergence-based models. The exception is the "Economists" category, for which the KL-divergence model is slightly better than the phrase-based model in terms of NDCG and MRR.

Another observation is that the search engine with query expansion performs worse than the original search engine. The reason is that the highest-*MI* keyphrase used for query expansion is often too long or too specific and hence dilutes the results of the expanded query.

The results in Table 2 are obtained by using only focused keyphrases for our phrase-aware model. We also compared the use of focused keyphrases versus using all noun phrases. The results show that the first are essential for the good performance of our model [17].

Ranking with visual grouping of images. Table 3 compares the re-ranking models when we group visually similar images, using the technique of Section 6. For consistency, we apply visual grouping to Google's ranking as well: starting from the top ranks of Google's list, whenever we meet a result that is visually similar to a result higher in the ranking, we remove the lower-ranked one. As a consequence of the visual grouping, the search engine's results are slightly better than the same results without grouping.

The phrase-aware model always improves MAP and NDCG compared to the search engine baseline. The words-aware and the KL-divergence-based models are also better than the baseline. They perform very well in this setting, but still lose against the phrase-aware model in most cases.

		Phr	Word	KL	G	GQE
T	MAP@50	0.599	0.591	0.591	0.587	0.344
	NDCG@50	0.931	0.924	0.926	0.885	0.893
	P@10	0.759	0.759	0.756	0.770	0.638
	MRR	0.956	0.941	0.944	0.897	0.910
W	MAP@50	0.618	0.593	0.589	0.588	0.210
	NDCG@50	0.894	0.882	0.876	0.883	0.682
	P@10	0.714	0.714	0.671	0.700	0.378
	MRR	0.886	0.889	0.848	0.964	0.611
E	MAP@50	0.628	0.621	0.625	0.572	0.163
	NDCG@50	0.895	0.887	0.897	0.855	0.664
	P@10	0.678	0.674	0.656	0.569	0.291
	MRR	0.935	0.917	0.946	0.935	0.625
R	MAP@50	0.594	0.578	0.552	0.499	0.259
	NDCG@50	0.934	0.924	0.909	0.823	0.742
	P@10	0.765	0.747	0.723	0.635	0.447
	MRR	0.970	1.000	0.970	0.779	0.778

Table 2: Evaluation for Turing Award winners (T), Waterfalls of BC (W), Economists (E), and Ruins (R), with Wikipedia seed pages.

Ranking based on non-Wikipedia seed pages. For all 4 entity categories, we also performed experiments using non-Wikipedia seed pages, obtained from the "wild Web". For each category we chose the five entities that performed worst in terms of MAP and NDCG of the Wikipedia-based experiment. This experiment was meant as a stress-test, geared towards the most difficult entities. Seed pages for the waterfalls or some of the ruins were typically very sparse, containing only a short paragraph. Seed pages for economists or Turing award winners were almost the opposite: very detailed but fairly verbose and thus very noisy.

As keyphrases, we extracted from the non-Wikipedia seed pages all noun phrases that are titles of Wikipedia articles, but did not use phrases with *MI* below some noise threshold. The results are shown in Table 4. For these very difficult entities, we observe that the phrase-aware model outperforms the search engine baseline and the KL-divergence-based model by a large margin. The words-aware model performs comparably to the phrase-based model, as, in these cases, many keyphrases were merely one-word phrases.

7.3 Discussion

Comparing the three main competitors – phrase-based model, words-aware model, and KL-divergence-based model – to the search engine baseline, we observe the following major trends. All three methods perform better than the search engine. The phrase-based method is almost never outperformed by the search engine, whereas the other two models are sometimes inferior to the baseline. The words-aware and KL-divergence-based models sometimes slightly outperform the phrase-based model, but the gains are statistically insignificant. Conversely, the gains of the phrase-based model over the KL-divergence-based one are statistically significant; they are most pronounced for the entities with Wikipedia seed pages from the "Ruins" and "Waterfalls" categories (see Table 2) and the most difficult entities

		Phr	Word	KL	G	GQE
T	MAP@50	0.643	0.639	0.615	0.604	0.422
	NDCG@50	0.928	0.926	0.902	0.873	0.891
W	MAP@50	0.647	0.643	0.610	0.625	0.208
	NDCG@50	0.889	0.888	0.857	0.878	0.675
E	MAP@50	0.632	0.649	0.636	0.612	0.197
	NDCG@50	0.874	0.884	0.887	0.859	0.668
R	MAP@50	0.592	0.584	0.564	0.512	0.251
	NDCG@50	0.915	0.908	0.904	0.814	0.726

Table 3: Evaluation with Wikipedia seed pages and visual grouping of images.

		Phr	Word	KL	G	GQE
T	MAP@50	0.476	0.484	0.405	0.308	0.375
	NDCG@50	0.906	0.911	0.853	0.686	0.863
W	MAP@50	0.644	0.646	0.557	0.518	0.178
	NDCG@50	0.915	0.913	0.856	0.823	0.562
E	MAP@50	0.542	0.498	0.489	0.344	0.272
	NDCG@50	0.909	0.854	0.876	0.725	0.786
R	MAP@50	0.558	0.546	0.459	0.331	0.297
	NDCG@50	0.920	0.920	0.884	0.686	0.706

Table 4: Evaluation with non-Wikipedia seed pages.

from all four categories for which we used noisy and sparse non-Wikipedia seed pages (see Table 4).

The phrase-based method performs particularly well for ambiguous names. For such entities, the search engine returns a mixture of relevant and irrelevant results, while our method successfully disambiguates the correct entity. An example is the "Sans-Souci Palace" from the "Ruins" category (Figure 2). There exist (at least) two palaces with the same name, one in Potsdam and one in Haiti.

In addition to entities with ambiguous names, our method performs very well also for rare entities in the Internet image space. For example, searching for images of the computer scientist "Robert Floyd" yields only 2 correct results in the top-50 result list, on ranks 3 and 15, while the phrase-based method ranks these matches on the first two ranks.

8. CONCLUSIONS

We have shown that our phrase-based approach can substantially enhance the ranking quality of image search for difficult entities in the long tail. Some of our techniques may resemble internal ranking techniques of commercial search engines, but these are not publicly documented at all. Moreover, Google and Bing operate solely at the level of query keywords and their proximity to images, whereas our approach is specifically designed for target entities of interest and uses automatically computed keyphrases for scoring. Our experiments have demonstrated that this entity-oriented re-ranking of Google image results leads to major improvements.

9. REFERENCES

[1] S. Auer et al. DBpedia: A nucleus for a web of open data. In *ISWC/ASWC*, 2007.

[2] R. Cummins and C. O'Riordan. Learning in a pairwise term-term proximity framework for information retrieval. In *SIGIR*, pages 251–258, 2009.

Figure 2: Top-5 results (no vis. grouping) for David Gale (economist) and Sans-Souci Palace (Haiti).

[3] J. Deng et al. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[4] C. Fellbaum. *WordNet: An Electronical Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[5] E. Frank et al. Domain-specific keyphrase extraction. In *IJCAI*, pages 668–673, 1999.

[6] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[7] K. Hofmann et al. The impact of document structure on keyphrase extraction. In *CIKM*, 2009.

[8] X. Jiang, Y. Hu, and H. Li. A ranking approach to keyphrase extraction. In *SIGIR*, pages 756–757, 2009.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[10] P. S. Member and J. R. Smith. Mpeg-7 multimedia description schemes. *IEEE TCSVT*, 11:748–759, 2001.

[11] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *EMNLP*, 2004.

[12] OpenNLP. http://opennlp.sourceforge.net/.

[13] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, 1998.

[14] F. Radlinski and N. Craswell. Comparing the sensitivity of information retrieval metrics. In *SIGIR*, pages 667–674, 2010.

[15] B. C. Russell et al. LabelMe: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[16] B. Taneva, M. Kacimi, and G. Weikum. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In *WSDM*, 2010.

[17] B. Taneva, M. Kacimi, and G. Weikum. Finding images of rare and ambiguous entities. Research Report MPI-I-2011-5-002, MPII, 2011.

[18] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *SIGIR*, 2007.

[19] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE TPAMI*, 30(11):1958–1970, 2008.

[20] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM TOIS*, 22:179–214, 2004.