Finding Images of Rare and Ambiguous Entities

Bilyana Taneva
Mouna Kacimi
Gerhard Weikum

MPI–I–2011–5–002 May 2011

Authors' Addresses

Bilyana Taneva, Gerhard Weikum
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Mouna Kacimi
Free University of Bozen-Bolzano
Italy

Abstract

Despite much progress on entity-oriented Web search and automatically constructed knowledge bases with millions of entities, it is still difficult to find images of named entities like people or places. While images of famous entities 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 with a variety of entity categories show the high precision-recall quality of our approach.

Contents

1 Introduction

1.1 Motivation

The digital information world is getting more and more organized. Wikipedia contains more than 3 million articles about general concepts and named entities (people, organizations, locations, etc.) with carefully curated content, including a rich categorization system and infoboxes with salient facts. Knowledge bases such as DBpedia [2] or Freebase [1] even go further and systematically organize trillions of facts into a formal representation based on the RDF data model. And all this keeps growing and gets cleaner and better.

However, despite these impressive 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 (e.g., `images.google.com` or `images.bing.com`). Even when the top-20 results contain a handful of 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:

Difficulty of entities in the long tail. 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 New Zealand FIFA football (soccer) player Tim Brown. Searching with "Tim Brown" yields only images of the American football wide receiver with the same name. The first correct image showing the FIFA player is down on the twentieth position in the result list. For the economist David Gale, the results 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, there are Goedel prize winners (the most prestigious award for theoretical computer science) such as Carsten Lund where the top-100 search results contained less than a handful of correct images at low ranks. Unfortunately, the names themselves do not give any cues as to whether an entity is a difficult case in terms of rarity or ambiguity.

Large scale and minimal supervision. A practically viable solution should be able to operate at Web scale and with minimum human supervision. The main prior work on this problem [32] highly depends on explicitly labeled training samples for each class of entities, and performs computationally expensive query expansions and aggregation steps.

Note also that 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 (as opposed to precision at top-10 or mean reciprocal rank for the first good result).

1.2 Contribution

Our approach of finding images for rare or ambiguous named entities operates in two major phases:

Seeding phase. For a given entity of interest, we start from a salient *seed page* (or ask the user for it). This could be the Wikipedia article for the entity, but we can handle arbitrary seed pages such as people's home pages on the Web, tourism or cultural Web sites, and so on. The only requirement is that the user herself can uniquely identify the entity from solely seeing the seed page. We then automatically extract from the seed page a ranked list of

keyphrases that are characteristic for the entity. While it would seem natural to use these keyphrases for query expansion, this does not work at all with Web and image search as long queries tend to get highly diluted results.
Reaping phase. We use the entity name – and only the name – to query image search engines and to obtain a pool of candidate images fetched with their underlying Web pages. Then we use a new model for *re-ranking* the results in the candidate pool, based on the entity-characteristic keyphrases found in the seeding phase. For each image in the pool we identify *full or partial matches* of keyphrases in the Web page containing the image, and compute a new form of relevance score used for re-ranking. One problem here is that for not so difficult entities, the re-ranking may actually become inferior to the original result list. Our method includes a *robustness test* for *entity difficulty*, to ensure that we keep the original ranking if it is already good. This is fully automated, without any training or other supervision.

The paper makes the following novel contributions:

- a principled model for re-ranking of images for rare or ambiguous named entities in the long tail;
- a phrase-aware scoring model for image candidates based on partial keyphrase matches in an image's underlying Web page;
- 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;
- a scalable system architecture for organizing the entire image gathering and ranking process, suitable for distributed and parallel processing;
- 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 [34] and LabelMe [27]. TinyImage [34] is a dataset of low resolution images collected from the Internet by sending all nouns in WordNet [8] 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 [27] 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), and learning about objects embedded in a scene.

A few projects tackle the more specific problem of *integrating images into knowledge bases* [6, 32]. ImageNet [6] 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* such as the Bridge of Sighs in Venice, the Blue Mosque in Istanbul, etc. Closest to our paper is the work of [32], 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 of interest 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 light-weight technique that overcomes these limitations.

Other projects such as [25, 36, 15, 18] pursue the dual aim of *mining text and visual information* to learn tagging-style properties of images. For example, [25] proposes an unsupervised learning approach to structure, interpret, and annotate large image collections. [36] develops a consistency learning model for the problem of collecting images of people, with emphasis on celebrities. [15] uses tags to enhance the search-result diversity for images of famous landmarks (e.g., the Golden Gate Bridge). These methods work well if there is a ample redundancy in the underlying Web data, but do not carry over to our problem with people and places where images are rare and cues for them suffer from name ambiguities.

Entity search has become an established part of IR, and is presumably supported by major search engines for specific kinds of entities such as locations or consumer products. Some of the best techniques are language based models (LM's) for entities: associating a word-level probability distribution with each entity name, automatically derived from Web documents, and ranking entities as results of a keyword query by their likelihoods of generating the query [3, 7, 23] (or equivalently, by distance measures like Kullback-Leibler Divergence). In all these settings, entities are the output of a query, which itself is standard keyword search. This is different from our problem where we start with an entity name. Moreover, none of the LM-based methods carry over to finding images. Alternative methods based on PageRank-style random walks have been proposed for both entity ranking and image search [29, 14]. However, these methods improve result quality only for prominent entities; they do not work well for entities in the long tail.

Keyphrase extraction is one of the components in our system. There are both supervised (e.g., [9, 35, 13]) and unsupervised (e.g., [16, 11, 21]) approaches. Supervised techniques use training data to learn models, such as Ranking SVMs, to determine characteristic phrases. All of these methods crucially depend on the availability of manually labeled training data. Unsupervised methods, on the other hand, do not need labeled samples and are domain-independent. They typically use IR measures like tf-idf, consider n-grams or richer linguistic features, and harness document structure such as XML tags. In our work, we adopt an unsupervised approach for keyphrase extraction to avoid training bottlenecks and for domain-independence. We use noun phrases with ranking based on Mutual Information measure, as described in Section 3.

Some of our techniques are related to *proximity-aware scoring* for standard keyword search, which consider the proximity of query keywords in result documents. We consider phrases for a very different purpose. Nonetheless, we can adapt and extend some existing approaches [33, 5, 4, 28, 30, 31] and adjust them to our setting, as described in Section 4.

3 Keyphrase Mining and 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* (or ask the user for it). 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.

On first thought, a good method for extracting keyphrases would be to identify all noun phrases in the seed page. For example, from the seed page of the economist David Gale[1], we gather phrases like "American mathematician", "Professor Emeritus", "partner Sandra Gilbert", "feminist literary scholar", "poet", "daughters", "grandsons", etc. Some of them are characteristic for our entity of interest, but others dilute the focus by being either too broad or misleading (e.g., the phrase "feminist literary scholar" 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", "convex analysis", etc. These are a judiciously chosen subset

[1] `http://en.wikipedia.org/wiki/David_Gale`

of the overall set of keyphrases. In addition to this selection step, we compute weights for the focused keyphrases based on Mutual Information (or alternatively *tf-idf* measures). In the following subsections, we describe the extraction of focused keyphrases and their weighting in more detail.

3.1 Keyphrase Extraction

Noun phrases. We use the OpenNLP tool [22] to extract all noun phrases from the text in the page as a tentative set of keyphrases.
Focused keyphrases. 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 [10] for this purpose. We also considered anchor text of links in the categories and in the external links parts of the Wikipedia page, but experimentally found these to be diluting. 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.

3.2 Keyphrase Weigthing

Mutual Information. To define how well a keyphrase characterizes an entity, we first compute the reduction in uncertainty about the entity given the keyphrase. A standard measure to this end is *Mutual Information* (*MI*). High *MI* indicates large reduction in uncertainty, low *MI* - small reduction; and zero *MI* between the keyphrase and the entity means that they are independent. More formally, for each entity of interest we have two possible classes of Web pages: one for pages about the entity and one for other pages. We denote them by c and $\overline{c}$ respectively. The Mutual Information is 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 $x_k = 1$ if the page contains the keyphrase and $x_k = 0$ if the page does not contain the keyphrase, and Y is a random variable that takes values $y_c = 1$ if the page is in class c and $y_c = 0$ 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.

Alternatively, we could use the standard *tf-idf* measure to estimate the importance of a keyphrase for an entity. In our problem setting, however, *MI* and *tf-idf* are highly correlated. The reason is that the class of Web pages representing a given entity consists of a single page and hence the Mutual Information measure strongly relates to the *idf* measure. In the phrase-aware scoring models presented in Section 4 either of these measures can be used as weight for an entity keyphrase. We experimented with *MI* and *tf-idf* separately and also with a linear combination of them, but the differences were very small. In our experiments we use only *MI*.

Note that our model can also be specialized to using 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 of Image Results

For an entity of interest we use its entity name to obtain a pool of image results by querying image search engines. The image results are retrieved together with their underlying Web pages, so there is a direct correspondence between an image and a Web page that contains it. For the same entity we are also given a set of characteristic weighted phrases as described in Section 3. The scoring models presented in this section operate as follows. For every image in the pool of image results we compute a *phrase-aware score*, which is a weighted sum over *keyphrase scores*. An individual keyphrase score is estimated by identifying matches or partial matches of a given keyphrase in the Web page that contains the image of interest. Finally the images in the pool of image results are ordered by their phrase-aware scores.

More formally, for each entity of interest e we are given a pool of image results and their underlying Web pages P_e. We denote the set of entity characteristic phrases by $K_e = \{k_1(e), \ldots, k_m(e)\}$, or $K = \{k_1, \ldots, k_m\}$ when the entity is uniquely given from the context. For each image result $p \in P_e$ we compute a phrase-aware score s:

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

where $w(k_i)$ is the *MI* weight of the keyphrase k_i (see Section 3). By $\mathcal{S}(k_i, p)$ we denote the keyphrase score for keyphrase k_i and image/page p.

The best Web pages for a given entity would ideally contain an entity-characteristic keyphrase exactly in its original form, but we have to be prepared for partial matches as well. 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", "University California", "UC Berkeley", etc. In such cases, a good image page should contain as many

of the keyphrase words as possible within close distance. This approach can be thought of as a relaxed phrase-matching method with an appropriately defined scoring function.

In our framework, we compute keyphrase scores for a keyphrase in a page based on three models: *Minimum Cover*, *Büttcher's scoring model*, and *Spans scoring model*. 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. This requires important extensions of the proximity-based models, as discussed in the following subsections.

4.1 Scoring based on Minimum Cover

The Minimum Cover [33, 5] 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 the Minimum Cover 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 segment in the text 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 takes values from 0 to 1. It is equal to 1 if there is an exact match of the words in $k \cap p$ in the page, and 0 if $|k \cap p| = 0$.

The original Minimum Cover model of [33, 5] for improved result ranking of standard text queries would consider only the first factor in the formula (with adaptation to its respective 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 in the formula. It captures how many keyphrase words are missing from the page and how characteristic they are for the keyphrase. This second factor is the weighted fraction of keyphrase words that appear in the page, where the words are weighted by *MI*, as described in Section 3. The second factor takes values from 0 to 1. It is equal to 1 if $|k \cap p| = |k|$, and 0 if $|k \cap p| = 0$.

We adjust the influence of the two factors in the formula using a parameter λ. To favor Web pages containing more keyphrase words with a relatively low *mincover* value, we set $\lambda > 1$ (e.g., 2). For example, assume that a keyphrase k consists of three words with equal *MI* weights. If a page p contains only one keyphrase word, and a page q contains all three keyphrase words matched exactly, p and q would have the same score for the first factor in the formula, which is 1. The second factor in the formula for $\lambda = 2$ takes the value $\left(\frac{1}{3}\right)^2$ for p, and 1 for q, which means that q is significantly better than p.

4.2 Alternative Scoring Models

We briefly discuss two alternatives to the minimum-cover-based model. Our experiments with all models showed that the minimum-cover approach performed best, although the differences were often only small. Thus, we only outline the alternatives here and will not explicitly show them in our experiments reported later.

Büttcher's scoring model [4, 28] linearly combines the probabilistic-IR $BM25$ score and a proximity score for the words in a given query. For our purpose, we use a variant of this model: instead of a standard *idf* measure to estimate importance of words, we use the specific weighting model presented in Section 3.

Given a keyphrase k and a Web page p, we define $A_p(k)$ as the pairs of adjacent occurrences of distinct words of keyphrase k in page p with non-keyphrase words in between. We also denote the word occurring at position i in page p by $s_i(p)$, or s_i when p is given by the context. We first compute an *accumulated score acc* for each keyphrase word t in p:

$$acc_p(t) = \sum_{(i,j) \in A_p(k): s_i = t} \frac{w(s_j)}{(i-j)^2} + \sum_{(i,j) \in A_p(k): s_j = t} \frac{w(s_i)}{(i-j)^2}$$

where $w(s_i)$ is the *MI* weight of word s_i (see Section 3). The keyphrase score of keyphrase k and image page p is then given by:

$$\mathcal{S}(k,p) = \lambda BM25^*(k,p) + (1-\lambda) \sum_{t \in k \cap p} w(t) \frac{acc_p(t)(d_1+1)}{acc_p(t) + D}$$

where $k \cap p$ denotes the set of words from k that are contained in p, and $BM25^*$ is a variant of the $BM25$ score, in which we use the weighting model of Section 3 instead of the *idf* measure. Parameters D and d_1 are set to 1.2, following [28] and specializing to our setting.

The *spans-based* approach of [30, 31] segments a page text into spans based on word matches and their positions, for enhanced scoring of standard keyword search. We extend this model to our setting by measuring the density of partial matches of an entity's keyphrases in a page text.

A span for keyphrase k is a short window of adjacent words, up to a length threshold d_{max} (e.g., 20) that contains as many words of k as possible but never the same word twice. Once the same word re-appears within distance $\leq d_{max}$, the current span candidate is split into two spans. We can split after the first occurrence of the repeating word or before the second occurrence. The choice is made so that the distance between the resulting spans is maximal. This way, spans can never overlap and tend to capture coherent groups of words that partially match the given phrase. The algorithm for demarcation of spans linearly scans the word sequence and makes splitting decisions based on a bounded buffer and the threshold parameter.

To assess a page's goodness for an entity-specific keyphrase, we use spans by adjusting the $BM25$ score: the score for keyphrase k and page p is

$$\mathcal{S}(k,p) = \sum_{t \in k \cap p} w(t) \frac{rc_{tp}(d_1+1)}{rc_{tp}+D}$$

where $w(t)$ is the *MI* weight of a word t. The word frequency tf_{tp} of a word t in page p in the $BM25$ score is replaced by a *relevance contribution* rc_{tp}, based on the spans in page p, denoted by $s_i(p)$ (or s_i, if the page is given in the context), in which the word t occurs:

$$rc_{tp} = \sum_{i, t \in s_i} n_i^{\alpha} d(s_i)^{-\gamma}$$

where

$$d(s_i) = \begin{cases} pos_{i,e} - pos_{i,b} + 1, & pos_{i,e} \neq pos_{i,b} \\ d_{max}, & \text{otherwise} \end{cases}$$

is the length of the span s_i, $pos_{i,b}$, $pos_{i,e}$ are the span's begin and end positions in the page text, n_i is the number of phrase words that occur in span s_i, d_{max} is the distance threshold, and α and γ are parameters. In experiments, we used parameter settings $\alpha = \gamma = 1.5$ and $D = d_1 = 1.2$. Note that a keyphrase can consist of a single word, which means that all spans of the phrase are also of length one. In this case we assign one to the length of a phrase span. The relevance contribution rc_{tp} then becomes equivalent to the frequency of the phrase word t in the text.

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 image results and should not apply our re-ranking models described in Section 4. 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 works by processing the list of Web pages in the 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, we use the cosine similarity based on *tf-idf* values for all words in a page, where the *tf* value for a given word is based on the frequency of the word in the corresponding Web page, and the *idf* value is estimated based on the Wikipedia full corpus.

6 Grouping of Visually Similar Image Results

In addition to exploring the underlying Web page of a given image result in the pool of entity images, we also consider the visual content of the image. We have developed a method that groups visually similar images into equivalence classes and then orders these classes by relevance scores. In the following we explain the phases of this approach.

For each entity image we first compute a phrase-aware score based on the entity keyphrases as described in Section 4. Independently of this step, the images in the pool are grouped into equivalence classes of near-duplicates using a *visual similarity test*, where each class is given a representative image. Finally the representative images of the classes in the pool are assigned scores which are computed by accumulating the phrase-aware scores of all images in the same class of near-duplicate pictures. The representative images of each group are ranked by their overall scores and in the final result list of entity images only the representatives are included.

The visual similarity test between two images is based on visual features like *SIFT* feature descriptors [19] and *MPEG-7* features [20]. To determine if two images are near-duplicates we have used similar techniques as in [32].

The effects of the visual similarity grouping approach presented above are two-fold. First, by including only the representative images from each equivalence class into the final ranking of entity images, we obtain a visually diverse list of images results. Second, since the score of each representative image is computed by summing over all phrase-aware scores of the images in the same equivalence class, we obtain better evidence for the relevance of the images in the corresponding class. The reason is that every image in an equivalence class has a different underlying Web page and all these pages contain different set of entity keyphrases.

7 Implementation

We implemented all models in a Java-based prototype system. The overall system architecture is illustrated in Figure 7.1. The system consists of five major components:

- The *keyphrase analysis* component obtains a seed page for an entity from the Web or from Wikipedia. As described in Section 3, we extract entity-characteristic keyphrases from the seed page. We use the Wikipedia corpus to derive *MI* measure for each of the extracted keyphrases and also for each of the individual words in the keyphrases.

- The *candidate gathering* component sends a keyword query using only the entity name to `images.google.com` and retrieves the top-50 results for each entity. We fetch both images and the complete Web pages in which they are embedded.

- The *phrase-based scoring* component processes each image/page in the candidate pool individually. Based on the partial matches of entity keyphrases in the image pages, we assign phrase-aware scores to the images using the models from Section 4.

- The *visual grouping* component groups images into equivalence classes of near-duplicates as described in Section 6. For extracting *SIFT* and *MPEG-7* features for each image and for testing pairwise similarity between images we used the Lire [17] and IVT [12] projects.

- The *ranking* component ranks the candidate results for each entity based on their phrase-aware scores. Optionally the component may obtain grouping of image results based on visual similarity test. In this case to the representative images of each group are assigned scores by summing over all phrase-aware scores of the images in the same visual group. The ranking of the representative images is based on these accumulated scores.

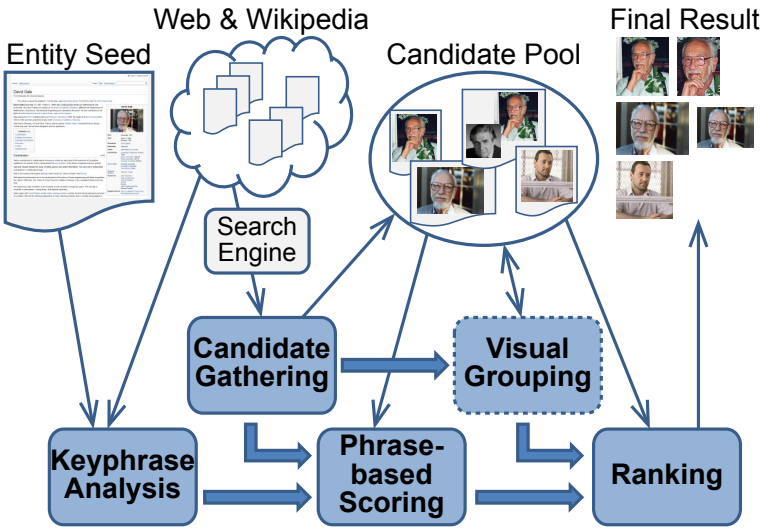

Figure 7.1: System Architecture. Rectangles are system components. Thick arrows denote control flow and thin arrows show data exchange. *Visual grouping* is an optional step.

Since the keyphrase analysis and candidate gathering components are independent, they can be easily parallelized. Different keyphrases can be processed in parallel, and different images can be downloaded independently. Partitioning the load by target entities is also straightforward. Thus, our system design easily allows scaling out the performance-critical parts of our prototype on clusters or cloud platforms.

8 Experiments

8.1 Setup

8.1.1 Methodology

We evaluated our phrase-based method of Section 4 using a variety of 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 disregarded extreme cases like "Basalt Falls" (located in BC, Canada), for which it is close to impossible to find a picture on the Internet at all (manual inspection showed not a single good result in the top-50 results of image search engines, even with reformulations of the query).

For each entity we used its seed page to extract (focused) keyphrases, for which we computed *MI* measures, as described in Section 3. Table 8.1 shows a few example entities and their best focused keyphrases ranked by *MI*. To collect a candidate pool of image results for each test entity, we posed a query with the entity name to `images.google.com` and retrieved the top-50 image results together with 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.

In total, the manual assessment consumed hundreds of person-hours. So it seems intriguing to "crowdsource" this evaluation task to Amazon Mechanical Turk (mturk). We tried this and found that the assessment of images of rare and ambiguous entities is too sophisticated for most mturk workers, as the task requires more detailed reading (beyond a short snippet) and great

	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
Entity: Per Krusell Category: Economists	1) Royal Swedish Academy of Sciences 2) macroeconomic equilibrium 3) Institute for International Economic Studies 4) rational expectations 5) Princeton University

Table 8.1: Keyphrase examples extracted from Wikipedia seed pages.

thoroughness.

8.1.2 Test Data

Our test data is based on Wikipedia categories of named entities. We used 2 Wikipedia lists with specific themes, which we perceived as typical for the long tail of entities, and 2 lists with broader but heterogeneous themes. The specific themes contain the entities of the following categories:

- "Turing Award laureates" with 56 entities, out of which 34 are difficult, as concluded by the test for entity difficulty presented in Section 5 and
- "Waterfalls of British Columbia" with 20 entities, with 14 difficult ones.

The broad themes contain the entities of the lists:

- "Economists" with 589 entities and
- "Ruins" with 788 entities.

We completely coassessed the image results for all entities in the first two categories. For the two broader and much larger categories we randomly

sampled 25 entities from each, excluding extremely prominent entities with perfect precision on the first page of Google's result list. We applied the entity difficulty test on the two samples of 25 entities and there were 23 difficult entities from "Economists" category and 17 from the "Ruins" category.

For retrieving the candidate pool, we used the Wikipedia article name as a keyword query to `images.google.com`, but removed qualifiers in parentheses (e.g., "John McCarthy (computer scientist)" became "John McCarthy"), as a user would usually not use a search engine with such a special and long query.

For all entity categories listed above we also performed experiments where the seed pages for the entities are not Wikipedia articles, but simple Web pages varying in text length and quality of entity description.

8.1.3 Methods under Comparison

We compare five methods:

- our new ranking method based on the minimum-cover matching of (focused) keyphrases,
- the words-aware model as a special case of our method,
- the original search engine, as a main baseline,
- the original search engine with query expansion, by including the highest-*MI* keyphrase in the entity query,
- 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 and entity retrieval [24, 37, 38].

Recall from Section 3 that our phrase-aware model can be specialized to words only, by selecting all single words that constitute the entity-specific keyphrases. The words-aware score for an image page is the sum of the *MI* weights of all keyphrase words that appear in the page.

Another possible opponent to our approach would be the method of [32] based on query expansions. However, that method is not really comparable to ours, since it depends on an ontological type system for entities, on training-based weights for each type, and on a knowledge base with salient RDF facts about each entity. Therefore, we do not include such a comparison here.

8.1.4 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, which would be decisive for finding a single or a few best photos of a celebrity but are less insightful for finding many images of difficult entities. We compute MAP similarly to [26] by considering only the top-k results:

$$MAP@k(R) = \frac{1}{|E|} \sum_{i=1}^{|E|} \frac{1}{n_i} \sum_{j=1}^{k} rel(d_j^i) Precision@j(R, e_i)$$

where E is the set of test entities, n_i is the number of known relevant results for entity e_i d_j^i is the j^{th} ranked result for entity e_i returned by a retrieval algorithm R, and $rel(d_j^i)$ is the binary relevance assessment for this result. In our setting we assume that the set of relevant results for an entity consists of the relevant ones retrieved by the original entity-name query or the expanded query with the best-*MI* keyphrase. The NDCG measure reflects the relevance of results using their (geometrically weighted) positions in the result list:

$$NDCG@k(R) = \frac{1}{|E|} \sum_{i=1}^{|E|} N_{ki} \sum_{j=1}^{k} \frac{2^{rel(d_j^i)} - 1}{log_2(1+j)}$$

where N_{ki} is a normalization factor calculated to make NDCG at k equal to 1 in case of perfect ranking. The MRR measure is given by: $MRR(R) = (\sum_{i=1}^{|E|} 1/r_i)/|E|$ where r_i is the rank of the first relevant result for the i^{th} entity.

8.2 Results

8.2.1 Ranking Based on Wikipedia Seed Pages

The results for the ranking models using Wikipedia seed pages are shown in Table 8.2. The phrase-aware model almost always improves all measures in comparison to the original search engine and the search engine with query expansion. In terms of our primary measures, MAP and NDCG, the original search engine is better than the phrase-aware model only for the category "Waterfalls of BC" for NDCG@20. The gains of the phrase-based model depend on the category with highest gains on the "Ruins" category.

		Phr	Word	KL	G	GQE
T	MAP@50	0.599	0.591	0.591	0.587	0.344
	MAP@20	0.360	0.354	0.355	0.350	0.276
	NDCG@50	0.931	0.924	0.926	0.885	0.893
	NDCG@20	0.943	0.938	0.942	0.900	0.904
	P@10	0.759	0.759	0.756	0.770	0.638
	P@20	0.690	0.684	0.676	0.684	0.515
	MRR	0.956	0.941	0.944	0.897	0.910
W	MAP@50	0.618	0.593	0.589	0.588	0.210
	MAP@20	0.448	0.409	0.404	0.406	0.161
	NDCG@50	0.894	0.882	0.876	0.883	0.682
	NDCG@20	0.902	0.898	0.885	0.907	0.686
	P@10	0.714	0.714	0.671	0.700	0.378
	P@20	0.625	0.603	0.582	0.611	0.314
	MRR	0.886	0.889	0.848	0.964	0.611
E	MAP@50	0.628	0.621	0.625	0.572	0.163
	MAP@20	0.517	0.504	0.506	0.437	0.141
	NDCG@50	0.895	0.887	0.897	0.855	0.664
	NDCG@20	0.909	0.904	0.923	0.879	0.667
	P@10	0.678	0.674	0.656	0.569	0.291
	P@20	0.541	0.524	0.502	0.472	0.233
	MRR	0.935	0.917	0.946	0.935	0.625
R	MAP@50	0.594	0.578	0.552	0.499	0.259
	MAP@20	0.460	0.444	0.400	0.335	0.207
	NDCG@50	0.934	0.924	0.909	0.823	0.742
	NDCG@20	0.946	0.942	0.932	0.825	0.741
	P@10	0.765	0.747	0.723	0.635	0.447
	P@20	0.668	0.644	0.603	0.565	0.359
	MRR	0.970	1.000	0.970	0.779	0.778

Table 8.2: Evaluation for the phrase-aware model (Phr), words-aware model (Word), KL-divergence-based model (KL), Google (G), and Google with query expansion (GQE) for the entity sets: Turing Award winners (T), Waterfalls of BC (W), Ruins (R), and Economists (E) with Wikipedia seed pages.

		Phr(F)	Phr(N)	KL	G
T	MAP@50	0.599	0.595	0.591	0.587
	NDCG@50	0.931	0.929	0.926	0.885
W	MAP@50	0.618	0.592	0.589	0.588
	NDCG@50	0.894	0.880	0.876	0.883
E	MAP@50	0.628	0.591	0.625	0.572
	NDCG@50	0.895	0.864	0.897	0.855
R	MAP@50	0.594	0.592	0.552	0.499
	NDCG@50	0.934	0.932	0.909	0.823

Table 8.3: Evaluation for the phrase-aware model with focused phrases (Phr(F)) and with all noun phrases (Phr(N)) extracted from Wikipedia seed pages. For abbreviations see Table 8.2.

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 much worse than the original search engine. The only exception is the "Turing Award laureates" category in terms of NDCG. The reason for this inferior behavior 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.

In Table 8.2 the results for the phrase-aware model are achieved by using only focused phrases as entity-specific keyphrases (see Section 3). Table 8.3 shows a comparison for the phrase-aware model with focused phrases versus using all noun phrases of the seed page. The results clearly show that focused keyphrases are essential for the good performance of the phrase-based model.

Overall, the main insight from these experiments is that the phrase-based model with focused keyphrases achieves significant gains over all alternative models. It wins in many cases, and these gains are statistically significant. In a few cases, other methods perform comparably or are slightly better, but these differences are negligible.

		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 8.4: Evaluation for entities with Wikipedia seed pages and visual grouping of images. For abbreviations see Table 8.2.

8.2.2 Ranking with Visual Grouping of Images

Table 8.4 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 amazingly well in this setting, but still lose against the phrase-aware model in most cases.

On average, the gains of the phrase-aware model over the alternatives are slightly lower, compared to using the same model without visual grouping. This is because duplicates and near-duplicates of good results are now discounted. We speculate that the gains would be higher for larger candidate pool per entity.

8.2.3 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

		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 8.5: Evaluation for entities with non-Wikipedia seed pages. For abbreviations see Table 8.2.

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 8.5. For these very difficult entities, we observe that the phrase-aware model outperforms both 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 of the keyphrases were merely one-word phrases.

8.3 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 and the most difficult entities from all four categories for which we used noisy and sparse non-Wikipedia seed pages (see Table 8.5).

The phrase-based method performs particularly well for ambiguous names. Examples are given in Figure 8.1. For such entities, the search engine returns a mixture of relevant and irrelevant results for the particular entity of interest,

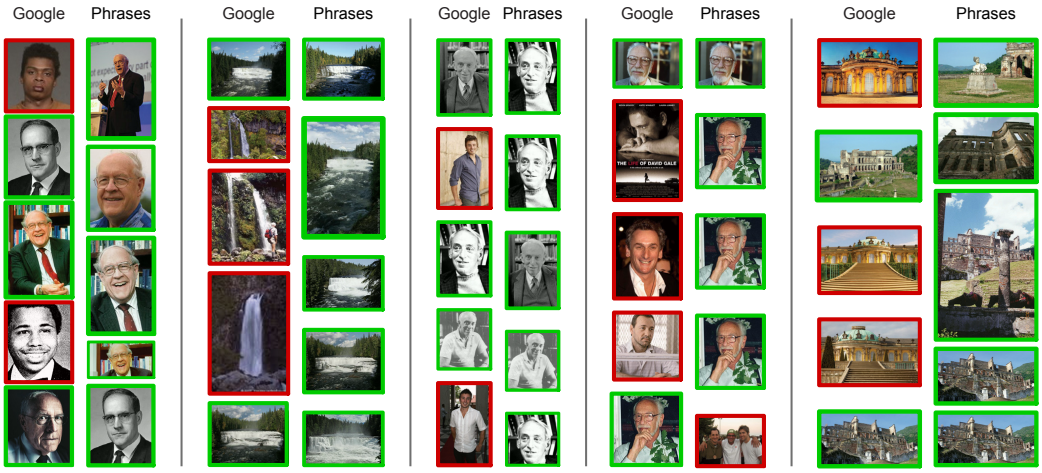

Figure 8.1: Examples for phrase-aware rankings (no vis. grouping): Fred Brooks - Turing Award winner; Dawson Falls - waterfall in BC; James Tobin and David Gale - economists; Sans-Souci Palace - ruin in Haiti

while our method successfully disambiguates the correct entity. An example is the "Sans-Souci Palace" from the "Ruins" category. There exist (at least) two palaces that have the same name, one in Potsdam and the other one in Haiti.

In addition to entities with ambiguous names, our method performs very well also for rare entities in the Internet image space. An example for such entity is the prominent computer scientist "Robert Floyd". If you search for images, Google returns 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.

One aspect in which our approach could be improved is the following. Since the ranking of entity images can only be as good as the information in their underlying Web pages, in some cases we boost the rank of an image from a highly relevant and informative page, even though the image itself is not good. A possible way to overcome this issue could be to reason on the images themselves, which we did in Section 6. However, because of the diversification efforts of search engines, our method was not able to gather enough statistical data and improve on the approach without grouping. The groups of near-duplicates had only very few images on average. As future work, we plan to address this issue and compile larger pools of images.

9 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 technqiues 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.

Our ongoing and future work will include experiments with larger collections, and going beyond the top-50 results from a search engine. To this end, we consider using the TREC ClueWeb corpus, but also plan on performing entity-focused crawling on the live Web. Another possible improvement of our approach is to consider multiple languages. For example, we could include different Wikipedia editions to determine good entity seeds, and then work with multi-lingual sets of keyphrases.

Bibliography

[1] http://www.freebase.com/.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. Dbpedia: A nucleus for a web of open data. In *ISWC/ASWC*, 2007.

[3] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Inf. Process. Manage.*, 45:1–19, January 2009.

[4] S. Büttcher, C. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *SIGIR*, pages 621–622, 2006.

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

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[7] H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR*, pages 418–430, 2007.

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

[9] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *IJCAI*, pages 668–673, 1999.

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

[11] K. Hofmann, M. Tsagkias, E. Meij, and M. de Rijke. The impact of document structure on keyphrase extraction. In *CIKM*, pages 1725–1728, 2009.

[12] IVT. http://ivt.sourceforge.net/.

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

[14] Y. Jing and S. Baluja. Pagerank for product image search. In *WWW*, pages 307–316, 2008.

[15] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *WWW*, pages 297–306, 2008.

[16] N. Kumar and K. Srinathan. Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *DocEng'08*, pages 199–208.

[17] Lire. http://www.semanticmetadata.net/lire/.

[18] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang. Retagging social images based on visual and semantic consistency. In *WWW*, pages 1149–1150, 2010.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, pages 91–110, 2004.

[20] P. S. Member and J. R. Smith. Mpeg-7 multimedia description schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001.

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

[22] openNLP. http://opennlp.sourceforge.net/.

[23] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *CIKM*, pages 731–740, 2007.

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

[25] T. Quack, B. Leibe, and L. J. V. Gool. World-scale mining of objects and events from community photo collections. In *CIVR*, pages 47–56, 2008.

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

[27] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 2008.

[28] R. Schenkel, A. Broschart, S. Hwang, M. Theobald, and G. Weikum. Efficient text proximity search. In *SPIRE*, pages 287–299, 2007.

[29] P. Serdyukov, H. Rode, and D. Hiemstra. Modeling expert finding as an absorbing random walk. In *SIGIR*, pages 797–798, 2008.

[30] R. Song, M. J. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu. Viewing term proximity from a different perspective. In *ECIR*, pages 346–357, 2008.

[31] K. M. Svore, P. H. Kanani, and N. Khan. How good is a span of terms?: exploiting proximity to improve web retrieval. In *SIGIR*, pages 154–161, 2010.

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

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

[34] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.

[35] Y. B. Wu, Q. Li, R. S. Bot, and X. Chen. Finding nuggets in documents: A machine learning approach. *JASIST*, 57:740–752, 2006.

[36] J. Yagnik and A. Islam. Learning people annotation from the web via consistency learning. In *MIR*, 2007.

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

[38] C. Zhai and J. Lafferty. A risk minimization framework for information retrieval. *Inf. Process. Manage.*, 42:31–55, 2006.

Below you find a list of the most recent research reports of the Max-Planck-Institut für Informatik. Most of them are accessible via WWW using the URL `http://www.mpi-inf.mpg.de/reports`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
– Library and Publications –
Campus E 1 4

D-66123 Saarbrücken

E-mail: `library@mpi-inf.mpg.de`

MPI-I-2010-RG1-001	M. Suda, C. Weidenbach, P. Wischnewski	On the saturation of YAGO
MPI-I-2010-5-008	S. Elbassuoni, M. Ramanath, G. Weikum	Query relaxation for entity-relationship search
MPI-I-2010-5-007	J. Hoffart, F.M. Suchanek, K. Berberich, G. Weikum	YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia
MPI-I-2010-5-006	A. Broschart, R. Schenkel	Real-time text queries with tunable term pair indexes
MPI-I-2010-5-005	S. Seufert, S. Bedathur, J. Mestre, G. Weikum	Bonsai: Growing Interesting Small Trees
MPI-I-2010-5-003	A. Anand, S. Bedathur, K. Berberich, R. Schenkel	Efficient temporal keyword queries over versioned text
MPI-I-2010-5-002	M. Theobald, M. Sozio, F. Suchanek, N. Nakashole	URDF: Efficient Reasoning in Uncertain RDF Knowledge Bases with Soft and Hard Rules
MPI-I-2010-5-001	K. Berberich, S. Bedathur, O. Alonso, G. Weikum	A language modeling approach for temporal information needs
MPI-I-2010-1-001	C. Huang, T. Kavitha	Maximum cfardinality popular matchings in strict two-sided preference lists
MPI-I-2009-RG1-005	M. Horbach, C. Weidenbach	Superposition for fixed domains
MPI-I-2009-RG1-004	M. Horbach, C. Weidenbach	Decidability results for saturation-based model building
MPI-I-2009-RG1-002	P. Wischnewski, C. Weidenbach	Contextual rewriting
MPI-I-2009-RG1-001	M. Horbach, C. Weidenbach	Deciding the inductive validity of $\forall\exists^*$ queries
MPI-I-2009-5-007	G. Kasneci, G. Weikum, S. Elbassuoni	MING: Mining Informative Entity-Relationship Subgraphs
MPI-I-2009-5-006	S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, G. Weikum	Scalable phrase mining for ad-hoc text analytics
MPI-I-2009-5-005	G. de Melo, G. Weikum	Towards a Universal Wordnet by learning from combined evidenc
MPI-I-2009-5-004	N. Preda, F.M. Suchanek, G. Kasneci, T. Neumann, G. Weikum	Coupling knowledge bases and web services for active knowledge
MPI-I-2009-5-003	T. Neumann, G. Weikum	The RDF-3X engine for scalable management of RDF data
MPI-I-2009-5-002	M. Ramanath, K.S. Kumar, G. Ifrim	Generating concise and readable summaries of XML documents
MPI-I-2009-4-006	C. Stoll	Optical reconstruction of detailed animatable human body models
MPI-I-2009-4-005	A. Berner, M. Bokeloh, M. Wand, A. Schilling, H. Seidel	Generalized intrinsic symmetry detection
MPI-I-2009-4-004	V. Havran, J. Zajac, J. Drahokoupil, H. Seidel	MPI Informatics building model as data for your research
MPI-I-2009-4-003	M. Fuchs, T. Chen, O. Wang, R. Raskar, H.P.A. Lensch, H. Seidel	A shaped temporal filter camera
MPI-I-2009-4-002	A. Tevs, M. Wand, I. Ihrke, H. Seidel	A Bayesian approach to manifold topology reconstruction

MPI-I-2009-4-001	M.B. Hullin, B. Ajdin, J. Hanika, H. Seidel, J. Kautz, H.P.A. Lensch	Acquisition and analysis of bispectral bidirectional reflectance distribution functions
MPI-I-2008-RG1-001	A. Fietzke, C. Weidenbach	Labelled splitting
MPI-I-2008-5-004	F. Suchanek, M. Sozio, G. Weikum	SOFI: a self-organizing framework for information extraction
MPI-I-2008-5-003	G. de Melo, F.M. Suchanek, A. Pease	Integrating Yago into the suggested upper merged ontology
MPI-I-2008-5-002	T. Neumann, G. Moerkotte	Single phase construction of optimal DAG-structured QEPs
MPI-I-2008-5-001	G. Kasneci, M. Ramanath, M. Sozio, F.M. Suchanek, G. Weikum	STAR: Steiner tree approximation in relationship-graphs
MPI-I-2008-4-003	T. Schultz, H. Theisel, H. Seidel	Crease surfaces: from theory to extraction and application to diffusion tensor MRI
MPI-I-2008-4-002	D. Wang, A. Belyaev, W. Saleem, H. Seidel	Estimating complexity of 3D shapes using view similarity
MPI-I-2008-1-001	D. Ajwani, I. Malinger, U. Meyer, S. Toledo	Characterizing the performance of Flash memory storage devices and its impact on algorithm design
MPI-I-2007-RG1-002	T. Hillenbrand, C. Weidenbach	Superposition for finite domains
MPI-I-2007-5-003	F.M. Suchanek, G. Kasneci, G. Weikum	Yago : a large ontology from Wikipedia and WordNet
MPI-I-2007-5-002	K. Berberich, S. Bedathur, T. Neumann, G. Weikum	A time machine for text search
MPI-I-2007-5-001	G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, G. Weikum	NAGA: searching and ranking knowledge
MPI-I-2007-4-008	J. Gall, T. Brox, B. Rosenhahn, H. Seidel	Global stochastic optimization for robust and accurate human motion capture
MPI-I-2007-4-007	R. Herzog, V. Havran, K. Myszkowski, H. Seidel	Global illumination using photon ray splatting
MPI-I-2007-4-006	C. Dyken, G. Ziegler, C. Theobalt, H. Seidel	GPU marching cubes on shader model 3.0 and 4.0
MPI-I-2007-4-005	T. Schultz, J. Weickert, H. Seidel	A higher-order structure tensor
MPI-I-2007-4-004	C. Stoll, E. de Aguiar, C. Theobalt, H. Seidel	A volumetric approach to interactive shape editing
MPI-I-2007-4-003	R. Bargmann, V. Blanz, H. Seidel	A nonlinear viseme model for triphone-based speech synthesis
MPI-I-2007-4-002	T. Langer, H. Seidel	Construction of smooth maps with mean value coordinates
MPI-I-2007-4-001	J. Gall, B. Rosenhahn, H. Seidel	Clustered stochastic optimization for object recognition and pose estimation
MPI-I-2007-2-001	A. Podelski, S. Wagner	A method and a tool for automatic veriication of region stability for hybrid systems
MPI-I-2007-1-003	A. Gidenstam, M. Papatriantafilou	LFthreads: a lock-free thread library
MPI-I-2007-1-002	E. Althaus, S. Canzar	A Lagrangian relaxation approach for the multiple sequence alignment problem
MPI-I-2007-1-001	E. Berberich, L. Kettner	Linear-time reordering in a sweep-line algorithm for algebraic curves intersecting in a common point
MPI-I-2006-5-006	G. Kasnec, F.M. Suchanek, G. Weikum	Yago - a core of semantic knowledge
MPI-I-2006-5-005	R. Angelova, S. Siersdorfer	A neighborhood-based approach for clustering of linked document collections
MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining linguistic and statistical analysis to extract relations from web documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment texture editing in monocular video sequences based on color-coded printing patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: index-access optimized top-k query processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-aware global df estimation in distributed information retrieval systems