Automatic Population of Knowledge Bases with Multimodal Data about Named Entities

Bilyana Taneva

Thesis for obtaining the title of Doctor of Engineering
of the Faculties of Natural Sciences and Technology
of Saarland University

Max-Planck Institute for Informatics
Saarbrücken, Germany, 2013

Dean:	Prof. Mark Groves Faculty of Mathematics and Computer Science Saarland University Saarbrücken, Germany
Colloquium:	2013-08-12 Max-Planck Institute for Informatics Saarbrücken, Germany

Examination Board

Supervisor and First Reviewer:	Prof. Gerhard Weikum Department for Databases and Information Systems Max-Planck Institute for Informatics Saarbrücken, Germany
Second Reviewer:	Dr. Fabian M. Suchanek Otto Hahn Research Group "Ontologies" Max-Planck Institute for Informatics Saarbrücken, Germany
Third Reviewer:	Dr. Mounia Lalmas Yahoo! Labs Barcelona, Spain
Chairman:	Prof. Dietrich Klakow Spoken Language Systems Saarland University Saarbrücken, Germany
Research Assistant:	Dr. Klaus Berberich Department for Databases and Information Systems Max-Planck Institute for Informatics Saarbrücken, Germany

Declaration
I hereby solemnly declare that this work was created on my own, using only the resources and tools mentioned. Information taken from other sources or indirectly adopted data and concepts are explicitly acknowledged with references to the respective sources. This work has not been submitted in a process for obtaining an academic degree elsewhere in the same or in similar form.

Eidesstattliche Versicherung
Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, May 2013
Bilyana Taneva

Abstract

Knowledge bases are of great importance for Web search, recommendations, and many Information Retrieval tasks. However, maintaining them for not so popular entities is often a bottleneck. Typically, such entities have limited textual coverage and only a few ontological facts. Moreover, these entities are not well populated with multimodal data, such as images, videos, or audio recordings.

The goals in this thesis are (1) to populate a given knowledge base with multimodal data about entities, such as images or audio recordings, and (2) to ease the task of maintaining and expanding the textual knowledge about a given entity, by recommending valuable text excerpts to the contributors of knowledge bases.

The thesis makes three main contributions. The first two contributions concentrate on finding images of named entities with high precision, high recall, and high visual diversity. Our main focus are less popular entities, for which the image search engines fail to retrieve good results. Our methods utilize background knowledge about the entity, such as ontological facts or a short description, and a visual-based image similarity to rank and diversify a set of candidate images.

Our third contribution is an approach for extracting text contents related to a given entity. It leverages a language-model-based similarity between a short description of the entity and the text sources, and solves a budget-constraint optimization program without any assumptions on the text structure. Moreover, our approach is also able to reliably extract entity related audio excerpts from news podcasts. We derive the time boundaries from the usually very noisy audio transcriptions.

Kurzfassung

Wissensbasen wird bei der Websuche, bei Empfehlungsdiensten und vielen anderen Information Retrieval Aufgaben eine große Bedeutung zugeschrieben. Allerdings stellt sich deren Unterhalt für weniger populäre Entitäten als schwierig heraus. Üblicherweise ist die Anzahl an Texten über Entitäten dieser Art begrenzt, und es gibt nur wenige ontologische Fakten. Außerdem sind nicht viele multimediale Daten, wie zum Beispiel Bilder, Videos oder Tonaufnahmen, für diese Entitäten verfügbar.

Die Ziele dieser Dissertation sind daher (1) eine gegebene Wissensbasis mit multimedialen Daten, wie Bilder oder Tonaufnahmen, über Entitäten anzureichern und (2) die Erleichterung der Aufgabe Texte über eine gegebene Entität zu verwalten und zu erweitern, indem den Beitragenden einer Wissenbasis nützliche Texteausschnitte vorgeschlagen werden.

Diese Dissertation leistet drei Hauptbeiträge. Die ersten zwei Beiträge sind im Gebiet des Auffindens von Bildern von benanntend Entitäten mit hoher Genauigkeit, hoher Trefferquote, und hoher visueller Vielfalt. Das Hauptaugenmerk liegt auf den weniger populären Entitäten bei denen die Bildersuchmaschinen normalerweise keine guten Ergebnisse liefern. Unsere Verfahren benutzen Hintergrundwissen über die Entität, zum Beispiel ontologische Fakten oder eine Kurzbeschreibung, so wie ein visuell-basiertes Bilderähnlichkeitsmaß um die Bilder nach Rang zu ordnen und um eine Menge von Bilderkandidaten zu diversifizieren.

Der dritte Beitrag ist ein Ansatz um Textinhalte, die sich auf eine gegebene Entität beziehen, zu extrahieren. Der Ansatz nutzt ein auf einem Sprachmodell basierendes Ähnlichkeitsmaß zwischen einer Kurzbeschreibung der Entität und den Textquellen und löst zudem ein Optimierungsproblem mit Budgetrestriktion, das keine Annahmen an die Textstruktur macht. Darüber hinaus ist der Ansatz in der Lage Tonaufnahmen, welche in Beziehung zu einer Entität stehen, zuverlässig aus Nachrichten-Podcasts zu extrahieren. Dafür werden zeitliche Abgrenzungen aus den normalerweise sehr verauschten Audiotranskriptionen hergeleitet.

Contents

Chapter 1

Introduction

1.1 Motivation

Knowledge bases have become of great importance for Web search, Knowledge Exploration and Analysis, and many other information retrieval and natural language processing tasks. Prominent examples of large-scale knowledge bases include DBpedia (`dbpedia.org`), Yago (`yago-knowledge.org`), and the Google Knowledge Graph which is centered around Freebase (`freebase.com`). Knowledge bases are large collections of world knowledge, represented in a machine-readable format. They contain information about real-world objects, called *entities*, such as people, places, events, songs, books, etc., and *facts* about these entities, such as the birth date of people, the location of places, the genres of books, and many others.

Many knowledge bases are derived from Wikipedia using its infoboxes, its category system, its articles' text, and other valuable information. On the other hand, the automatic maintenance and expansion of knowledge bases, using other natural language resources in addition to Wikipedia, have become an important task in many research projects. Prominent examples include the automatic extraction of facts about entities [Carlson et al., 2010; Nakashole et al., 2011] and the extraction of new relation patterns between entities [Banko et al., 2007a; Fader et al., 2011; Nakashole et al., 2012] among others.

1.1.1 Lack of Multimodal Data in Knowledge Bases

Despite the proliferation of Wikipedia and the advances in information extraction, there are still major shortcomings in the organization of multimodal data about entities, such as images, videos, or audio recordings. Even if Wikipedia contains a large amount of articles with images, these articles are mainly about prominent entities, such as celebrities, major events, or popular touristic attractions. Articles for lesser known people or places very often do not contain a picture of the entity. Moreover, even for the more popular entities, currently Wikipedia contains only a few images while some users might be interested in seeing a larger variety of pictures showing the entities (e.g., people shown at different ages or occasions, landmarks shown from different perspectives or at different weather/light conditions, etc.). There is also lack of multimodal data in the knowledge bases. Since today's knowledge bases are mostly centered around Wikipedia, they contain only images or videos from

Wikipedia. Moreover, information extraction methods, which aim at automatically expanding knowledge bases, consider mainly facts about entities, relation types, and other textual information. They are less interested in populating and maintaining knowledge bases with multimodal data.

Representing people, places, or other real-world entities with multimodal information is beneficial for many reasons. Many users comprehend entities easier and faster by looking at images that show the entities rather than reading articles about them. In general, the visual perception of objects or people is crucial for their understanding. For example, a good description of a person includes not only textual information about the person, but also an image that shows her/him. Regarding audio information, listening to a carefully chosen audio recording of a major event or in general about an entity of interest can provide the user easily with concrete information about the entity. Furthermore, many users would rather watch a video about an event rather than read about it.

1.1.2 Bottleneck of Maintenance on Less Popular Entities

While Wikipedia and the knowledge bases are up-to-date on prominent entities, their maintenance on less prominent entities and the acquisition of knowledge about newly emerging entities are bottlenecks. The reason is that human contributors need to continuously identify and read relevant sources about entities to update articles or structured knowledge. We can notice the delay in the population of Wikipedia by considering its stub pages. There are hundreds of thousands of articles containing the statement "*This article about ... is a stub. You can help Wikipedia by expanding it.*". In many cases, information about such less popular entities is easily available on the Web and could be used to expand the knowledge about the entity.

To help authors/editors of knowledge bases to maintain encyclopedic contents in a timely manner we can provide them with intelligent recommendations about concise text fragments that contain relevant information for given entities of interest. These recommendations can be used as an input for updating or expanding the knowledge base without the need to search for other related sources.

1.2 Problem Statement

In this thesis we consider the following high-level problem statement. Given an entity, represented by a background information in the form of facts from a knowledge base or a short description (e.g., part of the Wikipedia article about the entity), our goal is to find more related information about the entity such as (1) images, (2) audio recordings, and (3) text fragments, which are relevant and valuable for the entity of interest.

Our objective is twofold: (1) to populate an existing knowledge base with multimodal information about entities, and (2) to ease the task of maintaining and expanding the textual knowledge about entities by recommending valuable text excerpts related to them. We focus on less prominent entities or such with ambiguous names. Furthermore, we aim at finding a diverse set of results without unnecessary

repetition. This means that our goal is to find visually different images, and different by content text fragments or audio recordings.

1.3 Challenges

Our problem has various difficulties depending on the type of extracted information. In the following we discuss the main challenges related to extracting images, audio recordings, and text contents relevant for a given entity of interest.

1.3.1 Extraction of Images for Entities

One option to find images of people or places is to use image search engines. However this works well only for popular entities, like celebrities or touristic attractions. It remains difficult to find images of less popular entities. A query with the entity name of a lesser known person or place returns good results on the top ranks but the precision quickly degrades. For a human user who knows the entity of interest it may be good enough if the top search results contain some correct pictures, but this is insufficient for an automatic extraction of images for entities. Another problem is the ambiguity of the entity names. The search results for entities with ambiguous names are typically a mixture of images showing different entities with the same name. To automatically populate a knowledge base, the correct images need to be discriminated from the images showing other entities with the same name. Finally, we aim at finding a visually diverse set of images for each entity of interest. However, often it is difficult to locate different pictures for a given entity using image search engines and querying with the entity name only.

1.3.2 Extraction of Audio Recordings for Entities

Finding audio recordings about an entity of interest requires analyzing the topic of each candidate recording and comparing it to the background knowledge about the entity. Speech-to-text transcriptions do not have structure in terms of paragraphs or sentences, and they are highly noisy due to the errors of the speech transcription. Furthermore, since we aim at populating a knowledge base with audio recordings about a given entity, we need to find such recordings or specific parts of recordings, which are highly related to the entity, as opposed to any audio stream which mentions briefly the entity.

1.3.3 Extraction of Text Contents for Entities

To extract text contents related to a given entity from various text sources, including speech-to-text transcriptions, news streams, or blog postings, our approaches need to be independent of the structure in the text, such as paragraphs or sentences. It is also important that the knowledge-base-contributors are provided with concise information about the entity of interest with evidence of relevance but without unnecessary repetition. In addition, the extracted text contents need to be novel to the human contributors in the sense it should not be already covered by the input knowledge about the entity.

1.4 Contributions

In this thesis we propose methods for populating knowledge bases with multimodal information, and for maintaining and expanding knowledge bases with textual information about entities. Our main contributions are as follows.

1.4.1 Knowledge Kaleidoscope with Query Expansions

First, we propose an approach for gathering images of entities which constructs a set of expanded queries for each entity of interest, where the expansions are automatically derived from already known facts in a knowledge base. The expanded queries are then posed to image search engines and for each query we retrieve the top-n image results. We rank the collected images by merging the results from the different query expansions, with specific weights for each query. The weights are automatically learned from training samples. This approach can be seen as a form of consistency checking of the search results, as reflected in the overlap of the results for the different query expansions. In addition, we consider image-content similarities among different images in order to enhance the visual diversity of the final results. We presented this approach and our experimental results at WSDM 2010 [Taneva et al., 2010].

1.4.2 Knowledge Kaleidoscope with Keyphrases

In addition to our approach for finding images of entities using query expansions, we propose a very different and more robust solution for the same problem. Since knowledge bases can be rather sparse in facts about less popular entities, we leverage a salient seed description about the entity of interest. This could be the Wikipedia article of the entity or an arbitrary short textual description of the entity. We automatically extract from the seed page a ranked list of keyphrases that are characteristic for the entity. Using only the entity name we query image search engines and obtain a pool of candidate images fetched with their underlying Web pages. Then we use a new model for ranking the candidate images which is based on the entity-specific keyphrases found earlier. For each image we identify full or partial matches of the keyphrases in the Web page containing the image, and compute a relevance score which is used for ranking.

Since for not so difficult entities, such as celebrities, popular landmarks, or entities with unambiguous names, the original search engine results are already very good, we do not need to run sophisticated algorithms for re-ranking of images. To selectively run our algorithms only when the search results can be improved, we propose an algorithm which estimates the difficulty of retrieving good images for a given entity. We apply our algorithms for re-ranking of images only for difficult entities. We presented this second approach of finding images for entities at CIKM 2011 [Taneva et al., 2011].

1.4.3 Entity-Knowledge Maintenance

The third contribution in this thesis is an approach for extracting text contents highly related to a given entity. Our method starts with a short seed text about

the entity, from which we derive a statistical language model for it. We obtain text sources which are potentially related to the entity of interest. Using minimal assumptions on their structure, we interpret the text sources as a stream of words. Then, we estimate how related to the entity is the stream of words at each position, by considering individual words as points of interest. For each word we compute a language-model-based similarity between the entity seed text and the context of the word. To capture coherent text excerpts with high score mass while meeting the constraint that a user should not be overwhelmed with information, we develop a budget-constraint optimization algorithm. It identifies variable-length text fragments which are salient for the entity and novel with regard to the entity seed text.

Since we do not pose any restrictions on the structure of the text sources, our method can be used to retrieve (parts of) audio recordings which are related to a given entity. From the speech-to-text transcriptions of given candidate audio recordings and the language model of the input entity, we extract excerpts which are related to the entity. We associate with the extracted transcription excerpts their respective audio signals and thus retrieve audio fragments which are focused on the entity of interest. This work has been accepted for publication at CIKM 2013 [Taneva and Weikum, 2013].

1.5 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 introduces the basics for knowledge bases and image-content similarities which we exploit in the thesis. Chapter 3 provides an overview of the work related to our three main contributions. In Chapter 4 we present our approach for finding images of named entities using query expansions. In Chapter 5 we present our approach for finding images of entities using entity-characteristic keyphrases. In Chapter 6 we describe our methods for extracting text fragments related to a given entity of interest. In addition, we also present an application scenario for these methods, namely the extraction of audio recordings which are relevant for a given entity. Finally, in Chapter 7 we summarize the research presented in this thesis.

Chapter 2

Background

2.1 Knowledge Bases

Knowledge bases are large collections of world knowledge. Manually compiled lexical and common-sense knowledge bases, like WordNet [Fellbaum, 1998] or Cyc [Lenat, 1995] have very high quality and exist for more than a decade. These are large repositories of general concepts, semantic classes, and relationships between classes, like subclass-of or part-of relationships. For example, WordNet knows that artists are humans, that electric guitar is a subclass of the guitar class, or that British Columbia is part of Canada. WordNet also knows the various meanings of the words. However, these common-sense knowledge bases have very limited knowledge about individual entities, such as people, landmarks, songs, events, countries, etc. For example, they do not know that David Patterson is a computer scientist or the countries that share a border with Germany.

Automatically constructed knowledge bases have been developed with great success in the last few years. Prominent examples include DBpedia [Bizer et al., 2009], Yago [Suchanek et al., 2007], KnowItAll [Etzioni et al., 2005], TextRunner [Banko et al., 2007b], or commercial services like Freebase (`freebase.com`), Evi which was formerly known as TrueKnowledge (`evi.com`), or WolframAlpha (`wolframalpha.com`). These are rich resources of real world entities like people, events, places, organizations, etc., automatically organized into semantic classes such as computer scientists, bass guitarists, waterfalls in Canada, the 2012 Summer Olympic Games, etc. Furthermore, these knowledge bases contain facts about entities. For example, they know the field of research for scientists, the birth date and the song albums of musicians, the location of waterfalls and mountains, and many others. Most of the knowledge bases mentioned above represent entities with unique identifiers, and facts according to the RDF data model in the form of subject-predicate-object (SPO) triples. In Figure 2.1 we show two example entities and some of the facts related to them according to the Yago knowledge base.

Today's knowledge bases would not have been possible without the immense amount of knowledge in Wikipedia. This encyclopedia contains infoboxes with clean facts, free text which describes the entity in natural language, category system, links among the entities in Wikipedia, and other useful information. For example, DBpedia leverages the Wikipedia infoboxes to extract various facts about entities. DBpedia also

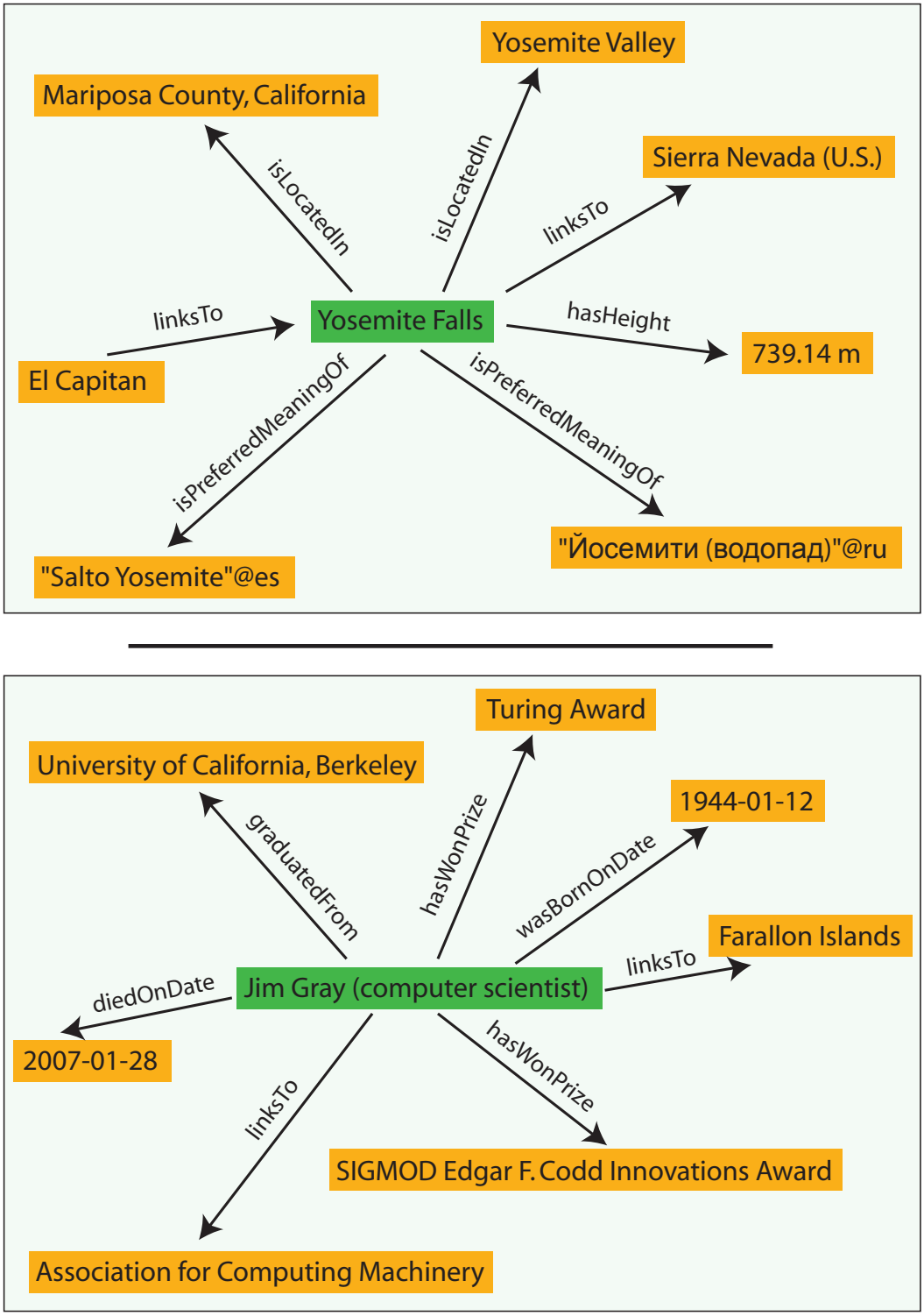

Figure 2.1: Example facts from Yago for Yosemite Falls and for the computer scientist Jim Gray.

provides links to other Web sources describing the entities. In addition to extracting facts from infoboxes, Yago integrates the class membership of entities (represented with Wikipedia categories) with the WordNet hierarchy. Yago2 [Hoffart et al., 2013], which is an extension of Yago, introduces three new dimensions for SPO facts, namely time, location, and contextual description in the form of keyphrases, which are extracted from the text of the Wikipedia articles.

The use and applications of knowledge bases is constantly growing. Knowledge bases have been used in various tasks related to Information Retrieval and Natural Language Processing. Some prominent examples include entity disambiguation and record linkage [Hoffart et al., 2011], question answering [Yahya et al., 2012; Lopez et al., 2007], query expansion [Bhogal et al., 2007], and machine translation [Knight, 1993]. Knowledge bases have been utilized also for automatic image annotation and classification. For example, ImageNet [Deng et al., 2009] leverages the hierarchy of WordNet to collect and classify images in different semantic classes. The use of medical ontologies like Medical Subject Headings, also known as MeSH (`nlm.nih.gov/mesh`), and SNOMED Clinical Terms (`ihtsdo.org/snomed-ct`), is of great importance, not only for patient care, but also for health research and analysis.

2.2 Visual Similarity between Images

Images showing named entities, like people or landmarks, which are retrieved using image search engines often contain many identical results. Since it is desirable to represent entities with a *diverse* set of images (e.g., from different time periods or from different perspectives), we need to diversify the images obtained from the search engines. Merely comparing the candidate images by their URIs does not always give satisfactory results. There are many identical images for a given entity with different URIs. Moreover, there are many near-duplicates, which for example have different sizes or illuminations, are slightly rotated, or are simply cropped. As a remedy, we exploit visual similarities in order to remove near-duplicate images and to produce a diverse set of result images for the entities of interest. For each pair of candidate images we estimate its visual similarity. To capture slight variations of the images in terms of scale, rotation or illumination, we extract local and global visual features of the images, and apply algorithms which compare them. We use SIFT and MPEG-7 feature descriptors, as well as approximate nearest neighbor estimation based on k-d trees and Best-Bin-First search. Finally, we check the correct geometric correspondence between the image features using RANSAC. In the following we explain these steps in more details.

2.2.1 SIFT-based Feature Descriptors

To estimate visual similarities between images we first extract a set of feature descriptors from each image. To this end we use the Scale-Invariant Feature Transform (SIFT) algorithm [Lowe, 2004] to detect and describe local features of a given image. The SIFT descriptors are specific for each image and are based on particular points of interest in the image. Moreover, they are known to be invariant under affine transformations and also robust to viewpoint changes or illumination variations. The SIFT

Figure 2.2: Example images with extracted SIFT feature descriptors in green.

features are represented by 128-dimensional vectors. In our work we extracted them using the IVT software library (`ivt.sourceforge.net`). In Figure 2.2 we give example images and their SIFT feature descriptors. The number of extracted features varies in different images from only a few features to hundreds of features.

2.2.2 Image Similarity with SIFT Feature Descriptors

To estimate if two images are visually similar we compare their SIFT feature descriptors. For each feature of the first image we find its nearest neighbor from the second image using Euclidean distance between the 128-dimensional feature vectors. We use k-dimensional trees (k-d trees) [Bentley, 1975] and Best-Bin-First search [Beis and Lowe, 1997] to find approximate nearest neighbors. First, we build a k-d tree with all features from one of the tested images. Then, for each SIFT descriptor of the other image, we search for its nearest neighbor in the k-d tree using the Best-Bin-First search algorithm. To reduce the number of searches we build the k-d tree using the image with more feature descriptors. Note, that sometimes the nearest neighbor of a

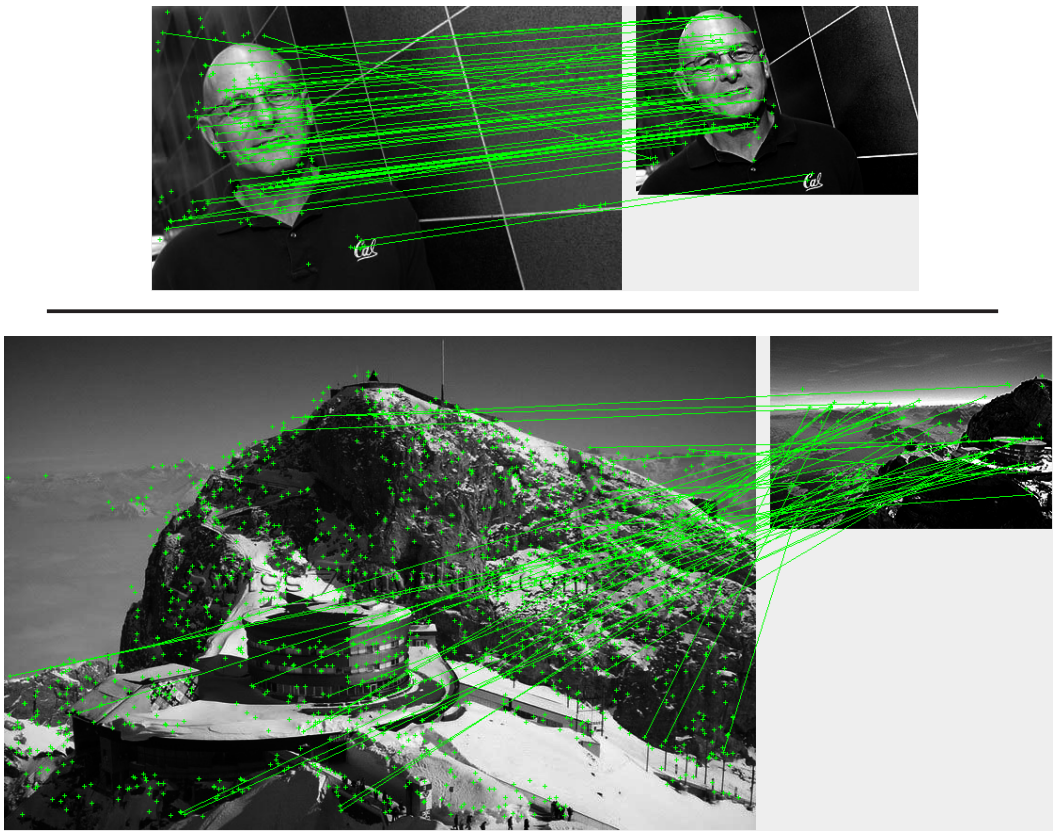

Figure 2.3: Example images with SIFT descriptors and their nearest neighbors computed with k-d trees and Best-Bin-First search (shown with green lines).

given feature in the k-d tree can be actually far away in the SIFT feature space. This is why, we use a threshold parameter $t_{sift-dist}$, and we check if the Euclidean distance between the feature and its nearest neighbor in the k-d tree is less than $t_{sift-dist}$. If the distance is smaller than the threshold, then we mark the two features as a pair. This means that we obtain a set of pairs $\{(x, x_{neighbor})\}$, where x is a descriptor from the first image, and $x_{neighbor}$ is the nearest neighbor descriptor of x from the second image within a distance $t_{sift-dist}$. In Figure 2.3 we show two pairs of images, their feature descriptors, and their pairs of nearest neighbors.

2.2.3 RANSAC

Finding the nearest neighbors of the feature descriptors in the images using Best-Bin-First search is not sufficient to decide if the images are visually similar. The reason is that this approach works only at the feature level and hence feature correspondences can refer to different objects in the images. In Figure 2.3 the second example shows a large number of feature pairs although the images are clearly dissimilar. To solve this inconsistency, we need to verify *geometrically* if the images are similar. To this end we find the *best* affine transformation between the two images using the pairs of

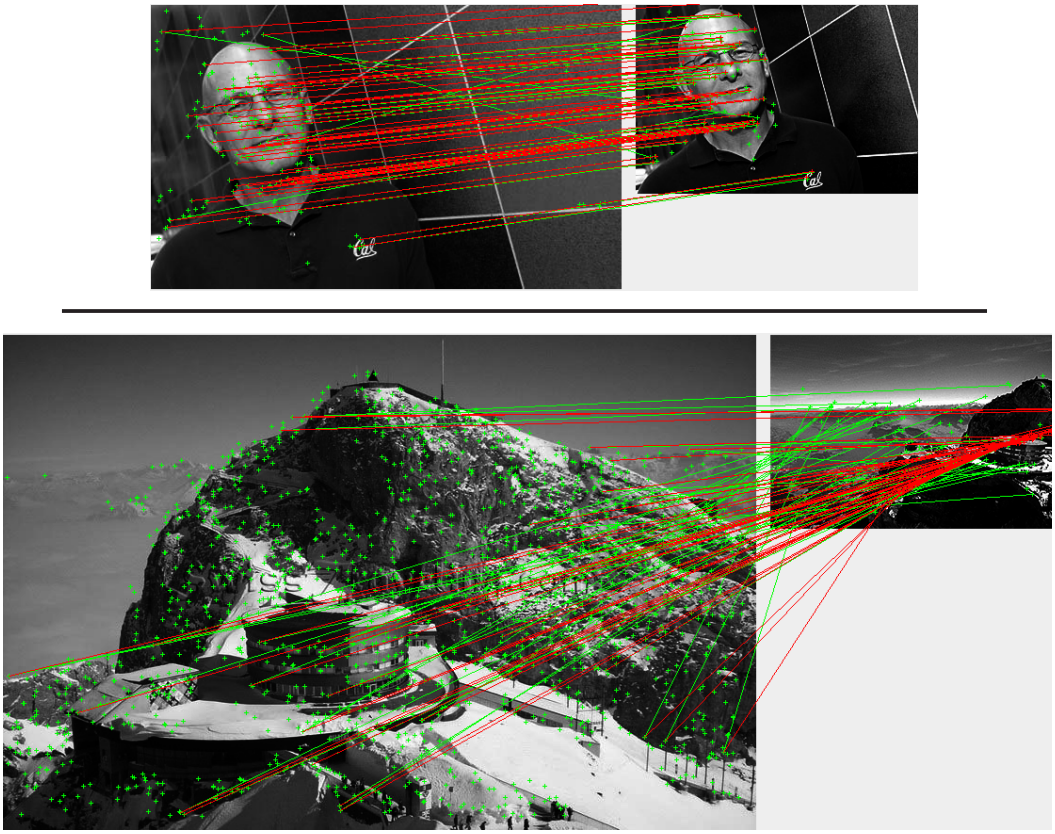

Figure 2.4: Nearest neighbors of SIFT descriptors, computed with k-d trees and Best-Bin-First search, are shown with green lines. The feature pairs according to the best affine transformation found with RANSAC are shown in red.

nearest neighbors, and check how many pairs satisfy this transformation. We define the best affine transformation to be the transformation which satisfies the largest number of pairs of nearest neighbors across many candidate transformations.

To find the best affine transformation we use the RANdom SAmple Consensus algorithm (RANSAC) [Fischler and Bolles, 1981]. RANSAC is an iterative method, which estimates parameters of a given model using data with a large number of outliers. To represent affine transformations, we use homogeneous coordinates of the SIFT features and a 3×3 matrix, in which the last row is (0,0,1). This means that there are 6 unknowns in the matrix. Each nearest neighbor pair found with the Best-Bin-First search leads to 2 equalities. Hence, to estimate the parameters of the matrix we need 3 pairs. At each iteration, RANSAC chooses 3 random pairs of nearest neighbors, and estimates the matrix parameters. According to each matrix, we compute the number of nearest neighbor pairs which satisfy the affine transformation with respect to a specified threshold t_{dist}. In more details, let $(x, x_{neighbor})$ be a pair of a feature and its nearest neighbor according to the Best-Bin-First search. Let x_{transf} be the affine transformation of x according to the current matrix. If the Euclidean distance between x_{transf} and $x_{neighbor}$ is smaller than t_{dist}, then the pair

$(x, x_{neighbor})$ satisfies the current matrix. At each iteration of RANSAC we compute the total number of pairs that satisfy the current matrix. After a number of iterations, we choose the matrix with largest number of pairs that satisfy it. This is our best affine transformation.

Finally, using the best affine transformation found with RANSAC, we decide whether the two tested images are visual duplicates. If the number of pairs that satisfy the best affine transformation is above a certain threshold t_{num}, we conclude that the images are visually similar. Alternatively, we could compute the percentage of all feature pairs found with Best-Bin-First search which satisfy the best affine transformation, and use another threshold parameter.

In Figure 2.4 we show two pairs of images. The first pair shows two visually similar images, which is also proved by the number of feature pairs that satisfy the best affine transformation estimated with RANSAC: they were 60 out of 64 feature pairs. For the second example, in which the images are not visual duplicates, there were 5 pairs which satisfied the best affine transformation, out of 46.

2.2.4 Filtering with MPEG-7 Feature Descriptors

To extract SIFT-based feature descriptors and to check whether two images are visually similar according to their SIFT features requires expensive computations. This is why we perform a filtering step beforehand, by using MPEG-7 global feature descriptors [Salembier and Smith, 2001]. We use the Edge-Histogram and Scalable-Color descriptors to identify those images that have high differences in these two descriptions. In this way we would not perform similarity checks based on SIFT features for clearly dissimilar images. We extracted MPEG-7 features using the Lire software library (`semanticmetadata.net/lire`).

2.2.5 Parameter Assignments

The algorithms described above need to be set up with a number of different parameters. To adjust all parameter values we used a set of example images of people and landmarks with many near-duplicates. For SIFT feature extraction with the IVT library we used quality threshold of 0.008. The number of k-d tree leaves was set to 150, and the threshold for feature similarity using k-d trees and Best-Bin-First search was $t_{sift-dist} = 0.22$. RANSAC was run with 1000 iterations, and the thresholds t_{dist}, and t_{num} were set to 3 and 9, respectively. According to our training data, if the Edge-Histogram was larger than 170 or the Scalable-Color was larger than 350, then the images were always dissimilar. In this case we did not test for visual similarity using SIFT features. To avoid incorrect judgments for image dissimilarity in the test data, we further increased the threshold parameters for the two MPEG-7 descriptors with 10%.

2.2.6 Similarity Function

Finally, we define a binary function $sim(p_i, p_j)$ for images p_i and p_j which returns 1 if the images are visually similar, and 0 otherwise. The computation of $sim(p_i, p_j)$ is as follows. Since image comparisons based on SIFT and MPEG-7 descriptors are

computationally expensive, we start with two simple comparisons. We compare the URIs of p_i and p_j. If they are the same, then $sim(p_i, p_j)$ is set to 1. If the URIs are different, we continue with a comparison of the SHA1 hash values of the images. If the images have the same hash values, then $sim(p_i, p_j) = 1$. If the hash values are different, we test p_i and p_j for MPEG-7 similarity. If the images are clearly dissimilar according to the thresholds of our two MPEG-7 descriptors from above, then $sim(p_i, p_j) = 0$. Otherwise, to verify image similarity we apply the SIFT-based similarity test followed by the RANSAC algorithm.

Chapter 3

Related Work

This thesis proposes solutions mainly to two problems, the population of a knowledge base with images of named entities, and the extraction of text contents related to given entities of interest. The work which is most related to these problems includes *Population of Ontologies with Images* and *Content Enrichment* among others. In the following we give a detailed review of the related works.

3.1 Image Retrieval and Analysis

Content-based image retrieval (CBIR) has been investigated very extensively during the last decade. [Datta et al., 2008] provide a survey on the ideas, trends, and topics in this field of research. CBIR aims at analyzing and organizing digital pictures by their visual content. Sub-tasks of content-based image retrieval include: estimating visual similarities between images, automatically annotating images, classifying or clustering images by their content, illustrating stories, and many more.

Internet image search engines, such as Google image search or Bing image search, index and retrieve images primarily by keywords, image captions, and other non-visual context that surround a picture on a Web page[1]. But they do provide options for smart processing like removing visually similar images to increase the diversity of the results, filtering the results by type (e.g., person faces), showing the highest quality pictures first in the result lists, and many others. Recently, Google image search has also started organizing image results by subject (e.g., images for Julia Roberts can be further organized by "Julia Roberts pretty woman", "Julia Roberts young", etc.). However, this feature works well only for celebrities; it does not work for lesser known entities or such with ambiguous names. Another recent feature of Google is the search by image, in contrast to posing a text query. The goal is to retrieve results, which are visually similar to the query image or alternatively to recognize the objects/places/people on the picture[2]. Note, that details for the methods or heuristics that the search engines use are not publicly available.

In the following we present a more detailed overview of the work related to the approaches for finding images of entities developed in this thesis.

[1] `stonetemple.com/articles/interview-peter-linsley.shtml`

[2] `stonetemple.com/search-algorithms-with-google-director-of-research-peter-norvig/`

3.1.1 Population of Ontologies and Object Classes with Images

Recently, a number of projects have started populating object classes or existing knowledge bases with representative images: [Schroff et al., 2007], TinyImage [Torralba et al., 2008], Optimol [Li et al., 2007a], LabelMe [Russell et al., 2008], [Yao et al., 2007], [Zhang et al., 2012], ImageNet [Deng et al., 2009], and Multipedia [García-Silva et al., 2011].

In [Schroff et al., 2007, 2011] a large number of images for a specific object class is generated (for example, penguins, sharks, airplanes, etc.). A multimodal approach which uses text, metadata and visual features is developed to gather and rank high-quality images from the Web. TinyImage [Torralba et al., 2008] is a dataset of low resolution images collected from the Internet by sending all nouns in WordNet [Fellbaum, 1998] 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. Optimol [Li et al., 2007a] collects images for object classes (e.g., panda) using image search engines and a few seed images. It incrementally learns a class model by using object recognition techniques. LabelMe [Russell et al., 2008; Torralba et al., 2010] 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 (incl. bounding boxes and polygons). Similarly to LabelMe, [Yao et al., 2007] have developed a labeling framework with rich representations for scene-level geometry, object segmentations and decompositions, and local geometric features. In [Zhang et al., 2012] a large dataset, called "Celebrities on the Web" is constructed. It contains 2.45 million distinct images of 421 436 celebrities.

ImageNet [Deng et al., 2009] is one of the closest projects to our work. Unlike the projects sketched above, ImageNet addresses the problem of integrating photos into a knowledge base with formalized entities and types, namely, WordNet [Fellbaum, 1998]. It builds a large-scale labeled image collection based on the taxonomic hierarchy of WordNet. To this end, ImageNet first collects candidate images for each synset (i.e. synonym set) in WordNet by querying image search engines with the synonyms in the synset. To collect as many images as possible, ImageNet expands the queries for each synset by appending them with descriptive words from parent synsets. Then, the candidate images are cleaned using Amazon Mechanical Turk where the users are provided with a set of candidate images in a given synset and the definition of the synset, and they are asked to verify the correctness of each image. The difference between ImageNet and our work in this thesis is that while ImageNet focuses on finding images of semantic classes such as towers, churches, etc., our work addresses photos of *individual entities* such as the mountain Siula Grande, the scientist Jim Gray, the Blue Mosque in Istanbul, etc.

Multipedia [García-Silva et al., 2011] enriches DBpedia [Bizer et al., 2009] with images of ontology instances retrieved from image search engines. This work has been published after our work presented in Chapter 4 and Chapter 5; nevertheless it is highly related as it solves the same problem of populating a knowledge base with images of entities. For each DBpedia instance, Multipedia uses the Wikipedia corpus

to gather context information. To this end it considers the words that appear around Wikipedia links which represent a mention of the instance. The class name of the instance is also included in the set of context words. Multipedia uses the context words to pose expanded queries to image search engines. In addition, it leverages the tags assigned to the retrieved images and it computes a semantic relatedness score between the tags and the context of the instance. Finally, the candidate images are ranked using an aggregation based on Borda's count [Saari, 2000]. This approach is evaluated on a set of ambiguous entities, where the entities are automatically selected from the Wikipedia disambiguation pages and are without a dominant meaning in Wikipedia.

3.1.2 Analysis of Text, Metadata, and Visual Information of Images

Other projects such as [Crandall et al., 2009; Serdyukov et al., 2009; Yagnik and Islam, 2007; Quack et al., 2008; Chen et al., 2012] pursue the dual goal of semantically organizing and interpreting a set of images by analyzing their metadata or visual information.

[Crandall et al., 2009] present techniques to automatically identify places shown on photos, using correlations between photos with GPS metadata and tagged but GPS-less photos on Flickr. The authors develop classification techniques for predicting these locations from visual, textual, and temporal features. Similarly, [Serdyukov et al., 2009] develop methods for placing photos, uploaded to Flickr, on the World map. They use a language model based on the picture annotations and they show how to incorporate GeoNames (`geonames.org`), which is a large database of locations. [Quack et al., 2008] propose an unsupervised learning approach to structure, interpret, and annotate large image collections. Geotagged photos are clustered into potentially interesting entities and events. Each cluster is assigned with text labels, which are then used to map the clusters to Wikipedia articles. For this mapping the approach critically relies on the availability of images in the Wikipedia articles for verification. [Yagnik and Islam, 2007] present a consistency learning paradigm to address the problem of learning face models for people names from weakly labeled training set. The resulting system learns different variations of face models (e.g., variations in age, expressions, makeup, etc.) for a large set of celebrities. In addition to visual or textual features of images, semantic hierarchies and ontologies (e.g., WordNet) have also been used to enhance the automatic classification and annotation of images: TinyImage [Torralba et al., 2008] and ImageNet [Deng et al., 2009], discussed above, [Chen et al., 2012], and others. In [Chen et al., 2012] object recognition together with a guide ontology is used to understand and represent images as an object relation network.

3.1.3 Diversification of Images

Projects such as [van Zwol et al., 2008; van Leuken et al., 2009; Kennedy and Naaman, 2008] aim at diversifying image results. In [van Zwol et al., 2008] a *topical (textual) diversification* of images is presented, where a retrieval model which incorporates image tags is developed. In [van Leuken et al., 2009] a *visual diversification* technique is presented. There clustering methods together with a dynamic weighting

function of visual features are developed. Representative images from each visual cluster are chosen to form a diverse result set. [Kennedy and Naaman, 2008] consider the combination of textual tags, location metadata, and visual features to apply clustering methods that generate diverse and representative image search results for landmarks.

All of the above projects exploit some form of semantic information about images to provide automatic annotation tools, and improve data retrieval and the organization of image collections. However, with the exception of Multipedia, discussed above, none of them pursues the integration of photos of individual entities into knowledge bases with formal notions of typed entities and relational facts. In Chapter 4 and Chapter 5 we show how to populate automatically a knowledge base with diverse sets of images of different types of people and landmarks.

3.2 Entity Difficulty for Image Retrieval

In Section 5.5 we present an approach for estimating the difficulty of retrieving good images for given entities. This approach allows us to selectively re-rank the search engine's results only when it is likely to improve them.

Query difficulty estimation has been an established problem in Information Retrieval [Carmel and Yom-Tov, 2010; Carmel et al., 2006; Hauff et al., 2009; Shtok et al., 2012; He and Ounis, 2004; Cronen-Townsend et al., 2002]. Its goal is to estimate and predict the quality of the search results when no relevance feedback is given. There are mainly two types of prediction approaches: *pre-retrieval* and *post-retrieval* approaches. The pre-retrieval approaches estimate the quality of the search results before the search takes place. They include mostly linguistic and statistical methods. In contrast, post-retrieval approaches analyze the top search results. Some of the well-studied methods focus on: (1) the Kullback-Leibler divergence between the language model of the returned results and the language model of the entire document collection; (2) the robustness of the results when there is a query or document perturbation; (3) the score distribution of the search results; and many others. However, all of these methods are proposed specifically for text queries and for text search results. There has been less interest in *predicting the quality of image search results*. In essence, our approach from Section 5.5 is a post-retrieval approach which analyzes the top image search results.

In [Li et al., 2012] an approach for query difficulty estimation for image retrieval is proposed. It employs methods developed originally for text queries to images represented by bags of visual words. Although this work considers image retrieval, its goal is very different than ours: (1) the query is an image, and (2) the results are considered relevant if they are visually similar to the query image. In our work, the query is a text query, and we consider as relevant all images that show the entity of interest (but not necessarily having similar visual content).

In [García-Silva et al., 2011] the problem of entity difficulty is discussed as well. The authors focus on ambiguous entities without a dominant meaning. To automatically decide if an entity is of this type, they analyze the frequency of each distinct

meaning of the entity in Wikipedia, where the entity meanings are extracted from the corresponding Wikipedia disambiguation page. However, there is no consideration of ambiguity or meaning dominance on the Web as opposed to Wikipedia, and there is no consideration of images.

3.3 Entity Search

Entity search has become an established part of Information Retrieval, 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 (LMs) 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 (or equivalently, by distance measures like Kullback-Leibler divergence) (e.g., [Balog et al., 2009; Fang and Zhai, 2007; Petkova and Croft, 2007]). In all these settings, entities are the output of a query, the query itself is standard keyword search. This is different from our problem where we start with an entity (given by its name and a short description or facts from a knowledge base). 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 [Serdyukov et al., 2008; Jing and Baluja, 2008]. However, these methods improve result quality only for prominent entities; random walks do not work well for entities in the long tail.

3.4 Keyphrase Analysis

Keyphrase extraction and matching are key steps of our approach for finding images of named entities presented in Chapter 5. In the following we review the work related to these two problems.

3.4.1 Keyphrase Extraction

There are both supervised (e.g., [Frank et al., 1999; Brook Wu et al., 2006; Jiang et al., 2009]) and unsupervised (e.g., [Kumar and Srinathan, 2008; Hofmann et al., 2009; Mihalcea and Tarau, 2004]) approaches for keyphrase extraction. 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 [Chakrabarti et al., 2011] the problem of entity tagging is considered. A set of descriptive phrases (referred to as entity tags) is associated with a given entity by leveraging a collection of Web documents that contain information about the entity of interest. First, a set of candidate tags is extracted from the documents using specific lexical patterns. Then, the candidate tags are associated with the entities of interest by considering textual proximity between the tags and the entities

in the documents. This method has the same goal as ours, namely to associate characteristic keyphrases to entities. The main difference with our work is that, instead of a document collection, we use an entity description from which we extract keyphrases.

In our work, we adapted an unsupervised approach for keyphrase extraction to avoid training bottlenecks and for domain-independence. We used noun phrases, extracted from the entity descriptions, with ranking based on the Mutual Information measure, as described in Section 5.3.

3.4.2 Keyphrase Matching

To match a given keyphrase in a document text we can use exact or partial matching. However, in practice exact matching is very limiting and often unrealistic. Furthermore, partial matches of entity-specific phrases can be still very good cues for the relevance of the documents. This is why, in our work we use partial matching of phrases.

Proximity-aware scoring for standard keyword search [Tao and Zhai, 2007; Cummins and O'Riordan, 2009; Büttcher et al., 2006; Schenkel et al., 2007; Song et al., 2008; Svore et al., 2010] considers the proximity of the query keywords in a result document. The purpose is to enhance the scoring of keyword search. Our goal is different in that we aim to match a given phrase in a document. Nonetheless, we adapt and extend the above techniques and adjust them to our setting, as described in Section 5.4.

In [Agrawal et al., 2009b] the extraction of all mentions of given entities from a document is an intermediate step. The authors reduce this problem to a multi-pattern matching problem and use the Aho-Corasick algorithm for exact matching [Navarro and Raffinot, 2002]. A partial match of entity names is also considered. The proposed approach first identifies "synonyms" of the entities in the reference set, following [Chaudhuri et al., 2009], and then applies again exact matching but now on the enhanced set of entity names.

3.5 Extraction of Text Contents

In Chapter 6 we present an approach for extracting text contents from candidate documents, which are highly related to a given entity. The goal is to enrich the knowledge about the entity. In the following we review some of the work related to this task.

3.5.1 Content Enrichment

Content enrichment, also referred to as document expansion, is the task of extending a given text with more related content. This task has various applications: question answering, information retrieval, entity disambiguation, fact extraction, and others. Recently, TREC has introduced the new challenge of *Knowledge Base Acceleration* (`trec-kba.org`): filter a time-oriented corpus of documents that are highly relevant to a given list of entities. The main application for this track is to help human

contributors of knowledge bases to maintain knowledge about entities in a timely manner. The goal is to provide the contributors with recommendations about salient facts or related documents, which should be considered for updates in the knowledge base. Our goal in Chapter 6 is very similar to this challenge.

[Schlaefer et al., 2011] present a source expansion algorithm. A given collection of documents, where each document represents a single topic, is extended with related contents from the Web. First, a set of candidate paragraphs is extracted based on HTML markup. These paragraphs are then ranked using Logistic Regression with various features, including topical, search, and surface features. The expanded corpus is used to enhance the quality of a question answering system, which means that the output is further processed by machines without space constraints. In contrast, the goal of our work from Chapter 6 is to extract related contents, which can be used as recommendations for contributors of knowledge bases. These recommendations need to be concise without unnecessary repetition to avoid overwhelming the authors with too much information. Furthermore, in our work we consider (1) no markup-dependence on the input text, (2) novelty with respect to the seed text, (3) diversification of the expanded content, and (4) independence of training data. All of these issues are not considered in [Schlaefer et al., 2011].

Similarly, [Efron et al., 2012] propose an algorithm for improving information retrieval of "short texts" through aggressive document expansion. Each short text is submitted as a pseudo-query in a large corpus. The obtained results are used to enhance the language model of the initial short document. The presented experiments are based on microblog data as a source for short texts.

The goal in [Mihalcea and Csomai, 2007; Agrawal et al., 2012] is to extract key concepts from a given text, which are then linked to external sources like Wikipedia. In both works, the key concepts are in the form of short keywords and keyphrases. In [Mihalcea and Csomai, 2007] the authors first identify important keyphrases in a text (also referred to as keyphrase extraction), and then link these phrases to their corresponding Wikipedia pages (also known as word sense disambiguation). Similarly, [Agrawal et al., 2012] present a framework for enriching textbooks with relevant content from the Web. The authors algorithmically identify sections in textbooks which can be extended. To expand the content, they extract key concepts from the text, which they augment with authoritative articles (e.g., articles from Wikipedia). In addition, augmentation with images is also considered. Our goal is not to link key concepts or keyphrases to existing articles in Wikipedia, but to collect related information for the entity of interest from the Web. In many cases such information is not available in Wikipedia.

Remotely related is the work of [Leong and Cucerzan, 2012] where the objective is to automatically retrieve supporting evidence from the Web for factual statements. The proposed system enriches Wikipedia facts with supporting external links, comparable to those manually selected by the Wikipedia authors.

3.5.2 Passage Retrieval and Text Segmentation

The goal of passage retrieval is to extract passages which are relevant for a given query. There are mainly two approaches. First, during passage retrieval there is no

knowledge about the query: [Callan, 1994; Salton et al., 1993; Schlaefer et al., 2011]. Passages are extracted based on sentences, paragraphs, HTML tags, n-grams, etc. Only then, the retrieved passages are evaluated using standard retrieval methods (e.g., language models) for their relevance to the query. Second, during passage retrieval there is prior knowledge about the query: [Clarke et al., 2001; Kaszkiel and Zobel, 2001; Li et al., 2007b]. In [Clarke et al., 2001; Kaszkiel and Zobel, 2001] the passage locations are fixed after the query is evaluated, such that they have highest relevance to the query. Instead of using predefined passages, the authors analyze the shortest segments (covers) in the text which contain all query words. However, the queries in these works consist only of a few keywords, while in our work the seed text can be of arbitrary length. In [Li et al., 2007b] passages are extracted by first assigning to each word a probability score, which depends on the query, and then selecting sequences of words with high scores. The probability scores of the words depend only on the words themselves; the surrounding words are not considered.

In our work from Chapter 6 to extract parts of an input text which are relevant to a given entity, we use as an input the entity seed and we do not consider predefined passages, such as sentences or paragraphs.

Text segmentation is the task of retrieving parts of the input text, which are semantically coherent. Existing approaches divide the given text when there is a shift from one topic to another by using change in the vocabulary [Choi, 2000; Utiyama and Isahara, 2001; Hearst, 1997] or by using statistical topic analysis [Li and Yamanishi, 2003; Misra et al., 2009; Brants et al., 2002].

3.5.3 Text Summarization

Prior work on summarization [Nenkova and McKeown, 2011], and especially extractive summarization, is naturally related to our problem from Chapter 6 for extracting text contents related to entities. However, the fact that summaries are intended for human readers mandates that summaries consist of entire sentences. This is a fundamental difference to our knowledge-oriented setting where any text snippet (e.g., captions) and even semi-structured fragments (e.g., table rows) can contribute to valuable text excerpts. Nevertheless, the specific directions of multi-document summarization [Haghighi and Vanderwende, 2009; Harabagiu and Lacatusu, 2010; Wan and Yang, 2008], where diversity matters, and query-driven summarization [Conroy et al., 2006; Daumé and Marcu, 2006; Li et al., 2010], where thematic focus matters, are applicable to the problem of gathering related text contents. In our experimental studies, we capture the essence of many methods along these lines by extracting, ranking, and diversifying sentences and HTML paragraphs from the input documents, as described in Section 6.7.

3.5.4 Diversification of Results

Search results diversification [Carbonell and Goldstein, 1998; Agrawal et al., 2009a; Chen and Karger, 2006; Radlinski and Dumais, 2006; Gollapudi and Sharma, 2009; Clarke et al., 2008; Borodin et al., 2012; Drosou and Pitoura, 2010] has been an established problem in ranking algorithms for Web search. One early work in this

direction is the "Maximal Marginal Relevance" approach introduced in [Carbonell and Goldstein, 1998], which we exploit in our methods for extracting related contents.

[Agrawal et al., 2009a] propose a diversification objective which tradeoffs relevance and diversity. The approach aims at minimizing the risk of dissatisfaction of the user, given that there exists a categorical information of the queries and the result documents. In contrast to this work, we are not given with a categorical information about the queries and the documents. [Gollapudi and Sharma, 2009] develop an axiomatic approach to characterize different diversification functions and show a reduction to the facility dispersion problem [Ravi et al., 1994].

Chapter 4

Knowledge Kaleidoscope with Queries

4.1 Introduction

One way to populate a knowledge base with images of entities is to consider as a starting point the facts about the entities stored in the respective knowledge base. Our approach presented in this chapter uses standard image search engines to retrieve images. Then it utilizes entity facts which (1) enhance the retrieval of large amount of relevant pictures, and (2) provide a mechanism for checking if the retrieved images are relevant for the entity.

Motivation. Knowledge bases such as DBpedia, Freebase, or Yago are rich sources of facts about people, locations, organizations, sports events, etc. For example, they would know the Alma Mater of scientists and awards that they have won, or the location and architect of culturally important buildings (churches, temples, castles, etc.). However, these knowledge bases are still fairly sparse in terms of multimodal information about entities, like photos, videos, audio recordings, etc. Even if Wikipedia contains large amount of articles with images, these are mostly articles of prominent entities, like celebrities or famous landmarks. Entities, which are less prominent or which are in the "long tail" are often neglected. For example, as of February 2013, Wikipedia does not know how Raghu Ramakrishnan (currently Technical Fellow at Microsoft and previously Vice President and Research Fellow for Yahoo! Inc.) or Kesselkogel (the highest mountain in the Rosengarten group in South Tyrol, Italy) look like (see Figure 4.1).

On the other hand, photos and videos of people and landmarks have become abundant on the Internet. Web 2.0 portals such as Flickr and YouTube even offer extensive tags and metadata, but these are often noisy or incomplete, and sometimes wrong. Recently, various projects such as [Deng et al., 2009; Crandall et al., 2009; Schroff et al., 2011; Torralba et al., 2008; Yagnik and Islam, 2007] have started analytic mining of the tags, metadata, GPS coordinates, or visual features of images in order to improve the semantic organization of such data collections. However, with the exception of ImageNet [Deng et al., 2009], none of them addresses the integration of photos into knowledge bases with formalized notions of entities, types, and facts.

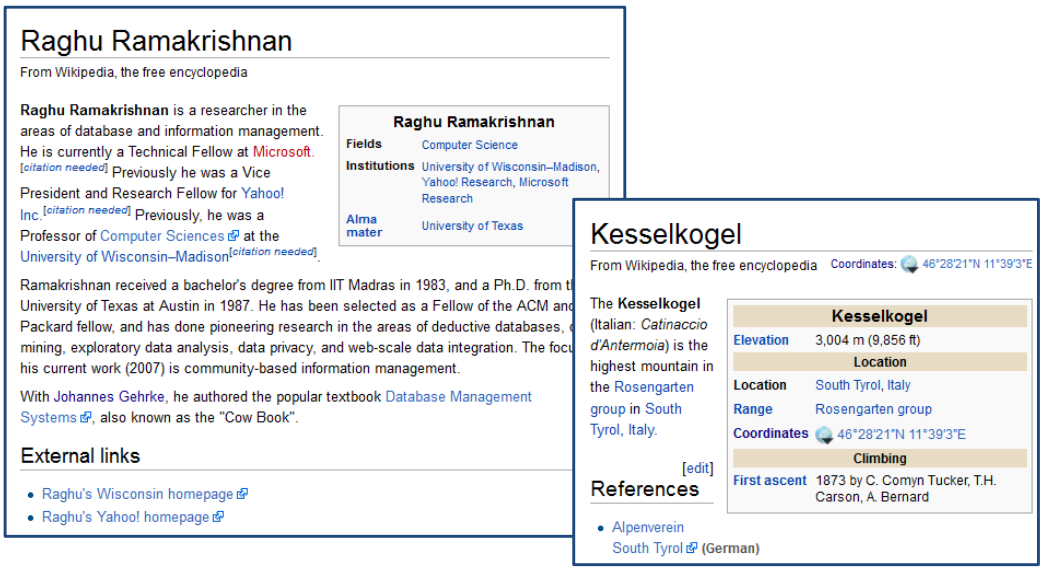

Figure 4.1: Examples for Wikipedia articles without pictures.

Challenges. In principle, it is not difficult to find photos of people or monuments using search engines like `images.google.com` or `images.bing.com` or searching `flickr.com` by tags. This works well for entertainment stars, important politicians, and tourist attractions. However, it remains difficult to find photos for entities in the long tail: lesser known but still notable people and places. Typically, a direct query with the entity name returns many photos with good results in the top ranks but quickly degrading precision with decreasing ranks. For a human user who knows the entity of interest, it may be good enough if the top-10 or top-20 contain a handful of correct photos, but this is insufficient for automatically enhancing a high-quality knowledge base.

In some cases, the ambiguity of the entity name dilutes the search engine results. An example is the Berkeley professor and former ACM president David Patterson. Most of the top-20 Google and Bing results do not show the target entity. Instead, they show the governor of New York from 2008 to 2010 (whose name is actually David Paterson). The results include a handful of correct photos, but it is difficult even for a human and extremely difficult for the computer to discriminate these from the photos of other people (with the same or very similar name). Another example is the Turing award winner John McCarthy. The top image results from the search engines are a mixture of correct pictures and pictures showing football players or referees with the same name. The same holds for Jim Gray, famous database researcher who disappeared at sea in 2007. The image search results include mostly pictures of a sportscaster with the same name.

Moreover, even for more prominent targets, it is desirable to have a diverse collection of photos (e.g., from different time periods). Such pictures might be rare and difficult to locate using search engines and querying with the entity name only.

None of the methods in the photo-mining projects mentioned above can solve

these problems. The closest project to our work is ImageNet [Deng et al., 2009], which enhances the WordNet thesaurus [Fellbaum, 1998] with photos. In contrast to our goal, however, the task there is to find representative photos of semantic classes such as towers, churches, mosques, cats, tigers, etc. There is no consideration on photos of individual entities such as the Five-Finger tower in Darmstadt, the Blue Mosque in Istanbul, etc.

Objective. Our goal is to automatically populate an existing knowledge base with photos of people and landmarks. We focus on people and places who are notable but not extremely prominent, or have ambiguous names. We aim at both high precision and high recall, so that quality measures like MAP (mean average precision) or NDCG (normalized discounted cumulative gain) are maximized across a large set of results for the same entity. Furthermore, we aim at gathering large amount of diverse photos for named entities. For example, we would like to collect pictures of people at different occasions or different ages, or pictures of landmarks from various perspectives or different light/weather conditions.

Approach and Contributions. Our approach constructs a set of expanded queries for each entity of interest, where the expansions are automatically derived from already known facts in a knowledge base. In our work we use the Yago knowledge base (`yago-knowledge.org`). For example, to find photos of the Berkeley professor David Patterson, we would use the hasAffiliation or worksInField relations of Yago and search for "David Patterson Berkeley" or "David Patterson computer science". These expanded queries are then posed to image search engines. The collected results are ranked based on merging the results from all query expansions, with specific weights for the different expansions. The weights are automatically learned from training samples. This approach can be seen as a form of (probabilistic) consistency checking of search engine results, as reflected in the overlap of the results for different expansions. In addition, we consider image-content similarities among different result candidates, using SIFT and MPEG-7 features, in order to enhance to visual diversity of the final results. Our experimental results demonstrate the high precision-recall quality of our approach. Our approach is further improved in case we consider visual similarity of images. Moreover, we show significant improvements of our methods over standard image search result lists.

The novel contributions in this work are the following:

- We show how to harness relational facts about named entities for gathering diverse images of the entities with high precision and high recall;
- We develop robust methods for estimating model parameters, so that our approach is applicable to a wide variety of different entity types;
- We integrate image-similarity computations for improving the final ranking of result photos and for gathering diverse set of images;
- We show experimental results, which demonstrate the high effectiveness of our approach as opposed to standard image search result lists.

Outline. The rest of the chapter is organized as follows. Section 4.2 presents the overall architecture of our system. Section 4.3 presents our scoring model and its training and ranking algorithms. In Section 4.4 we extend this scoring model to consider image similarities for an alternative ranking with improved diversity. Section 4.5 presents a regression model for our problem. Section 4.6 demonstrates our experimental results. Finally, Section 4.7 provides a summary of the chapter.

4.2 System Architecture

Entity Types. In our work, we consider named entities e of different types t, for example, scientists, politicians, buildings, mountains, etc. We assume that, for each type t, we have specific relations $R_i(subject\ e, object\ o)$, $i \in \{1, \ldots, m(t)\}$ populated in the knowledge base. These could be, for example, the affiliation, Alma Mater, and scientific field for scientists; the geographic areas (country, state, city) of activities and political positions held by politicians; the country and height of mountains and the person who climbed it first; and so on. We can use these facts to generate specific queries that we can send to image search engines or other services on the Internet.

Training Data. Some relation types are too specific and do not yield good result photos. For example, using the exact birth date of a politician does not yield good results, as many biographies on the Web do not contain this information. Conversely, some types of relations are too unspecific and thus can lose focus and dilute the topic. For example, the names of the songs for musicians often yield only pictures of the album cover. However, many relations help in finding correct images of the entity (e.g., the field of research for scientists, the range and location of a mountain, etc.). We learn how indicative a certain relation is for a given entity type by using training data for each entity type t: examples of photos and their URIs that correctly show a given entity, for a small set of entities. Note that in our experimental studies the target entities are disjoint from the training entities.

System Architecture. The overall system architecture is illustrated in Figure 4.2. It consists of the following *preprocessing* and *harvesting* components:

Preprocessing components:

- The *Query Expansion Generation* component obtains relational facts about entities from the knowledge base and generates different keyword queries from them. The queries always contain the original entity name as well.
- The *Data Gathering* component invokes queries on different photo search engines and retrieves the top-100 results for each query.
- The *URI Image Comparison* and the *Visual Image Comparison* components compare the pictures from the training data and the pictures retrieved by using the various query expansions generated on the previous steps. The URI-based image comparison utilizes only the URIs of the images. The visual image comparison is based on the visual content of the images (see Section 2.2).

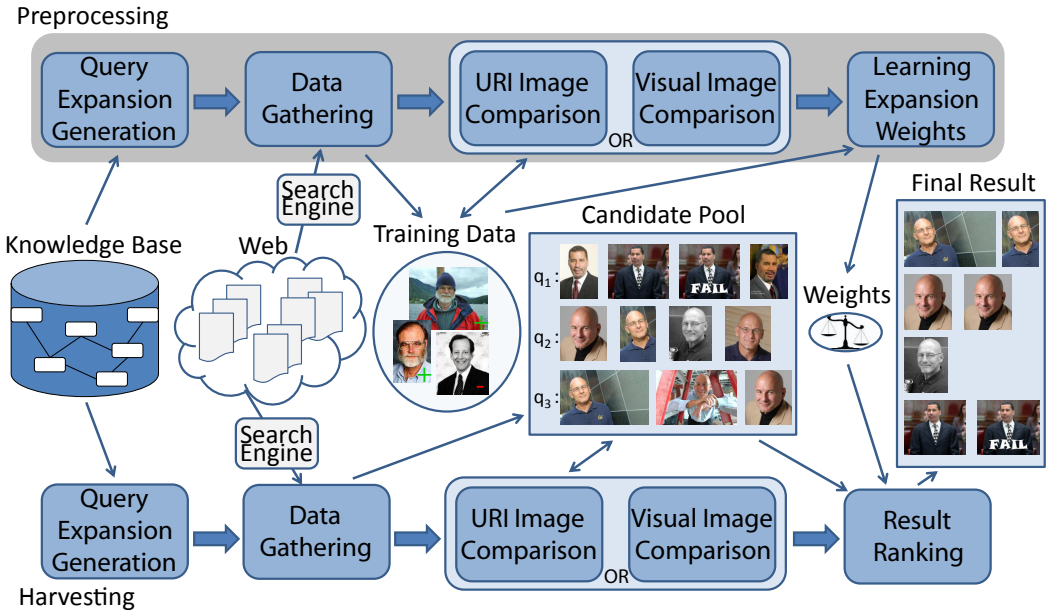

Figure 4.2: System architecture. Rectangles are system components. Thick arrows denote control flow and thin arrows show data exchange.

- The *Learning Expansion Weights* component assigns the best suitable weights to each relation type for each entity type. We compute two types of weights: (1) using URI image comparison, and (2) using visual image comparison. The computation of the weights is described in Section 4.3 and Section 4.4, respectively. These weights are later used to rank pictures of new entities.

Harvesting components:

- The *Query Expansion Generation* obtains relational facts about entities from the knowledge base and generates different keyword queries from them, similarly as in the preprocessing step.
- The *Data Gathering* invokes queries on different photo search engines and retrieves the top-100 results for each query, similarly as above.
- The *URI Image Comparison* and the *Visual Image Comparison* components compare the pictures retrieved on the previous step to detect duplicates or near-duplicates (using URI comparison or visual similarity testing).
- The *Result Ranking* component applies the ranking models from Section 4.3 and Section 4.4 to rank images for new entities. Depending on the type of image comparison, we use the respective type of relation weights and ranking procedure. In case of ranking with visual similarity, duplicate and near-duplicate images are grouped into equivalence classes as described in Section 4.4. We show only one representative picture per group and thus enhance the diversity

of the final results. Finally, the best pictures are added to the knowledge base, along with information about their provenance (based on our scoring model).

4.3 Ensemble Voting Model

The easiest way of obtaining pictures for a given entity (person or landmark) is by using the entity's name to issue a query to an image search engine. However, the results with this approach are often unsatisfactory. Even if good results appear on some of the top ranks, the entire ranking, say the top-100 results, is noisy and contains a significant number of incorrect photos or near-duplicates (although more and better photos exist at much lower ranks). We exploit the knowledge base to issue a variety of meaningful query expansions, each separately, and then analyze the results and rankings of different queries for agreement.

Our approach can be seen as an *ensemble voting method* to arrive at a consistent ranking of the entire pool of retrieved photos. The ensemble consists of different queries $q_1(e), q_2(e), \ldots, q_m(e)$ about the entities of interest. Query $q_1(e)$ is always the name of the entity. The rest of the queries are generated from specific relations that the knowledge base has for the given entity type $t(e)$. We retrieve the respective facts from the knowledge base and generate expanded queries with them. These queries always contain the entity name as well. We discriminate entities into types like scientists, politicians, buildings, mountains, etc. Interesting relations for generating queries are birth date, affiliation, Alma Mater, field of research, contributions, political party, location, range, elevation, and so on. Different entity types should favor different relations even if they were applicable uniformly, for reasons explained below. For example, to retrieve photos for the computer scientist David Patterson, we generate the following queries:

$q_1(e) =$ *David Patterson*
$q_2(e) =$ *David Patterson computer science*
$q_3(e) =$ *David Patterson U.C. Berkeley*
$q_4(e) =$ *David Patterson RISC*
$q_5(e) =$ *David Patterson RAID*

In principle, the queries q_2 through q_m would only yield subsets of the results that we obtain from the simple name query q_1. However, the results exhibit significant differences in their rankings. As search engines often return hundred thousands of results, we can practically access only top-ranked subsets of the query results, so that virtually no two queries show any subset-superset relationship. Therefore, photos returned by the top-100 of many queries for the same entity are more likely to yield more correct matches, compared to using only the query q_1.

Each query expansion assigns high ranks to photos from Web pages where the query keywords appear prominently and close to the photos. Although this is an oversimplified view of how modern image search engines work, it reflects the essence of their ranking criteria. Thus, accepting a photo if and only if multiple queries *agree* on the photo being relevant can improve the precision of the overall result set.

Figure 4.3: Intuitive illustration for ensemble voting.

Each query *"votes"* for a photo, and receiving many votes indicates a better result. Figure 4.3 illustrates this idea using 3 expanded queries for David Patterson.

4.3.1 Binary Voting

With each photo p in the union of the result images (actually the top-k prefixes of the result lists that we retrieve) of queries $q_i(e), i \in \{1, \ldots, m\}$ for entity e, we associate indicator variables $X_i(p)$ set to 1 if p occurs in the result of query $q_i(e)$, and 0 otherwise. Then the *voting score of a photo p with regard to entity e* is computed by the aggregation:

$$s(p, e) = \sum_{i=1..m} X_i(p)$$

We compare p with all other pictures in the lists q_i using simple URI comparison (a more sophisticated visual similarity comparison between pictures is described in Section 4.4). The score $s(p, e)$ is an aggregation over all duplicate occurrences.

On first glance, it seems that this method merely helps improving the precision of the overall results by simple ensemble voting. However, it can also improve recall and diversity of the results for a given entity. The reason is that we are not able to retrieve the complete result for a given $q_i(e)$ from any of the big search engines. Thus, running different queries whose results have very different ranks in different queries allows us to fetch a wider variety of photos at affordable cost.

4.3.2 Weighted Voting

Not all of the possible query expansions $q_i(e)$ have good yield. Some are overly specific and thus return too few results. An example would be adding the exact birthday of a person to the person's name; there are not that many biographies on the Web that have this information and at the same time contain a good photo. Other examples include the doctoral advisor or students for scientists, the exact coordinates

of a landmark or mountain, etc. Other query expansions are too unspecific, lose focus, and are susceptible to topic drifting. For example, expanding a musician's name with names of songs or albums may return mostly photos of the album cover. On the positive side, however, many expansions help in focusing the photo search. For example, searching for computer scientists who wrote popular text books (e.g., Hinrich Schütze, Hanan Samet, Soumen Chakrabarti, etc.) simply by person names often returns mostly book covers. This and similar problems may be overcome by query expansions that add the affiliation, an important award, or similarly salient facts about the person of interest.

The variability in the precision and recall of different query expansions is taken care of by giving different *weights* w_i to the various queries $q_i(e)$ in our voting scheme. It is straightforward to extend our approach into a *weighted voting score*:

$$s(p,e) = \sum_{i=1..m} w_i X_i(p)$$

The weights in this scheme could be the same across all entity types or specifically chosen for each type. The latter is more powerful and indeed advantageous for our scenarios. For example, while the birth year is a beneficial expansion for scientists, it is not nearly that helpful for musicians.

4.3.3 Learning Query Weights

The proper weights for a given entity type can be learned from explicitly labeled training data. We assume that we have at least a few correct photos of a few entities, for each type. These may be celebrities or famous landmarks where photos are ample (incl. photos in Wikipedia), or less prominent entities with photos on the Web. The test cases for our calibrated model would then be different entities, most of which are less prominent.

We estimate the query-specific weights w_i for a set T of training entities of type t, each with a ground-truth set of correct photos $P(e)$ and query results $Q_i(e)$ for query expansion $q_i(e)$, by:

$$w_i = \frac{1}{|T|} \sum_{e \in T} \frac{|Q_i(e) \cap P(e)|}{|P(e)|}$$

The weights w_i do not reflect the true fraction of correct photos retrieved by query q_i because we do not have ground-truth labels for all photos in the result set of q_i. The w_i values reflect the relative recall of the various query expansions.

4.3.4 Rank-based Weighted Voting

A final piece of information that we can exploit in the scoring function is the fact that Internet search engines return ranked lists rather than result sets. Photos at higher ranks are usually better matches, with a higher likelihood of really showing the entity of interest. This is a reasonable postulate regardless of our treating search engines as black boxes. It suggests moving from binary voting to *rank-based voting*, with the same query-specific weighting. Let $r_i(p)$ denote the rank of photo p in the

result of query q_i. The ranks $r_i(.)$ are numbers 1, 2, etc., with low numbers denoting high ranks. In case q_i does not contain p in its top-k results, $r_i(p) = k+1$. The score of p should decrease with the value of $r_i(p)$, which leads to rankings based on the following scoring formula for result pools gathered by retrieving the top-k results of each q_i:

$$s(p,e) = \sum_{i=1..m} w_i \frac{k+1-r_i(p)}{k}$$

We compare p with each picture in the lists q_i for URI identity. The score $s(p,e)$ is a weighted sum over all duplicate occurrences, considering weights for each query expansion and ranks of the individual pictures.

4.4 Voting Model with Visual Similarity

Query result lists for entities may contain many duplicate or near-duplicate photos. Since one of our goals is to find rankings of diverse images, we need a way to capture similarity or identity of photos. Merely comparing result images by their URIs sometimes does not give satisfactory results. There are many identical photos for a given entity with different URIs. Moreover, there are many near-duplicates that have, for example, different sizes, slightly different illuminations, or are simply cropped. As a remedy, we exploit visual similarities in order to remove near-duplicates and produce a better *diversity-aware ranking* of the images.

To estimate if two pictures are visually similar we use the approach from Section 2.2. We use visual similarities in two different phases of our scoring model: during learning of query expansion weights and in the final result ranking step, in which we also remove all near-duplicates.

4.4.1 Learning Query Weights with Visual Similarity

We estimate query-specific weights w_i for a set T of training entities of type t, each with a ground-truth set of correct photos $P(e)$ and query results $Q_i(e)$ for query expansion $q_i(e)$. The weights w_i are estimated by checking how many of the images in $Q_i(e)$ are similar to the images of the ground-truth set $P(e)$. More formally:

$$w_i = \frac{1}{|T|} \sum_{e \in T} \frac{\sum_{p \in P(e)} \sum_{x \in Q_i(e)} sim(x,p)}{|P(e)|}$$

where $sim(x,p)$ is a binary function, which returns 1 if x and p are visually similar images, and 0 otherwise. The function $sim(.)$ is defined in Section 2.2. This way we boost the weights for "good" relations, which find photos that are similar to those in the ground-truth set.

4.4.2 Rank-based Weighted Voting with Visual Similarity

With the similarity-enhanced weights from above, we can compute the ranked results for a new entity as outlined in Section 4.3. However, we can further enhance this ranking into a potentially better one by the following procedure. For each photo p in

the union of result images from the queries $q_i(e)$ for entity e we compute its voting score by the aggregation:

$$s(p,e) = \sum_{i=1..m} w_i \left(\sum_{x \in Q_i(e)} sim(x,p) \frac{k+1-r_i(x)}{k} \right)$$

where k is the number of results in $q_i(e)$ and $r_i(x)$ is the rank of photo x in $q_i(e)$. This way we give high ranks to those images that have many near-duplicates in the result lists across all queries.

However, it is not sufficient only to compute enhanced voting scores of the candidate images. In addition we need to remove all near-duplicates, so that the final list of images contains only visually diverse results.

4.4.3 Grouping of Visually Similar Images

To obtain a visually diverse results, we develop an algorithm for grouping similar images. We group images into equivalence classes and assign to each class a representative image. We compute ranking scores for the representative images by aggregating

Algorithm 4.1 Group Visually Similar Images

Input: Entity e; Set of images P; Voting score of image $s(.)$; Similarity function between images $sim(.)$

Output: Set of image groups $\mathcal{G}$ with selected representative images and updated voting scores $s(.)$ of the representatives

```
1:  function GROUP(e, P, s(.), sim(.))
2:      G ← ∅                                   ▷ The final set of image groups
3:      for p ∈ P do
4:          isDistinct = true
5:          for G ∈ G do                        ▷ G is set of images
6:              if sim(p, r_G) = 1 then
7:                  G ← G ∪ {p}
8:                  s(r_G, e) += s(p, e)
                                    ▷ Update voting score of representative image
9:                  isDistinct = false
10:                 break
11:             end if
12:         end for
13:         if isDistinct = true then
14:             G' ← {p}
15:             r_G' ← p
16:             G ← G ∪ G'
17:         end if
18:     end for
19:     return G                                ▷ Final set of image groups
20: end function
```

the rank-based weighted voting scores (see Section 4.3) of all images in the respective classes. Then, we include only the representative images in the final ranking. Note that the representative images obtain exactly the enhanced voting scores, defined above.

Our grouping algorithm is presented in Algorithm 4.1. It starts with an empty set of groups $\mathcal{G}$. Then it processes all candidate images, for which we have already computed their rank-based weighted voting scores as described in Section 4.3. We compare a current image p with all current groups in $\mathcal{G}$ for visual similarity. For a group $G \in \mathcal{G}$ we test if p is visually similar to its representative image r_G (see Line 6), using the binary function $sim(.)$ defined in Section 2.2. If p is visually similar to r_G, we add p to the group G and update the voting score of r_G by adding to it the voting score of p (see Line 8). If p is not visually similar to any of the representative images in the current groups, we create a new group of images, containing only p and assign as a representative image for this group p.

The worst case complexity of our algorithm is quadratic in the number of images. The algorithm has certain limitations. First, the outcome depends on the order of processed images. Second, the $sim(.)$ function, based on which we add images to groups, is not a distance function. Third, due to its high computational cost, the visual similarity check is performed only with respect to the representative images of the groups, which can result in assigning visually different images to the same group. However, in our experiments we show that the use of our grouping algorithm and the removal of near-duplicate images from the result list can greatly enhance the results.

4.5 Logistic Regression Model

Instead of the above model for ranking images, we could alternatively model our problem as a classification task for recognizing correct photos or as a regression problem for scoring the retrieved results. We consider the following binary random variables:
$Y = 1$, if a given photo is correct for a target entity e, and $Y = 0$ otherwise, and
$X_i = 1$, if a given photo is retrieved by the query $q_i(e), i \in \{1, \ldots, m\}$, and $X_i = 0$ otherwise.
We devise a classification model which reasons about the probability $P[Y|X_1 \ldots X_m]$, by using a logistic-regression model of the following form [Mitchell, 1997]:

$$P[Y|X_1 \ldots X_m] = \frac{exp(\sum_{i=1..m} w_i X_i)}{1 + exp(\sum_{i=1..m} w_i X_i)}$$

where w_i are feature weights that are learned by maximizing the (regularized) log-likelihood of the training data using Quasi-Newton optimization methods. A new test photo is accepted by a logistic-regression classifier if its in-class probability exceeds the out-of-class probability.

An analogous model can be devised by using instead of our binary variables X_i, integer-valued random variables: $R_i = j$, if a given photo is returned at rank j by query q_i, and $R_i = 0$ if the photo is not retrieved by q_i. In our experiments, the use of the rank-based variables did not improve the quality of the results. This is why, we show experimental results only with the binary variables X_i.

4.6 Experiments

We present an experimental evaluation of the proposed ranking methods for images of entities. The goals of our studies are as follows:

- To study the effectiveness of our rank-based weighted voting method in two settings: with and without visual grouping of images.
- To compare our rank-based weighted voting with rankings returned from standard image search engines and with a logistic regression approach.

4.6.1 Experimental Setup

Data. We used four classes of entities: scientist, politician, religious building, and mountain. Each class contains 15 training entities and 10 test entities (disjoint from the training set). Each of the training entities has between 10 and 100 hand-selected photos, depending on whether the entity is highly notable or not so notable. To generate the queries for each entity we use relational facts specific for each class. Table 4.1 lists a few test entities and a subset of their relational facts.

Methodology. For each test entity we posed the generated queries to Google and Bing for the people classes and to Google and Flickr for the landmark classes. We collected the top-100 from each result list, and applied our scoring models. We showed the entire pool of results to human judges for binary relevance assessment. The judges considered a photo as relevant if they could clearly recognize the target entity, possibly after reading the Web page where the image was found. For the people classes, not only personal photos were accepted, but also when the person could be recognized in a group with others. For the landmark classes the judges accepted images that show the place including unusual perspectives, but disregarded those images that did not show anything specific for the place and could have been taken in many other places (e.g., a close-up of a snow patch on a mountain). In case we consider visual similarity between pictures and we group similar images into classes only the representative images are shown to the judges.

Competitors. For each entity type and search engine we compare three methods:

- **Original:** the original search engine rankings;
- **Voting:** our rank-based weighted voting methods from Section 4.3 and Section 4.4;
- **Regression:** the logistic regression model with binary features from Section 4.5. We used the ridge logistic regression implementation provided by the WEKA toolkit [Hall et al., 2009].

We present results for two different kinds of rankings:

- **Normal rankings:** we consider pictures to be duplicates only by URI comparison;

Class	Entity	Relational Facts
scientist	Alfred Louis	field: Mathematics institution: Saarland University
	David Patterson	known for: RISC, RAID institution: University of California, Berkeley awards: ACM IEEE Eckert-Mauchly Award
	Niklaus Wirth	Alma Mater: ETH Zürich awards: Turing Award known for: Pascal, Algol W, Modula, Oberon
politician	Jon Huntsman	political party: Republican position: Governor of Utah
	Ignatz Bubis	birthplace: Breslau death year: 1999 profession: Jewish leader
	Niels Annen	political party: SPD position: Jusos
building	Wat Arun	location: Bangkok known for: Buddhist temple names: Temple of the Dawn
	Einsiedeln Abbey	known for: Benedictine monastery location: Switzerland
	Boyana Church	location: Sofia, Bulgaria known for: Boyana Master
mountain	Siula Grande	location: Peru height: 6344 range: Cordillera Huayhuash
	Mount Ararat	names: Mountain of Pain location: Dogubayazit location: Agri Province, Turkey
	Dreieckhorn	range: Bernese Alps height: 3811 location: Switzerland

Table 4.1: Examples for entities and relational facts.

- **Diversity-aware rankings:** we group visually similar images into equivalence classes and we include only the representative images of each class in the final ranking. For the rank-based weighted voting model we use the method from Section 4.4. The results from the logistic regression model are diversified in an analogical way. We apply visual grouping to the original search engine rankings as follows: starting from the top ranks, whenever we meet a result that is visually similar to a result higher in the ranking, we remove the lower-ranked one.

Quality Measures. To compare the results of the different methods, we use three quality measures: Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and a preference-based measure (*bpref*).

The MAP measure is the mean of the precision scores obtained at the ranks of each relevant image, which is an interpolated approximation of the area under the precision-recall curve. It is computed as follows:

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

where $E = \{e_1, e_2, \ldots\}$ is the set of test entities, n_i is the number of relevant images for entity e_i, d_j^i is the j^{th} ranked result for e_i returned by a retrieval algorithm R, and $rel(d_j^i)$ is the binary relevance assessment for this result.

Additionally, we compute NDCG to measure the usefulness (gain) of images based on their (geometrically weighted) positions in the result list. It is computed as follows:

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

where N_{ki} is a normalization factor calculated to make NDCG at k for entity e_i equal to 1 in case of perfect ranking.

Recall that our use of search engine queries can practically retrieve only a small subset of the full result sets, the top-100 in our setup. By inspecting only the top-k results for each query, it is impossible to know whether a relevant image has not been found at all or simply because the rank of the image is higher than k. And some sophisticated queries may return less than k results. This situation is rectified as follows (using TREC-style practice). Consider m methods (runs) under comparison. Each method returns a ranked list, truncated at rank k. Suppose we have a total of N distinct results from all the result lists ($N \leq k \times m$)). From the N results, the human assessors give us a set of R relevant images. The next step is to pad each result list with the missing relevant images. For each method m_j that has R_j ($R_j < R$) relevant results and k results overall, we add the remaining $R - R_j$ relevant results on (virtual) ranks $k+1$, $k+2$, etc. If method m_j has only $k' < k$ results overall, then we consider ranks $k'+1$, $k'+2$,...,k as non-relevant and add the remaining $R - R_j$ relevant results at ranks $k+1$, $k+2$, etc. This way all methods are evaluated as if they had 100% recall, based on the pooled results of all methods, and we can compute the standard MAP measure.

		Google			Bing/Flickr		
		Original	Voting	Regression	Original	Voting	Regression
S	MAP	0.620	0.718	0.600	0.614	0.684	0.545
	NDCG	0.818	0.914	0.821	0.829	0.891	0.818
	bpref	0.600	0.755	0.704	0.664	0.768	0.657
P	MAP	0.789	0.801	0.719	0.730	0.759	0.674
	NDCG	0.938	0.945	0.906	0.914	0.929	0.879
	bpref	0.737	0.780	0.706	0.747	0.814	0.749
B	MAP	0.728	0.772	0.696	0.780	0.839	0.800
	NDCG	0.867	0.907	0.852	0.875	0.907	0.891
	bpref	0.627	0.746	0.650	0.660	0.741	0.662
M	MAP	0.805	0.830	0.820	0.857	0.847	0.805
	NDCG	0.931	0.956	0.948	0.964	0.961	0.949
	bpref	0.670	0.710	0.689	0.727	0.712	0.686

Table 4.2: Evaluation measures for normal result rankings for entity classes: scientist (S), politician (P), building (B), mountain (M).

Note that because 1) the true recall can be much larger than our pooled result set, and 2) each method in our setup typically returns a very small subset of the full recall (top-100 out of potentially many thousands of photos), the padded result lists tend to have similar MAP values when $R >> k$. For this reason, we also computed the *bpref* measure which is highly correlated to MAP when complete information is provided and more robust otherwise. For a bounded ranked list with top-k results and a total of R relevant results, *bpref*(k) is defined as follows:

$$bpref(k) = \frac{1}{R} \sum_{r} 1 - \frac{|\#n \text{ ranked higher than } r|}{k + R}$$

where the summation ranges over the ranks r of relevant retrieved results and $\#n$ counts non-relevant results. *bpref* does not depend on potential results (from the pool of all methods' results) on ranks $> k$. Thus, it does not degrade as much as MAP when $R >> k$.

4.6.2 Results

Normal Rankings. The results for normal rankings are shown in Table 4.2. For all baselines Google, Bing, and Flickr, our voting method almost always improves all three measures MAP, NDCG, and *bpref*. (The one exception is the Flickr ranking for mountains; see discussion below.) We observe that the gains vary depending on the entity type. For example, for the scientist class, when using Google, the MAP value increases from 0.62 to 0.718. In contrast, for the politician class the absolute improvement is less than 2%. We note that *bpref* shows higher gains for reasons discussed above. Similar observations hold for Bing and Flickr. The results also show that the logistic regression model does not perform well in the grand total. Our

		Google			Bing/Flickr		
		Original	Voting	Regression	Original	Voting	Regression
S	MAP	0.561	0.632	0.543	0.519	0.602	0.544
	NDCG	0.794	0.878	0.799	0.775	0.862	0.829
	bpref	0.633	0.809	0.806	0.712	0.797	0.763
P	MAP	0.727	0.768	0.662	0.656	0.721	0.626
	NDCG	0.915	0.936	0.870	0.886	0.917	0.860
	bpref	0.748	0.846	0.793	0.708	0.826	0.775
B	MAP	0.665	0.726	0.672	0.729	0.822	0.778
	NDCG	0.845	0.878	0.839	0.860	0.904	0.885
	bpref	0.573	0.809	0.745	0.631	0.789	0.732
M	MAP	0.764	0.822	0.829	0.824	0.828	0.805
	NDCG	0.921	0.954	0.957	0.954	0.957	0.949
	bpref	0.605	0.757	0.769	0.669	0.740	0.710

Table 4.3: Evaluation measures for diversity-aware result rankings for entity classes: scientist (S), politician (P), building (B), mountain (M).

unusual notion of "features" derived from noisy query results seems to be difficult to handle by standard machine learning. However, for a few individual entities, the regression model actually performed best.

Table 4.4 and Table 4.5 show the weights for (a subset of) different types of relational facts that our voting method uses, based on its parameter estimation from the training entities. Note that the weights are not normalized (and do not need to be). Not surprisingly, the original name tends to have the highest weight, and there are big differences in the usefulness of the other relations. The most useful relations were: the field of research for scientists, the political party for politicians, and the location for the two landmark classes.

Diversity-Aware Rankings with Visual Similarity. We have also applied the extended scoring model with visual similarity to the three methods Original, Voting, and Regression. In this case, near-duplicates are clustered and only one representative of each cluster is included in the final result list. Table 4.3 shows the different measures for these diversity-aware rankings. Similarly to the results with normal rankings, our voting method consistently improves all three measures (now without any exceptions). On average, the gains over the baseline competitors were even higher here than in the normal ranking comparison. The *bpref* measure shows the largest improvements. For example, for scientists using Google, our method improved *bpref* from 63% to 80% and achieved a similar gain for politicians. For buildings, we gained even more: from 57% to 80%; and even for the difficult mountain class, the *bpref* improvement is substantial (from 60% to 75%).

But there are again major differences in the magnitude of the improvement, depending on the entity type. Note that the absolute values of MAP, NDCG, and *bpref* are slightly lower than for normal rankings, because duplicates and near-duplicates

	entity name	birth year	field	institutions	polit. party
scientist	0.594/1.3	0.328/0.829	0.411/1.066	0.314/ 0.814	n/a
politician	0.579/1.254	0.314/0.731	n/a	n/a	0.461/0.878

Table 4.4: Normal weights / similarity weights for the scientist and politician classes using Google.

	entity name	location	height	range	known for
building	0.598/1.448	0.514/1.222	n/a	n/a	0.351/0.863
mountain	0.573/1.079	0.354/0.703	0.256/0.619	0.294/0.616	0.257/0.622

Table 4.5: Normal weights / similarity weights for the religious building and mountain classes using Google.

of good results are now discounted. Also, the relative weights of different types of relational facts are adjusted (see Table 4.4 and Table 4.5) because visual similarity is considered for the photos of the training entities as well. For example, the *knownFor* relation is additionally boosted with visual similarity, relative to other relations such as *location*. In fact, our experiments show that this leads to better results.

4.6.3 Discussion

Our experimental results show that our voting method is almost always more effective than the native rankings of image search engines, by a significant margin. Sometimes, however, the gains are small and generally depend on the entity type or even on the individual instance. In the following, we discuss some of the specific strengths of our method by means of anecdotic examples. We also point out limitations of our approach.

Specific Strengths. We are performing particularly well for entities with ambiguous names or when an entity is very rare in the Internet photo space. Examples are shown in Table 4.6 for normal ranking and Table 4.7 for diversity-aware ranking. Figure 4.4 shows top-ranked result photos, with visual-similarity grouping, for our method vs. those ranked high by image search engines, for a couple of example entities. Each block shows the top-5 groups (from top to bottom). Only up to 3 photos per group are shown; some groups contained many photos, others were small.

In the scientist class, the search engines confused David Patterson with the New York governor Paterson. This is shown in the upper right part of Figure 4.4. Our voting method's result is not perfect either, but at least has 4 correct (groups of) photos in the top-5. William Vickrey, in the upper left part of Figure 4.4, turned out to be a difficult case because many of his photos are on content-rich Web pages with lists of Nobel Prize winners in Economy and many photos. Here, our top-5 results are perfect, whereas the search engine got only 3 out of 5 results right. Other difficult cases can be found in Table 4.6 and Table 4.7. They include Emmy Noether,

as search engines also returned winners of an Emmy Noether Fellowship (by the German Science Foundation, named after her), Alfred Louis, as his last name is also a common first name. In the politicians class, we performed particularly well on lesser known people such as Ignatz Bubis or Renate Blank. Their names do occur often in news about parliamentary debates and other events of this kind, but these news contain photos of other people related to the same event.

We observed similar effects for the two landmark classes. For example, the mountain Pilatus, shown in the lower left part of Figure 4.4, turned out to be ambiguous because there is also an aircraft model called Pilatus. For landmarks, some individual entities were challenging due to the fact that they are often mentioned on tourist sites that have many photos but not for every attraction that they talk about. For example, in the Google results for the Church of Christ Pantocrator (in Nessebar, Bulgaria), shown in the lower right part of Figure 4.4, 3 out of the top-5 results are wrong: at ranks 1, 3, 4. They show an icon and a relief from other churches and a similar but different church, all of which are mentioned together on popular tourist sites about Balkan culture. In contrast, our voting model improved the results and achieved 60% precision in the top-5 results. In general, for entities of this difficult nature, we achieved major gains over the baseline competitors.

Limitations. Although we aimed at entities in the "long tail" of notable but not famous people and places, the need for manually assessing the correctness/relevance of results entailed that our test entities were actually a mix of still fairly popular entities and some lesser known ones. For the popular entities, it was virtually impossible to beat the top-100 results of the two image search engines (unless the entity name was highly ambiguous). When search engines can choose from result sets with hundred thousands of photos, their ranking criteria obviously work extremely well. Thus, for famous people such as Frank Wilczek or Nelson Mandela we could not gain anything over Google and Bing, and occasionally even lost slightly in precision.

Likewise, for popular places, Flickr seems like a gold standard, given its rich tagging assets, and Google also performed extremely well. For example, the results for Wat Arun or Mount St. Helens could simply not be beaten. We realized, however, that Flickr tags are sometimes noisy; for example, an entire photo series on a Himalaya trip was uniformly tagged with "Himalaya", "India", "Tibet", "Everest", "Kailash", etc., although it is geographically impossible to have both Mount Everest and Mount Kailash displayed in the same photo. Unfortunately, these wrong tags also misled our method. In this regard, it would be interesting to use voting across results of different search engines. The combination of results from Flickr *and* Google, for different query expansions, may have the potential for overcoming this issue with noisy tags.

4.7 Summary and Outlook

Retrieval and ranking of photos has received great attention in the prior literature. In our work, we viewed this problem from the new angle of populating a knowledge base about people and landmarks with a large set of diverse pictures. In contrast to

previous work, where the focus has been on semantic classes or prominent entities, we paid particular attention to individual entities in the long tail of popularity.

In this chapter, we showed how to populate a knowledge base with images of named entities by utilizing relational facts about entities from the knowledge base. We developed methods for retrieving and ranking images for different entity types. Furthermore, we presented an approach which computes image-content similarities, which we used to diversify the final list of results. With our experimental studies we showed the effectiveness of our approaches and the improvements over standard image search result lists.

Outlook. Our approach for finding images of entities achieved very good experimental results but had significant limitations: (1) dependence on ontological facts which are not always available, either because the knowledge base is not well populated for the entity of interest, or because the entity is not listed at all in the knowledge base, (2) the need for training samples for each entity type which can be a bottleneck, because it requires significant human supervision, and (3) the high overhead caused by query expansions resulting in a large number of search-engine requests. These limitations are overcome with our methods presented in the next chapter.

Figure 4.4: Example results with visual similarity grouping. Each block shows the top-5 visual groups (from top to bottom). Only up to 3 pictures per group are shown (from left to right); the leftmost picture is representative for the group.

		Google			Bing/Flickr		
		Original	Voting	Regression	Original	Voting	Regresssion
scientist	Alfred Louis	0.020	0.810	0.078	0.085	0.767	0.455
	David Patterson	0.130	0.569	0.433	0.157	0.685	0.771
	Emmy Noether	0.838	0.892	0.847	0.908	0.956	0.957
politician	Ignatz Bubis	0.539	0.736	0.638	0.695	0.749	0.686
	Jon Huntsman	0.851	0.952	0.953	0.819	0.845	0.852
	Renate Blank	0.507	0.590	0.474	0.524	0.665	0.620
building	Church of Christ Pantocrator	0.301	0.678	0.413	0.362	0.779	0.566
	San Lorenzo	0.029	0.142	0.069	0.020	0.030	0.020
mountain	Pilatus	0.406	0.533	0.529	0.879	0.886	0.910
	Mönch	0.309	0.669	0.756	0.986	0.996	0.872

Table 4.6: Examples for MAP values of normal rankings for individual entities.

		Google			Bing/Flickr		
		Original	Voting	Regression	Original	Voting	Regresssion
scientist	David Patterson	0.105	0.416	0.263	0.144	0.551	0.572
	Emmy Noether	0.593	0.698	0.647	0.748	0.842	0.889
	William Vickrey	0.499	0.691	0.585	0.558	0.673	0.578
politician	Ignatz Bubis	0.550	0.696	0.563	0.590	0.697	0.601
	Stephen Crabb	0.562	0.618	0.470	0.549	0.679	0.442
	Luisa Diogo	0.766	0.823	0.786	0.720	0.767	0.681
building	Church of Christ Pantocrator	0.234	0.442	0.265	0.335	0.769	0.592
	Boyana Church	0.755	0.782	0.708	0.738	0.844	0.800
mountain	Tre Cime di Lavaredo	0.954	0.974	0.988	0.833	0.869	0.817
	Aiguille d'Argentiere	0.787	0.788	0.674	0.876	0.895	0.877

Table 4.7: Examples for MAP values of diversity-aware rankings for individual entities.

Chapter 5

Knowledge Kaleidoscope with Keyphrases

5.1 Introduction

Chapter 4 presented an approach for populating a knowledge base with images of named entities. It utilized relational facts for the entities already known in the knowledge base. In the following we present a very different, more light-weight, and more robust approach for solving the same problem.

Motivation. Knowledge bases such as DBpedia or Freebase organize millions of entities and 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 images of entities. For example, out of the 9921 articles in the Wikipedia category *Competitors at the 2012 Summer Olympics*, many articles do not have an image of the sports competitor. The same problems hold for scientists, artists, and landmarks 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, different ages, or from different perspectives.

Challenges. It is often difficult to find good images of long-tail entities 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 entity). Furthermore, 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 dominated by the actor Kevin Spacey who acted in the movie "The Life of David Gale", which is totally unrelated to the economist (see Figure 5.1). 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-20 search results for Robert Floyd, who has received the Turing award, contain only two correct results, at ranks 3 and 7, as shown in Figure 5.1.

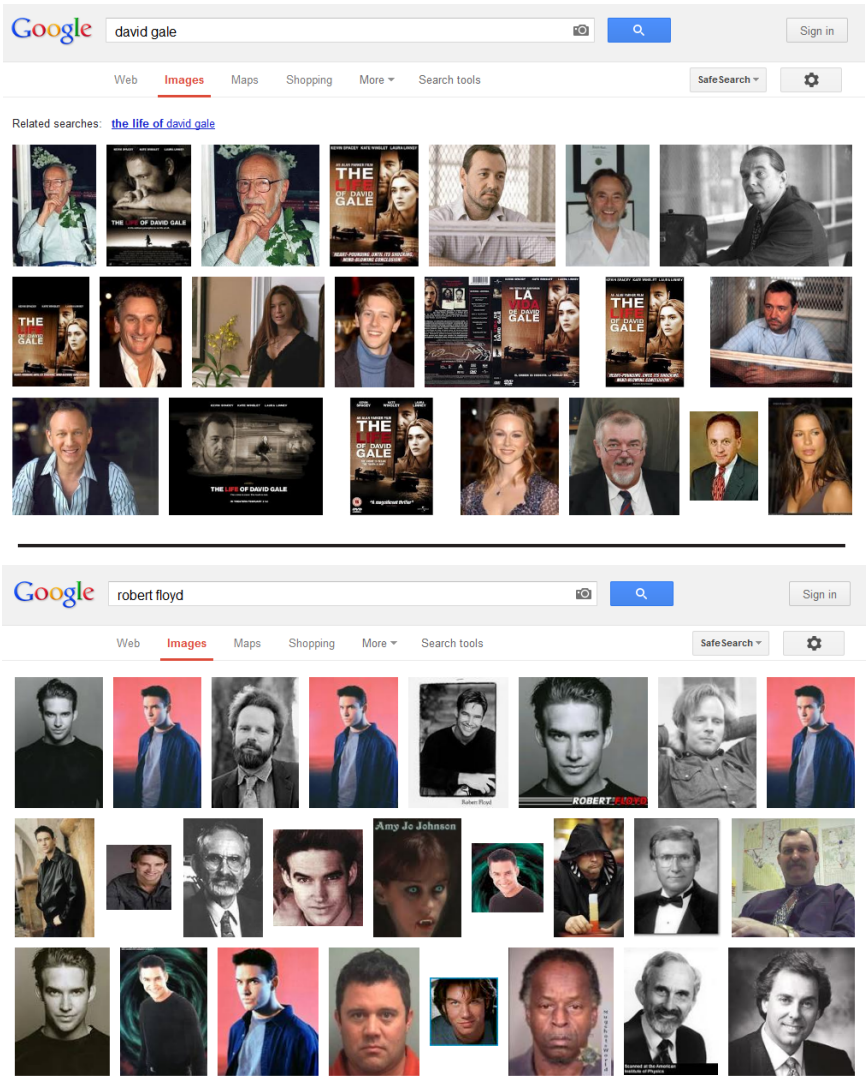

Figure 5.1: Top results returned by Google image search for "David Gale" and "Robert Floyd" (as of February 2013).

Objective. Similarly as in Chapter 4 our goal is to automatically populate a knowledge base with images of named entities like people or places. We focus on entities which have ambiguous names or are rare on the image Web space. Our goal is not just finding one image of the entity, but to find many (ideally different) images of the entity, where good images are ranked 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).

Approach and Contributions. Our approach for finding images of rare or ambiguous named entities operates as follows. 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 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. We 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 often the keyphrases are very long, and long queries tend to get highly diluted results.

We use only the entity 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 earlier. For each image in the pool we identify *full or partial matches* of the keyphrases in the Web page containing the image, and compute a new form of relevance score used for re-ranking. In addition, we optionally group visually similar images to obtain a diversified final list of results. Our framework supports various kinds of score-aggregation models; we experimentally found that a novel form of cover-based model works best. 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.

In summary, this chapter 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 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.

Outline. The rest of the chapter is organized as follows. Section 5.2 presents the system architecture of our approach. Section 5.3 introduces our approach for keyphrase extraction and mining. In Section 5.4 we describe our scoring model. Section 5.5 presents our test for entity difficulty, and Section 5.6 briefly describes our visual grouping of images. We describe our experimental results in Section 5.7. Finally, Section 5.8 summarizes the chapter.

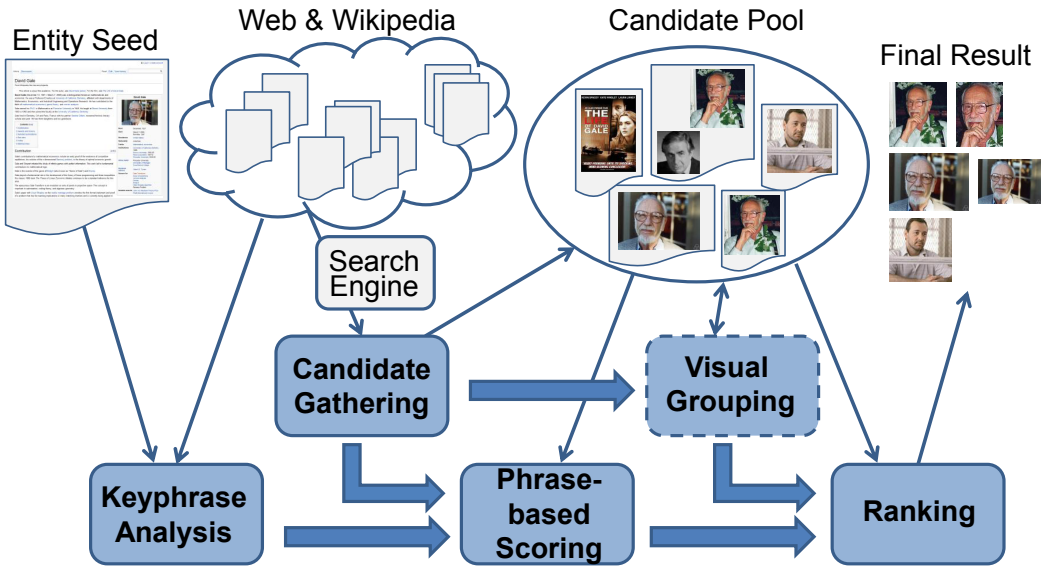

Figure 5.2: System architecture. Rectangles are system components. Thick arrows denote control flow and thin arrows show data exchange. *Visual grouping* is optional.

5.2 System Architecture

The overall system architecture is illustrated in Figure 5.2. The system consists of five major components:

- The *Keyphrase Analysis* component obtains a seed page for a given entity from the Web or from Wikipedia. In Section 5.3 we describe our approach for extracting entity-characteristic keyphrases from the seed page. We use the Wikipedia corpus to compute a ranking score for each of the extracted keyphrases and also for each of the individual words in the keyphrases. This score measures how well the keyphrase characterizes the given entity.

- 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 the entity keyphrases in the image pages, we assign phrase-aware scores to the images using the models from Section 5.4.

- The *Visual Grouping* component groups images into equivalence classes of near-duplicates as described in Section 5.6.

- The *Ranking* component ranks the candidate results for each entity based on their phrase-aware scores. Optionally this component may obtain a grouping

of the image results based on our visual similarity test. In this case, we assign a score to each group of images by summing all phrase-aware scores of the images in this group. Then, we rank the image groups based on their accumulated phrase-aware scores.

Since the keyphrase analysis and the 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.

5.3 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 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*) measure. In the following subsections, we describe the extraction of focused keyphrases and their weighting in more details.

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

5.3.1 Keyphrase Extraction

Noun Phrases. We use the OpenNLP tool[2] 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 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 texts of these links as focused keyphrases. We use the WikiPrep tool [Gabrilovich and Markovitch, 2007] for this purpose. We also considered anchor texts 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.

5.3.2 Keyphrase Weighting

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:

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

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 5.4.

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 5.4 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.

[2] `http://opennlp.apache.org/`

5.4 Phrase-Aware Scoring of Image Results

Assume that for an entity of interest we are given a candidate pool of images obtained from 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 this image. For the same entity we are also given a set of characteristic weighted phrases as described in Section 5.3. 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*. A single 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. We denote the set of entity characteristic phrases by $\{k_1(e), \ldots, k_m(e)\}$, or $\{k_1, \ldots, k_m\}$ when the entity is uniquely given from the context. For each image/page p we compute its phrase-aware score $s(p)$ as follows:

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

where $w(k_i)$ is the weight of the keyphrase k_i based on the MI measure computed as described in Section 5.3. By $\mathcal{S}(k_i, p)$ we denote the 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.

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.

Special Case of Words-aware Model. 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. The score $\mathcal{S}(k_i, p)$ is either 0 or 1, as a single word is either in the page or not.

5.4.1 Scoring based on Minimum Cover

The Minimum Cover [Tao and Zhai, 2007; Cummins and O'Riordan, 2009] 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 give implementation details about the computation of *mincover* below). 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 presented in [Tao and Zhai, 2007; Cummins and O'Riordan, 2009] 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 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 5.3). In this way, if some characteristic words from a keyphrase are missing in a page, the final keyphrase score is low. 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). 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 the page q is significantly better than the page p.

Algorithm 5.1 Compute Minimum Cover

Input: Inverted index lists $L[i], i \in \{1, \ldots, n\}$
Output: Minimum cover value

```
1:  function MinCover(L)
2:      mincover ← ∞                                   ▷ Minimum cover value
3:      H ← {0, ..., 0}                ▷ H[i] points to the current element in L[i]
4:      loop
5:          J ← { j | H[j] ≠ nil }      ▷ J is set of lists with unprocessed elements
6:          if J = ∅ then break
7:          v_min ← min_{j∈J} L[j, H[j]]
8:          v_max ← max_{j∈J} L[j, H[j]]
9:          cover ← v_max − v_min + 1                  ▷ Current cover value
10:         if cover < mincover then mincover ← cover
11:         idxMin ← arg min_{j∈J} L[j, H[j]]
12:         if H[idxMin] < size(L[idxMin]) − 1 then
13:             H[idxMin] ← H[idxMin] + 1
14:         else
15:             H[idxMin] ← nil
16:         end if
17:     end loop
18:     return mincover
19: end function
```

Computation of mincover. As explained above, $mincover(k \cap p, p)$ returns the length of the shortest text segment in page p where all words in $k \cap p$ appear at least once. Our pseudo-code for computing *mincover* is shown in Algorithm 5.1. Assume that $|k \cap p| = n$ and that we have inverted index lists for all phrase words in the page p: $L[i], i \in \{1, \ldots, n\}$. Index list $L[i]$ contains the positions of the i-th phrase word in the text in increasing order. Our algorithm scans the index lists from their first to last elements. We use the list H to point to the current elements in the lists: $H[i]$ points to the current element in $L[i]$. We start with the first elements from all index lists and we compute a current cover value (see Line 9). Then we choose the index list $L[idxMin]$ with the smallest current element (see Line 11), and we increase $H[idxMin]$ to point to the next element from this list. At each change of the current elements in the lists, we compute new cover value and compare it to the current minimum cover. Finally, we return the smallest *mincover* value.

5.4.2 Alternative Scoring Models

In this section we discuss two alternative models to the minimum-cover-based model. These models consider not only the best match of a keyphrase in a page, but all occurrences of the keyphrase words in the page. Our experiments with all phrase-based models showed that the minimum-cover approach is most effective and that the alternative models are comparable among each other.

Büttcher's Scoring Model. Büttcher's model [Büttcher et al., 2006; Schenkel et al., 2007] linearly combines a 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 5.3.

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 5.3). The keyphrase score of image page p and keyphrase k is then given by a linear combination of a variant of the BM25 score and an adapted proximity score:

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

where $k \cap p$ denotes the set of words from k that are contained in p. BM25* is a variant of the BM25 score:

$$\text{BM25}^*(k,p) = \sum_{t\in k\cap p} w(t) \frac{tf_{tp}(d_1+1)}{tf_{tp}+D}$$

where $w(t)$ is the MI weight of a phrase word t instead of the *idf* measure, and tf_{tp} is the frequency of t in p. The parameters D and d_1 are set to 1.2, following [Schenkel et al., 2007] and specializing to our setting, and λ is set to 0.2.

Scoring based on Spans. The spans-based approach of [Song et al., 2008; Svore et al., 2010] 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.

For example, suppose we want to score a page for the keyphrase "Escalante Grand Staircase National Monument" (e.g., to obtain photos for the landmark Coyote

Gulch). Consider the page text "Visit the Escalante canyons in the Grand Staircase. The Escalante river forms grand arches and stone monuments. This is very different from the Grand Canyon National Park.". Assume that $d_{max} = 8$ and that we disregard capitalization and singular/plural differences (so that "grand" and "monuments" are matches). The first span would begin at the first occurrence of "Escalante" and end at the second one. However, double occurrences are not allowed. So the first span is terminated at "Staircase", accumulating 3 matching words within span length 6. The second occurrence of "Escalante" would then start a new span and extend until "monuments", accumulating 3 matching words within span length 8. An alternative split could be to terminate the first span at the first "Escalante", thus creating a length-1 span, and combining everything "Grand Staircase. The Escalanate" into the second phrase, terminated at the second "Escalanate" because the next phrase-word match is a repeating "grand". This actually produces a wider gap between the first and the second span, and is therefore preferred. Finally, we obtain a third span "grand arches and stone monuments", with 2 word matches and length 5, and a fourth span "Grand Canyon National" with 2 word matches and length 3.

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

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

where $w(t)$ are *per-word weights* based on MI (see Section 5.3) and rc_{tp} is a "relevance contribution" of a word t in page p, which replaces the standard word frequency tf_{tp} by down-weighting occurrences in long spans. rc_{tp} is 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.

5.5 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 5.4.

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 group the set of Web pages that contain the image results using a simple grouping method, which produces a variable number of groups depending on a threshold for similarity. If an entity's results produce many groups (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 grouping method processes 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 group of that previous page; otherwise we create a new group. As a distance/similarity measure, we use the cosine similarity based on the *tf-idf* values of the words in the pages, 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 full Wikipedia corpus.

5.6 Grouping of Visually Similar Images

In addition to exploring the text of the pages containing the candidate images for entity-specific keyphrases, we can optionally consider the visual content of the images. We group images into groups of visual near-duplicates by using an algorithm analogous to Algorithm 4.1. The output of this algorithm is a set of image groups, where each group is assigned with a representative image. According to the models in Section 5.4, we compute a phrase-aware score for each candidate image. To order the image groups we assign new scores to each representative image by summing over all phrase-aware scores of the images in the respective group. The image groups are then ranked by the overall scores of their representatives.

In the final ranking we include only the representative images from each group and thus we obtain a visually diverse list of results. Furthermore, since the score of each representative image is computed by accumulating the phrase-aware scores of the images in the same group, we obtain better statistical evidence for the relevance of the images in the groups. The reason is that all images in a group have different underlying Web pages, and distinct pages have a different set of entity keyphrases.

5.7 Experiments

We present an experimental evaluation of the proposed methods for finding images of difficult entities. The goals in these experiments are the following:

- To study the effectiveness of our phrase-aware approach for finding images in various settings: with and without visual grouping of images, using different sources for entity seed pages (e.g., Wikipedia pages and arbitrary Web pages), with different sets of keyphrases (e.g., focused and noun phrases), and using different models to compute keyphrase scores (e.g., scoring based on Minimum Cover, Büttcher's model, and spans-based model).

- To compare our phrase-aware approach with other approaches for finding and ranking images of named entities, including the rankings of standard image search engines and a language-model-based approach.

5.7.1 Experimental Setup

Methodology. We evaluated our phrase-based 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", for which the image search engines perform already very good. To decide whether an entity is difficult or not, we used our robustness test for entity difficulty presented in Section 5.5. 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 5.3. Table 5.1 shows three 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 standard 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 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:

- "Scientists" with 56 entities taken from the "Turing Award laureates" category. From these 34 were difficult, as concluded by the test for entity difficulty from Section 5.5;
- "Waterfalls" with 20 entities taken from the category "Waterfalls of British Columbia", out of which 14 difficult ones.

The broad themes contain the entities of the lists:

- "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 two broader and much larger categories we randomly sampled 25 entities from each, excluding extremely prominent entities with perfect precision on the first

	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 5.1: Examples for highest-MI focused keyphrases extracted from Wikipedia.

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 the "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.

Competitors. We compare the following methods:

- **Phrases:** our phrase-aware model based on the **minimum-cover** matching of (focused) keyphrases from Section 5.4.1;
- **Büttcher:** our phrase-aware approach using Büttcher's model to compute keyphrase scores (see Section 5.4.2);
- **Spans:** our phrase-aware approach using spans to compute keyphrase scores (see Section 5.4.2);
- **Words:** our words-aware model as a special case of the phrase-based method (see Section 5.4);
- **Google:** the original search engine, as a main baseline;

- **Google-Exp:** the original search engine with query expansion, by including the highest-MI keyphrase in the entity query;
- **KL:** 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 [Zhai and Lafferty, 2006; Zhai, 2008].

Another possible opponent to our phrase-based approach would be our method from Chapter 4 based on query expansions. However, such comparison is not directly feasible for the following reasons. The method from Chapter 4 depends on (1) an ontological type system for entities, (2) training-based weights for each type, and (3) a knowledge base with salient RDF facts about each entity. Therefore, we do not include such a comparison here.

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 [Radlinski and Craswell, 2010] by considering only the top-k results:

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

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:

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

where N_{ki} is a normalization factor calculated to make NDCG at k equal to 1 in case of perfect ranking.

The Precision at k (P@k) measure is defined by the fraction of the top-k results that are relevant. The MRR measure is the average of the reciprocal ranks of the results for a set of test entities. A reciprocal rank for an entity e_i is the multiplicative inverse of the rank of its first relevant result r_i: $\text{MRR}(R) = (\sum_{i=1}^{|E|} 1/r_i)/|E|$.

		Phrases	Words	KL	Google	Google-Exp
Scientists	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
Waterfalls	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
Economists	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
Ruins	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 5.2: Evaluation for entity categories with Wikipedia seed pages.

5.7.2 Results

Ranking based on Wikipedia Seed Pages. The results for the ranking models using Wikipedia seed pages are shown in Table 5.2. The phrase-aware model based on minimum-cover matching of keyphrases 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 "Scientists" and "Waterfalls" categories, respectively. 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 are amazingly good. They are more effective than the search engine baseline on all entity categories with the exception of the waterfalls. The phrase-aware model is almost always better than 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.

Overall, the main insight from these experiments is that the phrase-based model with minimum cover achieves significant gains over all alternative models. In a few cases, other methods have comparable or slightly better results, but these differences are negligible.

		Focused Phrases	Noun Phrases	KL	Google
Scientists	MAP@50	0.599	0.595	0.591	0.587
	NDCG@50	0.931	0.929	0.926	0.885
Waterfalls	MAP@50	0.618	0.592	0.589	0.588
	NDCG@50	0.894	0.880	0.876	0.883
Economists	MAP@50	0.628	0.591	0.625	0.572
	NDCG@50	0.895	0.864	0.897	0.855
Ruins	MAP@50	0.594	0.592	0.552	0.499
	NDCG@50	0.934	0.932	0.909	0.823

Table 5.3: Evaluation for the phrase-aware model with focused phrases and with all noun phrases extracted from Wikipedia seed pages.

		Phrases	Büttcher	Spans	KL	Google
Scientists	MAP@50	0.599	0.590	0.592	0.591	0.587
	NDCG@50	0.931	0.924	0.926	0.926	0.885
Waterfalls	MAP@50	0.618	0.578	0.592	0.589	0.588
	NDCG@50	0.894	0.881	0.881	0.876	0.883
Economists	MAP@50	0.628	0.558	0.586	0.625	0.572
	NDCG@50	0.895	0.845	0.863	0.897	0.855
Ruins	MAP@50	0.594	0.572	0.576	0.552	0.499
	NDCG@50	0.934	0.917	0.921	0.909	0.823

Table 5.4: Evaluation for the phrase-aware model with the minimum-cover-based model, Büttcher's and Spans-based models.

Ranking with Noun Phrases. In Table 5.2 the results for the phrase-aware model are obtained using only focused keyphrases (see Section 5.3). Table 5.3 shows a comparison for the phrase-aware model (with minimum cover) between using focused phrases and using all noun phrases from the seed page. The results clearly show that focused keyphrases are essential for the effectiveness of the phrase-based model.

Ranking with Büttcher's and Spans-based Models. In Table 5.4 we present a comparison of our phrase-based methods from Section 5.4: the minimum-cover-based model, Büttcher's model and the spans-based model. The results show that our minimum-cover-based model is most effective with largest gains for the "Waterfalls" and "Economists" categories. Another observation is that the spans-based model is better than Büttcher's model for all categories. Furthermore, the spans-based model is almost always better than the original search engine's ranking (except for the "Waterfalls" category w.r.t. NDCG), while Büttcher's model loses on two categories.

Ranking with Visual Grouping of Images. Table 5.5 compares the re-ranking models with grouping of visually similar images (see Section 5.6). We apply the visual

		Phrases	Words	KL	Google	Google-Exp
Scientists	MAP@50	0.643	0.639	0.615	0.604	0.422
	NDCG@50	0.928	0.926	0.902	0.873	0.891
Waterfalls	MAP@50	0.647	0.643	0.610	0.625	0.208
	NDCG@50	0.889	0.888	0.857	0.878	0.675
Economists	MAP@50	0.632	0.649	0.636	0.612	0.197
	NDCG@50	0.874	0.884	0.887	0.859	0.668
Ruins	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 5.5: Evaluation with Wikipedia seed pages and visual grouping of images.

		Phrases	Words	KL	Google	Google-Exp
Scientists	MAP@50	0.476	0.484	0.405	0.308	0.375
	NDCG@50	0.906	0.911	0.853	0.686	0.863
Waterfalls	MAP@50	0.644	0.646	0.557	0.518	0.178
	NDCG@50	0.915	0.913	0.856	0.823	0.562
Economists	MAP@50	0.542	0.498	0.489	0.344	0.272
	NDCG@50	0.909	0.854	0.876	0.725	0.786
Ruins	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 5.6: Evaluation with non-Wikipedia seed pages.

grouping to the words-aware model in a similar way. 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 achieve very good results in this setting, but still lose against the phrase-aware model in most cases.

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 with worst results 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 the economists or the scientists 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 since they are not informative for the entity. The results are shown in Table 5.6. For these very difficult entities, we observe that the phrase-aware model is more effective than the search engine baseline and the KL-divergence-based model by a large margin. The words-aware model is comparable to the phrase-based model, as, in these cases, many keyphrases were merely one-word phrases.

5.7.3 Discussion

Comparing the three main competitors – phrase-based model with minimum cover, words-aware model, and KL-divergence-based model – to the search engine baseline, we observe the following major trends. All three methods achieve better results than those of the search engine. The phrase-based method is almost always better than the search engine, whereas the other two models sometimes achieve worse results compared to the baseline. The words-aware and KL-divergence-based models sometimes are slightly better than the phrase-based model, but the gains are negligible. Conversely, the gains of the phrase-based model over the KL-divergence-based one are significant; they are most pronounced for the entities with Wikipedia seed pages from the "Ruins" and "Waterfalls" categories (see Table 5.2) and the most difficult entities from all four categories for which we used noisy and sparse non-Wikipedia seed pages (see Table 5.6).

Specific Strengths. The phrase-based method achieves particularly good results for entities with 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 (see Figure 5.3). There exist (at least) two palaces with the same name, one in Potsdam and one in Haiti. Other examples of this nature include David Gale, Dawson Falls, Fred Brooks, etc. Note that the results shown in Figure 5.3 are obtained using a candidate pool of images retrieved in May 2011 (and hence the results for David Gale do not correspond to the top-k results shown in Figure 5.1).

In addition to entities with ambiguous names, our method performs very well also for entities, which are rare in the Internet image space. For example, the search engine results (as of May 2011) for the computer scientist Robert Floyd contained only 2 correct images in the top-50, on ranks 3 and 15, while the phrase-based method ranked these matches on the first two ranks.

Limitations. Since we use entity-specific keyphrases to compute ranking scores for the candidate images, the ranking of the images can only be as good as the textual information in their underlying Web pages. This is why 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. We tried to overcome this issue by reasoning on the visual content of the images (see Section 5.6). However, because of the small set of candidate images and the diversification efforts of the 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.

5.7.4 Potential Improvements

To overcome the limitations of our approach mentioned above we can improve our method in the following directions.

Larger Pool of Images Using Web Search. One option for collecting a larger pool of images is to consider standard Web search instead of image search. We would query the search engines with the entity name, but then we would collect Web pages, from which we select the candidate images. By using Web search we speculate that we would obtain a larger pool of images, because image search engines aim at diversifying the result list of images, which means that many visual near-duplicates will not appear in the result lists (or in their top-k prefixes). Furthermore, by using Web search instead of image search, we would depend less on the hidden functionality of image search engines.

Larger Pool of Images Using Query Expansions. Another option for collecting candidate images is to use the entity-characteristic keyphrases to pose many expanded queries to image search engines. As candidate images we would then use the results from all queries. We tried this approach but we observed a lot of noise in the image results, which led to unsatisfactory results. The main reason is that not all keyphrases are suitable for query expansions. Overly long keyphrases sometimes yield result lists without a single relevant image. Furthermore, the pages retrieved by using expanded queries contain the entity name and the keyphrase used for expansion, but they do not necessarily contain good result images. Yet, these wrong images appear in the results from Google image search and they also sometimes obtain relatively high phrase-aware scores. To overcome this issue we could further reason on the keyphrases and learn which ones yield potentially good result images.

Features of Image Search Engines. Features, which are probably essential for the good performance of today's image search engines include the proximity of the query keywords to the image in the respective Web page, the quality and size of the image, face recognition, and many others. The ranking order of the image results as returned by the search engine encodes all of these features and could be very helpful as well.

To enhance our methods for re-ranking of images we could consider some features used by the search engines. While some features are straightforward to implement, others require more efforts. For example, in order to consider proximity between an image and the query keywords in a Web page, it is not sufficient to use only the HTML text of the page. Often large distance between two points of interest in the HTML text appears very small on the screen, and vice versa. To consider proximity between the image and the query keywords we could use an HTML layout engine such as WebKit (`webkit.org`) and compute distances between their frames.

Multilingual Keyphrases. Our notion of keyphrases includes phrases which are extracted only from Wikipedia articles or standard Web pages in English. However, often there are relevant images of given entities, which are contained in Web pages

written in other languages. Since our methods for ranking utilize mainly the text in pages that contain the candidate images, images which have underlying Web pages in a language other than English, will receive very low ranking scores. A possible solution for this problem is to consider multiple languages for the seed pages of the entities, for example different Wikipedia versions. In this way we would obtain keyphrases in different languages, which in turn would help us reason on pages with image results in different languages.

5.8 Summary

In this chapter we showed how to populate a knowledge base with images of difficult named entities using entity-characteristic keyphrases. We have developed methods for finding matches or partial matches of the keyphrases in the Web pages that contain the candidate images. Our experimental studies demonstrated that this entity-oriented ranking of images leads to better results compared to the ranking returned by Google image search. 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 mostly at the level of query keywords and their proximity to images. In contrast our approach is specifically designed for target entities of interest and uses short initial descriptions of the entities.

Figure 5.3: Examples for phrase-aware rankings and Google result rankings without visual grouping: Sans-Souci Palace - ruin in Haiti; Dawson Falls - waterfall in British Columbia; James Tobin - economist; Fred Brooks - Turing Award winner; David Gale - economist.

Chapter 6

Entity-Knowledge Maintenance

6.1 Introduction

In this chapter we propose an approach that automatically extracts, from the Web, text contents for given input entities. Our goal is to accelerate the task of knowledge maintenance on entities in the long tail by retrieving salient contents which are related to them.

Motivation. Knowledge bases such as DBpedia, Freebase, or Yago have become essential assets for Web search, recommendations, analytics, and more. For example, the Google Knowledge Graph is centered around Freebase and used for many purposes within Google. Wikipedia, from which many knowledge bases are derived, has been used as a source for distant supervision in numerous tasks in IR and NLP. While knowledge bases are up-to-date on prominent entities, their maintenance on entities in the long tail and the acquisition of knowledge about newly emerging entities are bottlenecks. The root cause here is the human contributors who need to continuously identify and read relevant sources, in order to update articles or structured knowledge (infoboxes, categories, etc.) on long-tail or emerging entities.

New articles in Wikipedia are first created as stub pages which lack the desired encyclopedic coverage. Currently, the English Wikipedia has several 10,000's of articles containing the statement "*This article about ... is a stub. You can help Wikipedia by expanding it.*" As an example, consider the Wikipedia article about Liu Yang, the first Chinese astronaut in space. Her Wikipedia page was created shortly before the launch of her mission, as a one-liner without any categories. It took several days before the page was expanded with contents. Another example of different nature is the article about the famous database researcher Jennifer Widom (see Figure 6.1). This page exists for 3 years, but is still extremely terse. It does not know that she graduated from Cornell, worked for IBM Almaden Research, and made important contributions to semi-structured data models and stream query languages. However, all this information is easily available on the Web and could be used to expand the Wikipedia page. This problem, which applies to other knowledge bases as well, has recently led to the TREC challenge of *Knowledge Base Acceleration* (`http://trec-kba.org/`): can we help authors/editors of knowledge bases to maintain encyclopedic contents in a timely manner, by giving intelligent recommendations about salient events and

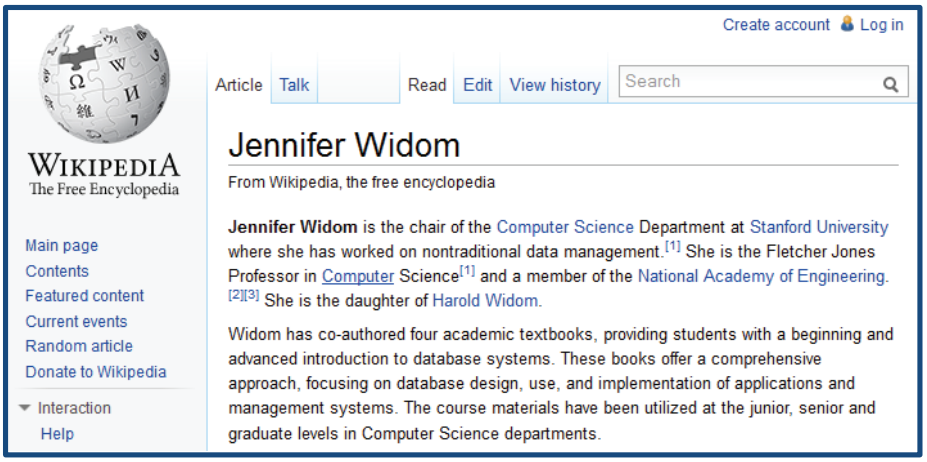

Figure 6.1: The Wikipedia article of Jennifer Widom as of February 2013.

facts that should be considered for updating the knowledge-base contents about an entity.

Problem Statement. Our goal in this chapter is to automatically compile such *salient contents* about entities, in order to ease knowledge bases maintenance. We refer to the output of this task as *"gems"*: text excerpts compiled from several relevant Web sources. In order to avoid overwhelming users, our goal is to compile highly informative, concise gems about *long-tail or emerging entities*. A good set of gems should not contain contents unrelated to the entity of interest and should not have redundancies. Gems should usually be short (e.g., a few hundred words in total), but allow user-configurable length constraints.

For compiling gems, we consider as input not only well organized documents, but also news streams, postings in social media, speech-to-text transcriptions from online videos, etc. Such sources may have explicit structure in terms of sentences and paragraphs, but many do not have it and rather form *streams of words (or tokens)* without sentence boundaries. As a result, a gem may not have any sentence structure either. It may consist of concatenated excerpts from different input sources (headings, image captions, incomplete sentences, table excerpts, etc.). However, the gem as a whole should be readable (by a smart human) in a self-contained manner. Table 6.1 shows a gem for the example entity Sutherland Falls (a lesser known waterfall in New Zealand).

The problem of computing entity-specific gems resembles issues in summarization, passage retrieval, document expansion, and novelty detection [Carbonell and Goldstein, 1998; Callan, 1994; Nenkova and McKeown, 2011; Schlaefer et al., 2011]. However, none of the prior work has tackled the combination where salient contents on an entity needs to be focused and novel yet must meet length bounds on the amount of returned information. Passage retrieval and document expansion are intermediate steps for user-oriented tasks (e.g., question answering or entity search); so their outputs are further processed by machines without space constraints. Sum-

Oceania > New Zealand > South Island > Southland > Fiordland National Park > Sutherland Falls named for Donald Sutherland, a prospector who found the falls in 1880. William Quill, whom the lake feeding the falls was named for, is thought to be responsible for the first measurement of the falls which was attained by actually scaling the headwall next to the waterfall. New Zealand is a country which has a very high concentration of waterfalls. Unfortunately many of the best ones are isolated deep in the backcountry and are extremely difficult to access. We are fortunate then that the absolute best waterfall in the country is easily accessed via a very popular trail system but can also be seen easily from the air thanks to the flourishing tourism industry in the area. If you are visiting the South Island of New Zealand, Sutherland Falls should be at the very top of your list of waterfalls to see. Sutherland Falls can only be accessed on foot via the popular Milford Track. Power

Table 6.1: An example gem for Sutherland Falls.

marization and novelty detection (e.g., for news streams), on the other hand, have focused on fixed granularities: sentences or paragraphs. Moreover, many prior methods critically rely on labeled training data. In contrast, our approach is unique in that it

- can tap into arbitrary word-stream sources including news feeds and speech-to-text transcriptions;
- does not use fixed granularities and can identify and compose arbitrary text units into entity's content gems;
- is unsupervised and does not rely on any training data;
- directly supports the end-user task of knowledge maintenance and judiciously avoids overwhelming the user with too much information.

Approach and Contributions. In this chapter we develop a full-fledged method for generating novel salient contents about a given entity, using minimal assumptions about the underlying sources. Our method, called *GEM* for Gem-based Entity-Knowledge Maintenance, identifies salient text pieces of variable granularity, using a budget-constrained optimization problem which decides upon which sub-pieces of an input text should be selected for the final result. Each of these text pieces is an output gem.

GEM represents the input sources as a *stream of words (tokens)*, where each word is associated with a score for estimating how related the context of the word is to the entity of interest. In this way, text fragments that contain densely packed entity-related words will obtain a high score mass. For computing the per-word entity-relatedness scores, we use a short *seed text* about the entity. We assume that the seed text is provided by the user, but we expect it to be merely one or two sentences. Then, for each word in the text stream, we compute a language-model-based similarity between the entity seed text and the context of the word. To capture coherent text excerpts with high score mass while meeting the constraint that a user should

not be overwhelmed with information, we develop a budget-constrained optimization algorithm mapped into an integer linear program. Our algorithm identifies variable-length fragments which stand out by their saliency on the entity and their novelty with regard to the entity seed text.

Our approach of extracting text gems about entities can be applied to any real-world entity with a brief textual description to start from. For example, we could start with a long-tail entity briefly described in Wikipedia, or with a book covered in online communities like `librarything.com` or `shelfari.com`, with a singer's or music band's homepage on the Web, or with a text snippet about a newly emerging entity mentioned in news on recent events. Neither the seed texts nor the candidate documents are necessarily well-organized text. They may contain incomplete sentences, image captions, lists, or social-media "slang". GEM processes all of these tokens uniformly, going beyond related work (e.g., on summarization) geared for sentences or paragraphs.

The fact that we do not pose any restrictions on the inputs, the entity seed and the retrieved source documents, makes GEM highly versatile and widely applicable. For example, we can tap into speech inputs from video footage, by using standard methods for speech-to-text transcription (e.g., using software APIs by Microsoft or Google), and then running GEM on this text input. This text will be noisy because of transcription errors and will not contain any sentence or paragraph structure, yet GEM can handle it smoothly.

In summary, this chapter makes the following novel contributions:

- formulating and modeling the new problem of compiling salient contents for a long-tail or emerging entity, to accelerate knowledge base maintenance;
- devising algorithms for extracting content gems from word-stream sources, by solving a budget-constrained optimization problem;
- conducting experiments that show the benefits of our GEM method in dealing with news streams and with query-based Web pages about long-tail entities;
- demonstrating a use-case with speech-to-text transcriptions as input.

Outline. The rest of the chapter is organized as follows. Section 6.2 presents the computational model of our approach. Section 6.3 describes our interpretation of text sources as a stream of words. Section 6.4 then presents our algorithms for extracting text gems about entities. Section 6.5 and Section 6.6 extend these algorithms by considering novelty and diversity of the extracted gems. Section 6.7, Section 6.8 present our experimental results and an application to audio streams. Finally, Section 6.9 and Section 6.10 provide a discussion and a summary of the chapter.

6.2 Computational Model

Given a short seed text about an entity of interest, our goal is to extract coherent text gems, which are highly related to the input entity. We tackle this problem in the following steps:

- Step 1: Process the seed text of the entity, to build a statistical language model for the entity.
- Step 2: Obtain a set of potentially related text sources.
- Step 3: Represent the text sources as a stream of words, and compute a relatedness score for each word in the stream (Section 6.3).
- Step 4: Run our method for extracting gems on the text stream, to obtain a set of text segments relevant for the entity (Section 6.4). If desired, further run novelty expansion (Section 6.5) or diversification (Section 6.6).

In Step 3, we represent the input text as a stream of words by concatenating the documents. We use this representation in order to handle various text inputs, including speech-to-text transcriptions, social-media postings, and others. For example, speech transcriptions contain only recognized words without sentence markup. Social-media postings or chat conversations have highly non-grammatical language without a clear sentence structure. Therefore, we need to treat each text source as a stream of tokens.

In this work, we focus on devising algorithms for the extraction of entity-related text excerpts from a given set of sources. Orthogonal to this task is the retrieval of as many entity-related sources as possible. This issue is not considered here.

6.3 Relatedness Function

Assume we have an input entity and a text source. In order to select fragments from the text which are highly informative for the entity, we first estimate how related the text is to the entity at each position, by considering individual words (or tokens) as points of interest. Then, we select consecutive parts of the text that contain words highly related to the entity and have a high density of such words.

More formally, we represent the input text for a given entity e by its ordered sequence of words: $S = (w_1, \ldots, w_n)$. Our goal is to estimate the *relatedness* to the entity e at each word $w_i, 1 \leq i \leq n$. However, individual words are meaningful only within their context: a window of surrounding words. So for each word w_i, we consider its *context* $c(w_i)$, which consists of a number of k words before w_i and k words after w_i, that is

$$c(w_i) = (w_{i-k}, w_{i-k+1}, \ldots, w_i, \ldots, w_{i+k-1}, w_{i+k})$$

We associate with each word w_i a *statistical language model* M_{w_i}. We estimate the parameters of M_{w_i} using the words in the context of w_i and their frequencies:

$$P(w|M_{w_i}) = \frac{\mathsf{tf}_{w,c(w_i)}}{\sum_{w' \in c(w_i)} \mathsf{tf}_{w',c(w_i)}}$$

We also build a statistical language model M_e for the entity e using its seed text $s(e)$:

$$P(w|M_e) = \frac{\mathsf{tf}_{w,s(e)}}{\sum_{w' \in s(e)} \mathsf{tf}_{w',s(e)}}$$

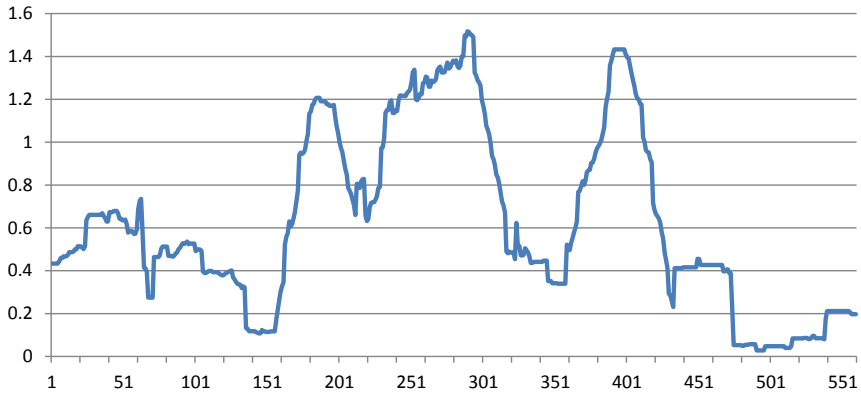

Figure 6.2: Example document with its words on the x-axis, and their relatedness values on the y-axis.

Finally, we compute a *relatedness function* which provides an estimate of how related the context of each word is to the entity:

$$f(w, e) = -KL(M_e||M_w)$$

Here $KL(M_e||M_w)$ is the Kullback-Leibler (KL) divergence between the language model of the entity (M_e) and the language model of the word w (M_w) given by:

$$KL(M_e||M_w) = \sum_t P(t|M_e) \log \frac{P(t|M_e)}{P(t|M_w)}$$

The use of KL divergence for retrieval and ranking of documents is state of the art in IR [Zhai, 2008]. It usually measures the relatedness between a document and a query: low values of the KL divergence $KL(query|doc)$ denote high likelihood that the document generates the query and thus that it is informative/relevant for the query. In our setting the entity is in the role of a query and the context of a given word is used as a document. We also apply Dirichlet smoothing for the context of the given word using the entire Wikipedia as a background corpus.

To extract text gems for the entity we use the relatedness scores of the words in the input text. Since high scores are assigned to words with context relevant to the entity, our goal is to select variable-length segments of the input text that contain many and densely packed words with high relatedness scores. To illustrate this idea, we show in Figure 6.2 an example text and the relatedness function computed over its word positions. We observe that certain parts of the text are more related to the entity than others.

6.4 Extraction of Text Gems

Using the relatedness function we represent the input text as a sequence of word scores. We now explain how to extract gems with high word-score mass. We present

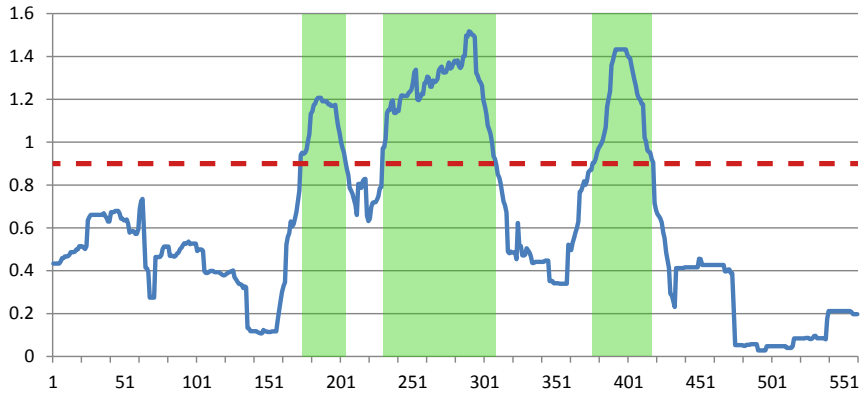

Figure 6.3: Extraction of gems using a threshold value for relatedness. The selected gems are shown in green and the threshold value is represented with red dotted line.

two novel approaches, one based on thresholding on the score distribution over the word positions, and one based on an integer linear program with a specifically designed objective function.

6.4.1 Threshold-based Method

One option to select text gems is to use as an input a *user-specified threshold value*. Then we select text segments of words with relatedness values larger than the specified threshold (see Figure 6.3).

More formally, for a given threshold δ and an input sequence of words $S = (w_1, w_2, \ldots, w_n)$, we select a gem $t = (w_i, w_{i+1}, \ldots, w_j), 1 \leq i \leq j \leq n$ as follows:

- $w_m \in t$ for all positions m such that $i \leq m \leq j$, and
- $f(w_m, e) \geq \delta$, for all $m, i \leq m \leq j$.

Note that for a given threshold value, there can be zero or more text segments extracted from a given text stream.

Gem Budget and Threshold Search. It is a difficult task for an ordinary user to choose suitable threshold values for different entities. The reason is that different entities can have relatedness functions with very different characteristics: smaller versus larger values, more peaks versus rather flat function, etc. To eliminate this complexity, we use as an input parameter a *user-specified budget of words* for the desired total number of words in the extracted gems. This parameter captures the idea that a knowledge-base editor or a user needs a bounded amount of highly informative text excerpts for extending or maintaining the knowledge about an entity; we use this parameter for the rest of chapter.

The threshold-based algorithm searches for the smallest threshold such that the total number of the words in all extracted gems does not exceed the input budget of words. To this end, we observe that the word count in the selected text excerpts

increases monotonically when the threshold decreases. Thus, we can run binary search on the threshold: in each step we first select the gems, following the above requirements, and then count their words. We define the search range by simply taking the minimum/maximum score over all words.

Gem Gaps and Minimum Length. Often two selected gems are within close proximity in the initial text. This means that the textual part in the gap between them is less related to the entity seed text, but the gap text may nevertheless contain new and interesting information about the entity. Recall that for computing the relatedness scores of the words, we use only the language model of the seed text. Thus, if the gap text talks about the entity but does not use its name and generally uses terminology that differs from that of the entity seed text, the relatedness score for the gap text is low. To capture this effect, we consider merging two neighboring gems together with their gap, but do this only when the selected gems are sufficiently close to each other in the input text. We consider two gems to be nearby if the distance between them is less than a certain number of words (e.g., the average length of one or two sentences, which varies between 10 and 60 words). In this case the selected gems are merged into a single one. To avoid very short gems, we require that all selected gems contain at least a certain number of words (e.g., the average length of a sentence).

We implement the above two heuristics by incorporating them into the word counting. Note that this does not violate monotonicity, and thus we can still use binary search.

6.4.2 ILP-based Method

We propose an alternative algorithm for extracting text gems, using an integer linear program (ILP). Similarly to the threshold-based method, the input of the algorithm is a budget of words for the total gem length. However, instead of using heuristics like merging nearby gems, we model the task as an explicit optimization problem and develop a principled solution.

Our goal is to extract the most valuable set of gems T from the stream of words S, while observing the budget B. The ILP formulation needs to capture three requirements:

- 1) the accumulated per-word relatedness scores of the selected gems should be as high as possible (to select only highly informative gems),
- 2) longer gems are preferred over short ones (to select self-contained gems, and to merge nearby gems as the text between them may be relevant to the entity),
- 3) the total length of the gems in T does not exceed the budget B.

We introduce binary decision variables X_i, $i \in \{1, \ldots, n\}$, where n is the number of word positions in the text stream S. $X_i = 1$, if the i-th word belongs to a selected gem, and $X_i = 0$ otherwise. Furthermore, we use binary variables $Y_{i,i+1}, i \in \{1, 2, \ldots, n-1\}$ for consecutive word pairs, which model the idea that we prefer (longer) sequences of words in the selected gems. $Y_{i,i+1} = 1$ if and only if $X_i = 1$ and

$X_{i+1} = 1$. Although the $Y_{i,i+1}$ reflect only two adjacent words, by considering $Y_{i,i+1}$, $Y_{i+1,i+2}$, etc. together, we obtain the intended effect of rewarding the selection of longer sequences.

We can now precisely formulate the optimization problem by the following ILP model:

$$\begin{aligned}
\text{maximize} \quad & \sum_i f(w_i, e) X_i + \alpha \sum_i Y_{i,i+1} \\
\text{subject to} \quad & \sum_i X_i \leq B \\
& Y_{i,i+1} \leq X_i \\
& Y_{i,i+1} \leq X_{i+1} \\
& Y_{i,i+1} \geq X_i + X_{i+1} - 1
\end{aligned}$$

The first summand in the objective function aims to select gems which consist of words with high relatedness scores. The goal is to select only informative text excerpts as gems. The second summand rewards longer sequences of words. To explain this, consider two selected gems, with two words between them, which have low $f(.)$ scores, for example w_i and w_{i+1}. If we merge the two gems, then we set three more Y variables to 1: $Y_{i-1,i} = Y_{i,i+1} = Y_{i+1,i+2} = 1$. This means that the merged gem has larger objective score for a specific choice of the parameter α. The constraints that refer to both X_i and $Y_{i,i+1}$ encode that $Y_{i,i+1}$ is 1 if and only if both X_i and X_{i+1} are set to 1. Finally, the constraint $\sum_i X_i \leq B$ encodes that the total length of the gems should not exceed the budget B.

The parameter α controls the trade-off between the relevance of the gems and their length (see Section 6.5). We show experiments with different values of α in Section 6.7.

6.5 Expanding Gems for Novelty

One drawback of the methods for extracting gems is that they critically rely on the wording of the entity seed text. Since we use the language model of the seed text to compute the relatedness scores of the words, text contents which does not have any words in common with the seed is always assigned very low scores. However, such text parts can still be relevant, especially when the seed text is very short. Furthermore, we would often miss out on novel information about an entity if expressed in terminology very different from the seed text, and capturing such novelty is exactly one of our key goals. Therefore, we present two extensions of our gem extraction methods to capture such novel information.

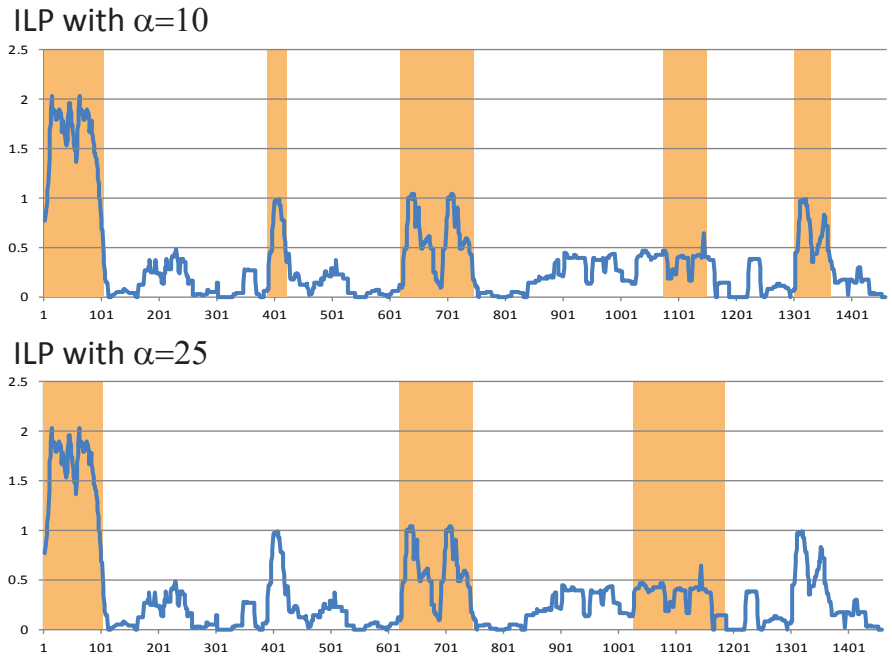

Figure 6.4: Comparison of selected gems using ILP with $\alpha = 10$ and with $\alpha = 25$.

6.5.1 Large α in the ILP-based Method

The parameter α in our ILP method regulates the length of the gems and their relatedness to the entity seed. Low values of α reward highly informative text gems while large values of α reward longer gems. As discussed earlier, often highly informative gems contain only information which is already in the seed text. To capture more novel information about the entity, we increase the value of α.

We give an example for the influence of α on the extracted gems in Figure 6.4, and we show more experimental results in Section 6.7. In Figure 6.4 for $\alpha = 10$ our ILP algorithm selects short but highly relevant text fragments. On the same input text, using $\alpha = 25$ we retrieve fragments which are longer but still contain many related words. We notice, that the second and the fifth gems chosen for $\alpha = 10$ are not selected for $\alpha = 25$. Instead, the forth gem is expanded with more words. To decide which gems to ignore and which gems to expand, the ILP uses the relatedness scores of the words and their neighborhoods. For example, the first gem for $\alpha = 10$ is not expanded for $\alpha = 25$ as the words on its right have very low relatedness values.

6.5.2 Gem Expansion

An alternative approach for finding novel information, which works for the threshold-based method as well, is to expand each gem by appending its surrounding text.

Let k be a parameter that specifies how much we want to grow each gem. To obtain a final set of gems consisting of B words in total, we first extract gems with a budget of B/k words using one of the approaches from Section 6.4. Then we append

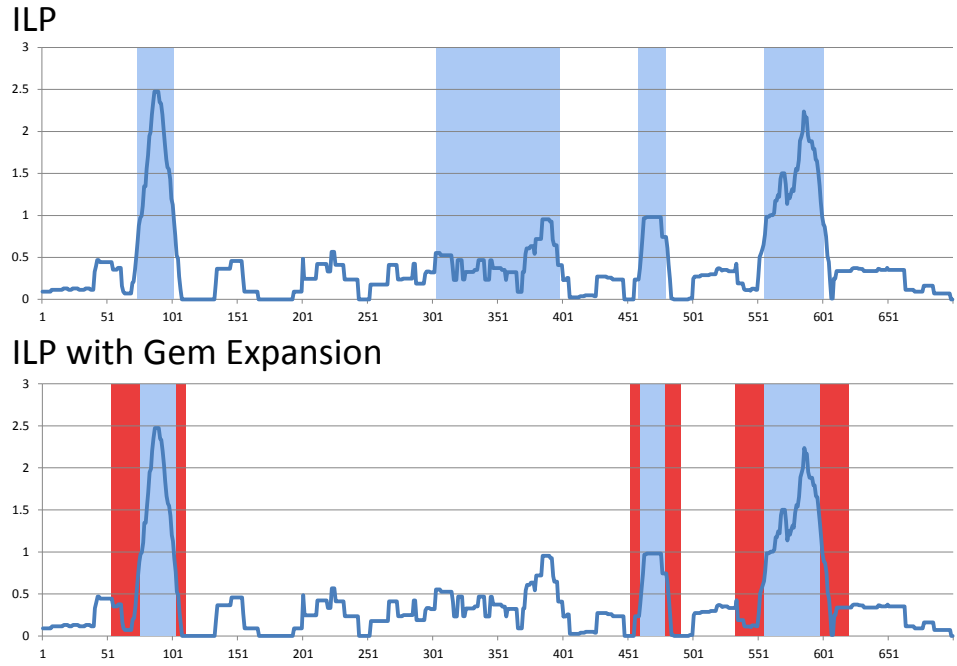

Figure 6.5: The upper image shows gems extracted with ILP for budget of 200 words. The lower image shows ILP with gem expansion: first ILP for budget 100 words extracts the blue gems, then the novelty expansion adds the red parts.

to each gem its surrounding text so that its length grows k times. Note that the left and the right textual parts adjacent to the gem can relate differently to the entity. Thus, we first estimate their relevance to the entity. We initially take both textual parts $k-1$ times larger than the gem length. We compute the relevance of each part $rel(text, entity)$ as the negative KL divergence between the seed language model and the language model of its text. Finally, we choose the lengths of the left and the right parts proportionally to their relevance.

More formally, let t_{left} and t_{right} be the left and right textual parts adjacent to a given gem t. Let t consists of L words, and let both, t_{left} and t_{right}, consist of $(k-1)L$ words. We denote by $M_{t_{left}}$ and $M_{t_{right}}$ the language models of t_{left} and t_{right}, and by M_e the seed text's language model. Then, we compute the scores

$$rel(t_{left}, e) = -KL(M_e||M_{t_{left}})$$

$$rel(t_{right}, e) = -KL(M_e||M_{t_{right}}).$$

We assign L_{left} and L_{right} words to the left and right parts, respectively, such that:

$$L_{left} + L_{right} = (k-1)L$$

$$L_{left}/L_{right} = rel(t_{left}, e)/rel(t_{right}, e).$$

In our experiments, we use $k = 2$, which we found to perform best after testing different values in the range [1.5,3]. In Figure 6.5 we compare the extraction of gems

using the ILP algorithm from Section 6.4.2 and using the expansion technique with $k = 2$. By using expansion, we discard parts of the stream which have high scores (e.g., the second gem in the upper image), and expand gems to the left and to the right aiming to capture information about the entity which is different and novel compared to the seed.

Note that longer gems are expanded with more words than shorter ones. Intuitively, long gems are very prominent for the entity, and thus they can potentially contain more relevant information on the right or on the left side, without matching the terminology in the seed. In contrast, short gems are not consistently relevant for the entity and by expanding them, we can introduce noise.

6.6 Diversification of Gems

By using the entity seed to extract content gems, the algorithms from Section 6.4 select text fragments which are similar to the seed text. Naturally, this can lead to the extraction of gems which are highly similar among each other. In the extreme case, if the seed is very short and/or the Web does not provide much information about the entity, the extracted gems could be almost identical. Since our goal is to extract as much relevant information as possible, we need to extend our extraction methods for *diversification* of gems. In the following, we present two different methods for this purpose.

6.6.1 Diversification Based on Updates

Our first diversification method analyzes the extracted gems for similarity. If there are similar gems, we update the relatedness function such that the parts of the function which correspond to "near-duplicate" gems are assigned with lower scores. Based on this updated function, a new run of the (threshold- or ILP-based) extraction method would likely drop those less valuable text fragments and pick up alternative gems.

In more details, assume a first run of gem extraction produces an initial set of gems. We compute a similarity measure among all pairs of gems. If they are pairwise sufficiently different, then this is our final set of gems. If there are gems that are very similar to others, we pick only the gem with the highest relevance score, and discard the ones that are similar but have lower scores. Next, we update the relatedness function by looking up the word positions of the discarded gems and penalize all these word scores by applying a multiplicative adjustment (e.g., halving the scores). When we subsequently re-run one of our gem-extraction methods, the new relatedness function will likely produce novel gems. This procedure can be iterated a number of times until the returned gems are sufficiently dissimilar among each other.

To measure the relevance of each gem regarding the input entity, we use the KL divergence between the entity seed and the gem:

$$rel(t, e) = -KL(M_e||M_t)$$

where M_e is the language model of the seed text and M_t is the language model of the gem text.

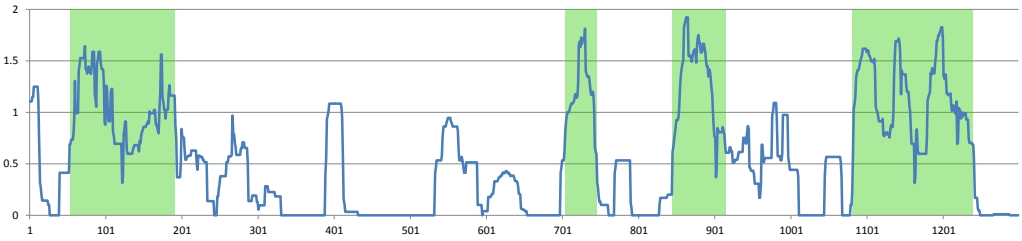

Figure 6.6: Example for extracted gems using ILP for budget of 400 words.

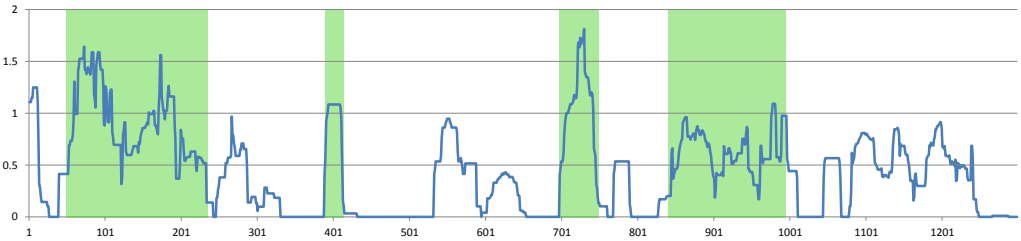

Figure 6.7: Example for extracted gems using diversification based on updates for budget of 400 words on the same input text as in Figure 6.6. Note that the relatedness function is different than the function in Figure 6.6. It is updated for the words corresponding to the last two extracted gems in Figure 6.6 by halving their scores.

To measure if two gems are similar, we use the square root of the Jensen-Shannon divergence between their language models. Formally, for gems t and t', and their language models M_t and $M_{t'}$, the Jensen-Shannon divergence (JSD) is defined as follows:

$$JSD(M_t||M_{t'}) = \frac{1}{2}KL(M_t||M) + \frac{1}{2}KL(M_{t'}||M)$$

where $M = \frac{1}{2}(M_t + M_{t'})$. We use a parameter θ which determines whether two gems are considered similar or not. If the square root of JSD between their language models is less than θ, they are marked as similar to each other. We experimented with different values of θ in the range [0.5,0.8], and $\theta = 0.6$ returned best results.

Figure 6.6 and Figure 6.7 show extracted gems on the same input text using the basic ILP technique from Section 6.4.2 and using the diversification technique from this section, respectively. The diversification technique first analyzes the extracted gems using ILP for pairwise similarity. For example, in Figure 6.6 the last two extracted gems (at positions [846 – 912] and [1082 – 1237]) are near duplicates of other gems. This is why, we update the relatedness function at these positions by halving the scores of the respective words (see Figure 6.7). Then a new run of the ILP model, considering the updated function, produces a new set of gems.

6.6.2 Diversification Based on MMR

The second method for gem diversification is based on the Maximal Marginal Relevance approach (MMR) introduced by [Carbonell and Goldstein, 1998].

The MMR approach re-orders a set of documents, retrieved for a given query, by incrementally choosing the next document which has maximal *marginal relevance*, until a cardinality constraint is met. The marginal relevance of each document is a linear combination of its relevance and its dissimilarity with the already chosen documents. We follow this approach and adapt it to our setting.

Let T be a set of extracted gems for an entity e. To find a subset $S \subseteq T$ of gems which are (i) relevant to the entity, (ii) diverse among each other, and (iii) have a total size that does not exceed the specified budget B, we execute the following steps:

- 1) Initialize the final set of gems S with the gem with highest relevance score.
- 2) Iterate over all gems in $T \setminus S$ and choose the gem t with maximal marginal relevance score:

$$t = \arg\max_{g \in T \setminus S} [\lambda rel(g, e) + (1 - \lambda) \min_{g' \in S} sim(g, g')]$$

- 3) Add the selected gem t to S.
- 4) If the total number of words in the gems in S is less than B, repeat steps (2), (3), and (4). Otherwise, remove the last words from t such that the total number of words in S is not more than B.

To compute the relevance of a gem with respect to an entity, we use the relevance score $rel(.)$ defined in Section 6.6.1. To compute the similarity between two gems, $sim(.)$, we use the square root of JSD between their language models, similarly as in Section 6.6.1. We apply this approach by first running some of the methods from Section 6.4 with larger budget of words, and then iteratively selecting gems, until we reach the desired budget.

6.7 Experiments

We address two experimental scenarios: one with news articles and one with Web search. We compare GEM to its competitors which leverage paragraph and sentence boundaries in the input text. The methods under comparison are:

- the *GEM method*, configured in 5 different modes:
  - **ILP**: the ILP method from Section 6.4.2;
  - **ILP-EXP**: the ILP method from Section 6.4.2 with the gem expansion for novelty from Section 6.5.2;
  - **ILP-UPDATE**: the ILP method from Section 6.4.2 with diversification based on updates (Section 6.6.1);
  - **ILP-MMR**: the ILP method from Section 6.4.2 with diversification based on MMR (Section 6.6.2);
  - **THR-SEARCH**: the threshold-based method from Section 6.4.1;

- **PAR**: a paragraph-based method which first extracts paragraphs using <p> tags, then ranks these paragraphs by the negative KL divergence between the paragraph and the seed, and finally outputs the best paragraphs that fit into the budget;
- **PAR-MMR**: diversification of paragraphs based on the Maximal Marginal Relevance approach;
- **SENT-MMR**: diversification of sentences based on the Maximal Marginal Relevance approach.

The ILP implementation uses the Gurobi Optimizer [Gurobi Optimization, Inc., 2012].

6.7.1 Experiments with News Articles

Experimental Setup

Data. We compiled a set of entities from `wikinews.org`, which consists of 30 emerging events from February to April 2013. 21 of these entities do not have Wikipedia articles. They include "Prague explosion injures dozens", "Ukraine plane crash landing kills five", "Stolen Utahraptor recovered in Australian Capital Territory", etc. We consider such entities as long-tail entities. The remaining events are mentioned in existing Wikipedia articles (e.g., "Pierre Deligne is awarded with Abel prize", "British explorer Ranulph Fiennes leaves Antarctic expedition after frostbite", etc.).

Seed Text and Input Text. We compiled a set of articles by using wikinews articles and their "Sources" links. For each entity we labeled the articles which are relevant for it. For each entity we also chose one of its relevant articles, and used its first one to three sentences as a seed text for the entity. The input text for the compared methods consists of *all* collected articles, except for the articles used as seed texts. This way, the seed texts and the input text are disjoint. In total we have 50 news articles as an input text. The GEM methods use a single stream of words, by first shuffling the 50 news articles and then concatenating them. Table 6.2 shows examples for seed texts.

Sir Ranulph Fiennes has begun his journey home after having to pull out of an expedition across Antarctica in winter because of frostbite.
Investigators have confirmed that the blast last week that ripped open a central Prague office building and injured some 40 people was caused by a gas leak.
Belgian Mathematician wins Abel Prize for Shaping Algebraic Geometry. Pierre Deligne netted the prize, one of the most prestigious in mathematics and worth about $1 million, for proving a deep conjecture about algebraic geometry which has helped to transform number theory and related fields.

Table 6.2: Examples of seed texts for events.

Ambiguity of Entities. Entity names are ambiguous. However, the entities are represented with their seed texts, which despite their brevity, turns out to provide enough information to disambiguate entities with the same or similar names.

Quality Metrics. For each entity we use the articles from the input text which are labeled as relevant for it. We measure which portions of these articles are extracted in the entity gems. We denote by R the text from all labeled relevant articles for a given entity, and by E the text from all gems extracted for this entity. From the word sequences of R and E we have removed all stopwords. For example, assume there is a single labeled relevant article for a given entity with word sequence $R = (w_1, w_2, \ldots, w_{10})$. Assume also, that there is a single extracted gem with text $E = (w_4, w_5, \ldots, w_{12})$. Then $|E \cap R| = 7$. We use the following metrics:

- **Text precision** measures the amount of extracted information (in terms of words) which is relevant to the entity: $\textit{text precision} = |E \cap R|/|E|$.
- **Text recall** measures the amount of relevant information (in terms of words) which is extracted: $\textit{text recall} = |E \cap R|/|R|$.
- **Text F1** is the harmonic mean of text precision and text recall:

$$\textit{text F1} = 2 \cdot \frac{\textit{text precision} \cdot \textit{text recall}}{\textit{text precision} + \textit{text recall}}$$

Results

Table 6.3 shows our experimental results for different sizes of the budget. Comparing ILP with PAR and PAR-MMR, we notice that ILP with $\alpha = 20$ is more effective than the paragraph-based methods by a large margin on all budget sizes. Table 6.4 shows example gems computed by ILP with $\alpha = 20$.

Figure 6.8 shows F1 measure for ILP and ILP-EXP for different values of the parameter α. We notice that for the ILP method F1 increases when the value of α increases. The reason is that ILP with large α selects longer segments which contain more information about the entity. Moreover, it avoids accidentally choosing short segments which contain some words related to the entity, but have different topics otherwise. From Figure 6.8 we also notice that ILP-EXP is always more effective than the paragraph-based baseline.

Table 6.3 shows that THR-SEARCH is inferior than ILP; its results are similar to PAR. Note however, that THR-SEARCH is computationally significantly cheaper than ILP and can potentially process much larger streams. Varying the minimum segment size (in words) p and the maximum distance between merged gems q, we found out that larger values of q slightly improve the results. We obtained best results for $p = 10$ and $q = 60$; we used these parameter values for the results in Table 6.3.

Our experiments show that ILP-MMR slightly improves ILP only for small values of α. For a budget of B words, we first run ILP with budget $2B$, and then follow Section 6.6.2 to choose the best gems that fit into B. F1 for ILP-MMR with $\alpha = 5$ is 0.466, 0.623, 0.634, 0.675 for a budget of 100, 200, 300, 400 words, respectively. Our second diversification method ILP-UPDATE performs similarly to ILP.

Method	Metric	Budget			
		100	200	300	400
ILP ($\alpha = 20$)	text precision	0.944	0.88	0.786	0.696
	text recall	0.367	0.612	0.739	0.82
	text F1	0.483	0.659	0.698	0.693
THR-SEARCH	text precision	0.855	0.755	0.672	0.609
	text recall	0.358	0.491	0.611	0.702
	text F1	0.458	0.54	0.586	0.593
PAR	text precision	0.898	0.781	0.673	0.583
	text recall	0.322	0.496	0.614	0.679
	text F1	0.435	0.555	0.587	0.574
PAR-MMR	text precision	0.887	0.755	0.653	0.565
	text recall	0.314	0.487	0.602	0.669
	text F1	0.426	0.541	0.572	0.559

Table 6.3: Evaluation for news articles.

We observe that the news articles typically contain short paragraphs with a few sentences only. Since short paragraphs do not always contain enough characteristic information about the entity, the baselines do not achieve good results. The same observation holds for sentence-based extraction (with and without diversification). The F1 measure for sentence-based ranking is 0.361, 0.479, 0.511, 0.5 for a budget of 100, 200, 300, 400 words, respectively (SENT-MMR returned worse results).

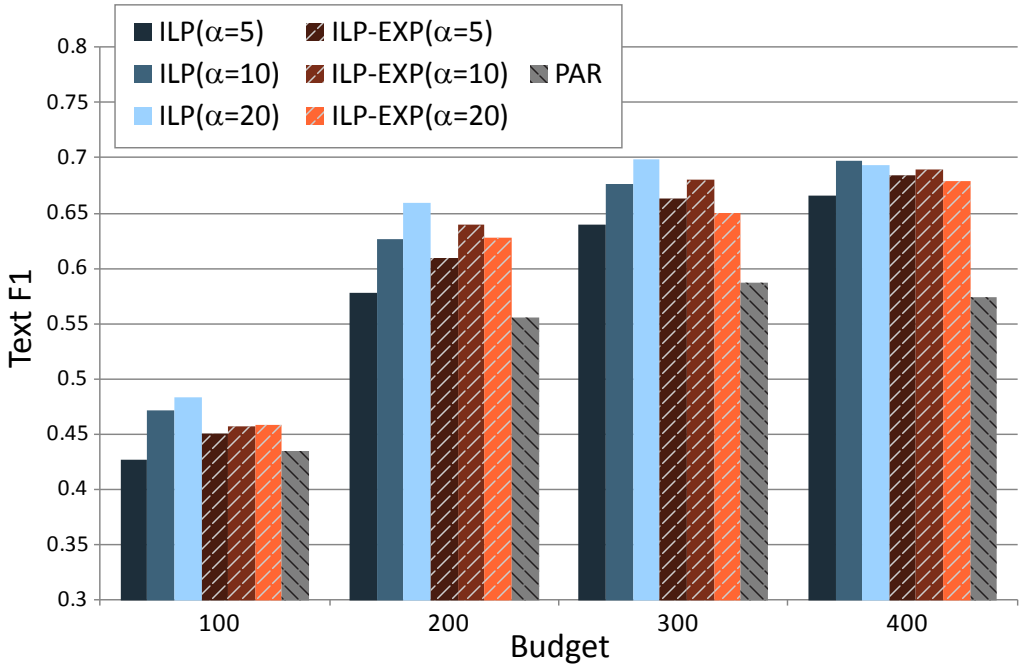

Figure 6.8: Evaluation for news articles of ILP and ILP-EXP with different values of the parameter α.

"British explorer Ranulph Fiennes leaves Antarctic expedition after frostbite": groups like al Shabaab in Somalia and Boko Haram in Nigeria. *Sir Ranulph Fiennes will be evacuated from the Antarctic after suffering severe frostbite, forcing him out of his latest expedition. The British explorer and his fellow adventurers were training to take part in the Coldest Journey, a six-month trek across the continent due to start next month. But the 68-year-old developed frostbite after he had an accident while skiing and had to use his bare hands to repair his ski bindings. Organisers said the decision was made to evacuate Sir Ranulph before the Antarctic winter starts. "The condition is such that he has very reluctantly decided with the support of the team doctor and in the interests of the success of the expedition and its associated aims, to withdraw from Antarctica while the possibility to do so still exists, before the onset of the Antarctic winter," a statement said. But severe weather conditions have halted his evacuation to Cape Town, organisers said. "This plan is currently being hampered due to a blizzard at their present location which is making the first stage of the evacuation impossible," they said. "Until there is a let up in the weather conditions, Fiennes will be unable to leave." Sir Ranulph's finger tips on his left hand were amputated after he sustained severe frostbite during an expedition to the North Pole in 2000. The Coldest Journey expedition, which will continue without Sir Ranulph, will see the team walk 2,000-miles (3,219km) across Antarctica during the winter - the first*
"Pierre Deligne is awarded with Abel prize": those of us who support them". *The Norwegian Academy of Science and Letters awarded Belgian mathematician Pierre Deligne with Abel prize of 2013 for his contributions toward shaping algebraic geometry. The award includes a 6 million Norwegian kroner ($1,026,000, 793,000 euro) prize. Timothy Gowers, a mathematician from Cambridge University, announced the award in Oslo yesterday. The Academy gave the award to Deligne for "seminal contributions to algebraic geometry and for their transformative impact on number theory, representation theory, and related fields". For example, in 1974, Pierre Deligne did a mathematical proof of fourth Weil conjecture, one of properties of Riemann zeta function. This concept is related to analysis of the prime-counting function and the currently unsolved Riemann's hypothesis. During the proof of the Weil conjecture, a concept of l-adic cohomology was introduced. Pierre Deligne said, "The nice thing about mathematics is doing mathematics. The prizes come in addition".* In the United Kingdom, television presenter Derek Batey has

Table 6.4: Extracted gems for events computed by ILP ($\alpha = 20$) for a budget of 200 words. The relevant text is shown in *black*.

Long-tail Entities	Standard Entities
Sutherland Falls Yumbilla Falls	Mahood Falls Hunlen Falls
Nicholas Pippenger David Eppstein	Frances Allen Samson Abramsky
"Lucky (memoir)" by A. Sebold "Rama II" by A. C. Clarke	"A Spot of Bother" by M. Haddon "Netherland" by J. O'Neill

Table 6.5: Examples of long-tail and standard entities.

Sutherland Falls: Sutherland Falls is a waterfall near Milford Sound in New Zealand's South Island. At 580 meters (1,904 feet) the falls were long believed to be the tallest waterfall in New Zealand.
"The Glass Books of the Dream Eaters": Gordon Dahlquist's debut novel is a big, juicy, epic that will appeal to Diana Gabaldon fans (see her quote below) and lovers of literary fantasy, like Keith Donohue's The Stolen Child. The Glass Books of the Dream Eaters begins with a "Dear Jane" letter in which Celeste Temple learns of the end of her engagement. Curiosity leads her to follow her fiance to London where she uncovers a secret.

Table 6.6: Examples of seed texts.

6.7.2 Query-based Experiments

In a second line of experiments, we used Google queries to obtain potentially relevant input text for entities.

Experimental Setup

Data. We compiled three sets of test entities: 25 waterfalls, 25 computer scientists, and 25 books. We consider both long-tail entities and "standard" entities; see Table 6.5 for examples. The former are poorly covered in Wikipedia (usually marked as stub articles); the latter have below-average article lengths.

Seed Text and Input Text. For waterfalls and scientists, the seed text consists of the first one to three sentences from the respective Wikipedia article, ranging between 10 and 50 words in length. For books the seed text was taken from the descriptions section of the respective `librarything.com` page, ranging between 50 and 120 words. Table 6.6 shows examples for seed texts. To gather input texts, we used the entity name (sometimes with a qualifier such as "Netherland Joseph O'Neill" instead of merely "Netherland") for querying Google, fetched the top-10 results, and concatenated their text into a single stream of words. We excluded all pages from Wikipedia and `librarything.com`.

Method \ Budget	400	500	600
ILP	0.474	0.504	0.531
ILP-EXP	0.483	0.516	0.542
ILP-UPDATE	0.477	0.505	0.532
ILP-MMR	0.476	0.514	0.528
THR-SEARCH	0.449	0.487	0.519
PAR	0.457	0.491	0.518
PAR-MMR	0.462	0.49	0.516
SENT-MMR	0.458	0.489	0.517

Table 6.7: Evaluation in terms of phrase recall.

Ground Truth. For all test entities we compiled "ideally informative" texts as ground truth for quality metrics. For waterfalls and scientists we used the rest of the respective Wikipedia article (excluding the seed text). For books we used the respective Wikipedia article. This way, the ground-truth text is disjoint from the seed text for each entity. From these ground-truth texts we extracted noun phrases with the OpenNLP library [OpenNLP], which we used as a gold standard for the quality of the gems. To ensure that phrases are truly informative, we filtered out all phrases that do not match any Wikipedia article title. For the 75 test entities, there are 3,330 noun phrases in total.

Quality Metrics. We use a recall-based metric computed over the relevant noun phrases for each entity. We measure the fraction of ground-truth phrases that are contained in the entity gems. Let R be the ground-truth text for an entity, and G the text in all computed gems for this entity. Let $F(D)$ be a binary vector representing phrases in a text D: $F^i(D) = 1$ if the i-th phrase is in D, and $F^i(D) = 0$ otherwise. Then,

$$phrase\ recall = \frac{\langle F(R), F(G)\rangle}{\langle F(R), F(R)\rangle}$$

where $\langle ., .\rangle$ is the inner product of two vectors. Note that this metric is a special case of the n-gram based metric ROUGE-N [Lin and Hovy, 2003] used in text summarization.

Results

Table 6.7 shows the experimental results for all compared methods, varying the budget of words. For all ILP variants, we show only the case with $\alpha = 20$, as this setting almost always performed best. THR-SEARCH was configured with minimum gem length of 10 words and a maximum distance between merged gems of 60 words, similarly to Section 6.7.1.

The main observation from Table 6.7 is that all our ILP variants outperform all baselines (PAR, PAR-MMR, and SENT-MMR). Among the ILP methods, the differences are relatively small; in most cases ILP-EXP achieves the best results. Larger values of α typically lead to better gem quality. Table 6.1 and Table 6.8 show example gems computed by ILP with $\alpha = 20$ for a budget of 400 words.

As for the inferior performance of the baselines, these methods suffer from their reliance on paragraphs or sentences. In the experiments, we often observed that relatively long paragraphs were selected, quickly exhausting the space budget and thus disregarding other valuable parts of the input texts.

The gains for GEM over the baselines are smaller compared to the experiments from Section 6.7.1. The reason is that the content of arbitrary Web pages retrieved by search engines is very noisy. The input stream for GEM contains a variety of contents such as image captions or entries from Web tables. Such contents can be valuable for the entity but this is not considered in our current evaluation strategy.

"Netherland" by Joseph O'Neill: played in New York since the seventeen-seventies. The man's name is Chuck Ramkissoon, and we first hear of him as a corpse. It is 2006, and the novel's narrator, a Dutch banker named Hans van den Broek, receives a call in London from a New York Times reporter. The remains of Khamraj Ramkissoon–"It's Chuck Ramkissoon," Hans corrects, on the phone-have been found in the Gowanus Canal, and wasn't Hans a business partner of the victim? No, just a friend, Hans says. Later, to his wife, Rachel, Hans describes Chuck as "a cricket guy I used to know. A guy from Brooklyn." We don't realize it yet, but the novel has just unfurled its great theme: this "cricket guy," an Indian from Trinidad, is an American visionary-Chuck, not Khamraj-and cricket is the macula of that mad vision, and "Netherland" has opened where "The Great Gatsby" ends, with its forlorn dreamer dead in the water. The unhappy news prompts Hans to recall his years in New York, and the first time he met Chuck, on a cricket field in Randolph Walker Park, on Staten Island, in the summer of 2002. Hans
Ronald Fagin: IEEE Computer Society Awards "For fundamental and lasting contributions to the theory of databases" Ronald Fagin is an IBM Fellow at the IBM Almaden Research Center. He has won an IBM Corporate Award, eight IBM Outstanding Innovation Awards, an IBM Outstanding Technical Achievement Award, and two IBM key patent awards. He has published well over 100 papers, and has co-authored a book on "Reasoning about Knowledge". He has served on more than 30 conference program committees, including serving as Program Committee Chair of four different conferences. He received his B.A. in mathematics from Dartmouth College, and his Ph.D. in mathematics from the University of California at Berkeley. He was named a Fellow of IEEE for "contributions to finite-model theory and to relational database theory". He was named a Fellow of ACM for "creating the field of finite model theory and for fundamental research in relational database theory and in reasoning about knowledge". He was named a Fellow of AAAS (American Association for the Advancement of Science

Table 6.8: Example gems computed by ILP with $\alpha = 20$ for a budget of 400 words.

6.7.3 Application to Question Answering

We address the use-case of Question Answering (QA) and we show that our GEM method can retrieve text contents which would allow a human user to quickly find the correct answer. We collected 75 questions and their respective answers from the Jeopardy! datasets. As seed texts we used the natural-language-questions and we ran our ILP method from Section 6.4.2 with different values of α. As input text we used the top-10 documents retrieved from Google using the question as a query. We used the phrase recall metric defined in Section 6.7.2. In these experiments we had a single relevant phrase, which was the correct answer to the question. The budget was limited to 50 words. In this way the user would find the answer very quickly, without the need to read a long text.

The average phrase recall for ILP with $\alpha = 5$, $\alpha = 10$, and $\alpha = 20$ is 0.76, 0.76, and 0.747, respectively. The phrase recall for sentence-based ranking is 0.707, and for sentence-based ranking with diversification based on MMR 0.72. We give examples for extracted gems in Table 6.9.

Question: Term for a list of items to be dealt with at a meeting Answer: agenda "written agenda is the list of items to be discussed in a meeting and the time provided to do this. A meeting needs purpose and structure ..."
Question: Large glaciers covering this island nation include Langjokull, Hofsjokull & Vatnajokull Answer: Iceland "... Iceland is renowned for glaciers, covering about 10% of the island. The ten biggest glaciers are: Vatnajokull, Langjokull, Hofsjokull, Myrdalsjokull, Drangajokull ..."
Question: Stewart Island, south of South Island, is this island country's 3rd largest Answer: New Zealand "... Go kayaking in the seas off Stewart Island South of the South Island of New Zealand, and an hour away by ferry from the town of Bluff south of Invercargill, lies the country's third largest island: Stewart Island.
Question: Incan ruins have been found on islands in this lake on the border of Bolivia and Peru Answer: Titicaca "... It is located in southeastern Peru, near the famous Incan ruins at Macchu Picchu. The Islands of Uros are a group of manmade islands floating in Lake Titicaca, on the border of Peru and Bolivia ..."

Table 6.9: Examples for gems computed by ILP ($\alpha = 10$) in a QA setting.

6.8 Application to Audio Streams

We address the following use-case, demonstrating the benefits of GEM being independent of sentence or paragraph boundaries. Given an audio stream of news and an entity or topic of interest, retrieve excerpts of the audio stream which are related to this entity. For example, if users are interested in the latest news about the hurricane Sandy, they could be directly pointed to the specific parts of the audio stream where this topic is discussed.

To solve the problem, we utilize *speech transcriptions* provided by a standard speech recognition system. From the transcriptions, we extract relevant content gems using the methods presented in this chapter. Since speech transcription also returns the time positions of the transcribed words, we associate with each extracted content gem its respective time interval in the input audio. This way, we retrieve audio segments relevant to a given entity.

6.8.1 Experimental Setup

Audio Streams. We collected 10 audio podcasts from NBC Nightly News for the days between December 9 and December 18, 2012. These audio streams provide reports of the most important international events that took place on the respective day. Each audio recording is approximately 20 minutes, some are longer.

Entities. To choose test entities, annotators listened to the complete podcasts and identified salient entities: events or people that are discussed. Our test data includes 10 entities: Nelson Mandela, Susan Rice, hurricane Sandy, Daniel Inouye, North Korea's satellite launch on December 12, gas explosion in West Virginia on December 11, plastic waste in Hawaii, Nicolas Checque, Ravi Shankar, and Jenni Rivera.

Seed Texts. For each test entity we compiled a seed text, which consists of the first 1 to 3 sentences from the respective Wikipedia page (bounding the total number of words to 50). For 4 entities we used seeds from manually retrieved Web pages, when there is no representative article for the person or event in Wikipedia (e.g., Nicolas Checque or the gas explosion in West Virginia on December 11). Examples for seed texts are shown in Table 6.10.

Relevant Audio Segments. While choosing the test entities from the audio podcasts, the annotators labeled time intervals, in terms of minutes and seconds, during which these entities are discussed. These timeframes are considered as ground-truth for the test entities.

The number of relevant timeframes varies across entities. Some topics are mentioned only once (e.g., the death of Ravi Shankar), while other topics are discussed several days (e.g., consequences from the hurricane Sandy). The relevant intervals can be short (1 or 2 minutes) or long (ca. 5 minutes).

Quality Metrics. We use the ground-truth timeframes for measuring gem quality. As each text fragment is associated with start and end time points, we compare the

Gas Explosion in West Virginia (Web seed text): West Virginia explosion of a natural gas line wiped out a wide swath of Interstate 77 and flattened homes. No deaths were caused by the West Virginia natural gas explosion, and federal and state authorities are investigating the cause.
Nelson Mandela (Wikipedia seed text): Nelson Rolihlahla Mandela (born 18 July 1918) is a South African politician who served as President of South Africa from 1994 to 1999, the first ever to be elected in a fully representative democratic election.
Nicolas Checque (Web seed text): Nicolas Checque, the 28-year-old SEAL Team 6 member who was in the helicopter assault to free an American doctor from the Taliban, was killed by a single gunshot to the head.
Ravi Shankar (Wikipedia seed text): Ravi Shankar (7 April 1920 – 11 December 2012) often referred to by the title Pandit, was an Indian musician and composer who played the sitar, a plucked string instrument. He has been described as the best-known contemporary Indian musician.

Table 6.10: Examples for audio entities and their seeds.

intervals of the extracted gems against the ground-truth intervals:

- **Time precision** measures the amount of extracted information (in terms of seconds) which is relevant to the entity. Let E be the set of extracted seconds, and R – the set of labeled relevant seconds. Then, $\textit{time precision} = |E \cap R|/|E|$
- **Time recall** measures the amount of relevant information (in seconds) which is extracted: $\textit{time recall} = |E \cap R|/|R|$
- **Time F1** is the harmonic mean of time precision and time recall:

$$\textit{time F1} = 2 \cdot \frac{\textit{time precision} \cdot \textit{time recall}}{\textit{time precision} + \textit{time recall}}$$

Transcriptions. To transcribe the news podcasts we use the System.Speech namespaces in the Microsoft .NET Framework. As parameters we use the "en-US" culture and the grammar is "DictationGrammar". The result is a single stream of words fed into our GEM methods; the speech recognizer does not provide any markup for sentences or paragraphs.

In order to emulate paragraph-based methods (for comparison), we used two modes for determining speech pauses. We varied the property "SpeechRecognitionEngine.EndSilenceTimeout", which determines how long the speech recognizer waits

until it finalizes the transcription and outputs a *recognized text segment*. Since the audio input is noisy and ambiguous, larger timeout (> 1 sec) results in more robust result. The recognized segments are longer (40 to 60 words) with low variance. Smaller timeout (the default is 150 ms) results in text segments of highly varying lengths, ranging from a couple of words to 50 words.

Competitors. The methods under comparison are:

- **GEM**: our ILP approach from Section 6.4.2 (without novelty expansion or diversification);
- **RT-S**: recognizing text segments with short timeout for pauses;
- **RT-L**: recognizing text segments with long timeout for pauses.

The input to our GEM method is a single stream of words; the information about speech pauses is not used in GEM.

6.8.2 Results

Table 6.12 and Table 6.13 show examples for extracted segments using GEM for budgets of 100 and 300 words, respectively. We notice that regardless of the noisy transcriptions, our approach captures the timeframes related to the entity of interest. In Table 6.11 we systematically compare GEM with its competitors. We configure the ILP method with $\alpha = 10$ ($\alpha = 20$ led to similar results).

We observe that our method extracts gems with high time precision and achieves close to perfect time recall for larger budgets. GEM retrieves almost all relevant information about the entity, while avoiding to overload the user with additional information. In contrast, the two baseline methods retrieve segments which are significantly less related to the entity; even for large budgets their time recall is much lower than the results for GEM. This can be also noticed from the F1 measures of the compared methods.

Figure 6.9 shows extracted segments using GEM and the RT-L baseline for two test entities. We notice that while GEM captures the exact moments when the entity is discussed, the baseline does not always succeed. Typically, GEM produces as many gems as the distinct moments in the audio stream when the entity is discussed. For example, in Figure 6.9 both entities are discussed only once. GEM also produces only single gems for these entities. In contrast, since the baseline is limited to the pseudo-paragraphs given by the speech recognizer, it extracts a mixture of relevant and irrelevant segments. The reason is twofold: (1) the recognized text segments are very noisy as they come from speech transcription, and (2) their lengths are sometimes insufficient to judge if the text is related to the entity or not.

Method	Metric	Budget			
		100	200	300	400
GEM	time precision	0.765	0.77	0.667	0.558
	time recall	0.439	0.755	0.885	0.94
	time F1	0.494	0.692	0.698	0.647
RT-S	time precision	0.663	0.488	0.411	0.341
	time recall	0.407	0.542	0.632	0.674
	time F1	0.445	0.455	0.448	0.41
RT-L	time precision	0.765	0.614	0.488	0.396
	time recall	0.497	0.675	0.762	0.784
	time F1	0.541	0.58	0.543	0.483

Table 6.11: Evaluation for audio streams.

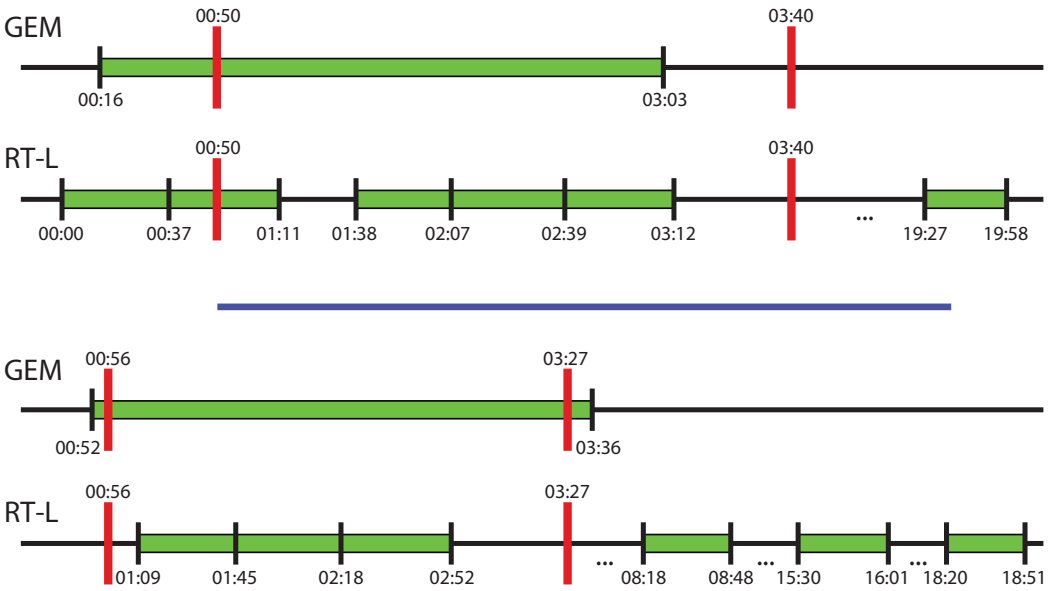

Figure 6.9: Examples for extracted segments from audio streams for two entities using GEM and the RT-L baseline for budget of 300 words. The labeled relevant timeframes are shown with red start and end points. The extracted segments are shown in green.

Ravi Shankar Labeled Relevant Timeframes: (15:53 – 16:20) Extracted Gem (15:22 – 16:26): NBC news New York and again tonight with more on the web including the helpful resources the AARP you can find it all and NBC nightly news staff when we come back is a big complaint about television viewing finally getting taken care of an *ravi shankar has died first time most of this heard the sound of that music of the signs to the beatles and was george harrison first fell in love with the sound of the sitar thanks to shankar virtuoso player and composer people of the sound of the east and west really be composed the score to the film gandhi he died this week following heart surgery on longest surviving family a storm nora Jones ravi shankar was ninety two* was a big ball around here at the today show today you know the theory we all have a double out there somewhere on OS De
Gas Explosion in West Virginia Labeled Relevant Timeframes: (05:32 – 06:38) Extracted Gem (05:24 – 06:14): today is clinical extremely difficult to get a deal done by Christmas offered sadly no surprise *there either liver breaking news tonight of West Virginia were a fiery explosion is the latest reminder that when you have to live near dangerous materials dangerous lines can sometimes happen tonight the problem is a gas pipeline and system go west virginia tom castello spent watching it for us from our washington bureau they taunted by brad the NTSB is on his way to West Virginia to assist in this investigation witnesses say the explosion was so loud that of plane had crashed and in the constant more than all the twenty eighth day as transmission line with flames shooting seventy five feet into the year it took firefighters some time to get into the area because of the intensity of the fire and they've been concerned about the possibility of another explosion*

Table 6.12: Examples for extracted segments from audio transcriptions using GEM for budget of 100 words. Start and end times of timeframes are given in {minutes:seconds} format. The relevant text in the gems with respect to the labeled relevant timeframes is shown in *black*.

Nicolas Checque
Labeled Relevant Timeframes: (04:00 – 06:30) and (11:12 – 11:37)

Extracted Gem (03:56 – 06:20):
church on starting as often as the White House lawn shock sides tonight in a *U.S. navy seal is being remembered as a military hero for his part in rescuing an American doctor kidnapped by the Taliban the ceo is a member of the very same storied seem a special operators sealed teen sex was last in the news for taking out of some of them log jam it was just be at the pentagon has our report on this tonight and it could even bryan president of online defense of the terrain leon panetta had held as navy seal for his valor willing to sacrifice his own life to save another tough a officer nicholas jett was a member of the navy's elite special operations Co. two sets of twenty eight year old is a highly decorated to use the ad that killed Sunday in afghanistan during the hostage rescue mission as ceo set out to rescue American relief worker doctor philip johnson did that last week on the road east of kabul joseph was held hostage of an enemy camp on and logon province in eastern afghanistan under the cover of darkness the navy seal rescue team was flown into the area aboard assault helicopters but as they approach the compound on flawed they came under intense enemy fire for making forty sevens and the machine guns and rocket propelled grenades in the fierce firefight petty officer jack took a single bullet and later died from as well as seals killed seven Taliban fighters captured two others and in the unit that battle racket was lee rescued doctor joseph on our jet was a season combat veteran of both iraq and afghanistan where he earned a bronze star and two other awards for valor income is it never on novell Pennsylvania worry lettered in wrestling at norway not and was widely admired by coaches and classmates a light are never met her well use the new one of those guys on the one team is in the special session has set off a doctor joseph works with an American relief organization morningstar providing free medical care for Afghan civilians he was safely flawed about brother base north of kabul and expected to return to the U.S. and military officials stressed tonight that nobody*

Extracted Gem (11:08 – 11:33):
victory which I know NBC news on the outskirts of level denied defense *officials tell our pentagon correspondent jennifer cessna the U.S. navy seal was killed during the rescue operation of an American doctor in afghanistan his name has not yet been released and happen during a way to save after available joseph and was kidnapped by the Taliban five days ago on the military says the operation*

Table 6.13: Examples for extracted segments using GEM for budget of 300 words. Start and end times of timeframes are given in {minutes:seconds} format. The relevant text in the gems w.r.t. the labeled relevant timeframes is shown in *black*.

6.9 Discussion

6.9.1 Specific Strengths

Independence of Document Structure. Our GEM method does not pose any restrictions on the data input. The entity seed text and the text sources can have different structures or no structure at all. For example, they can contain well-formed sentences as well as incomplete sentences, short phrases, speech transcriptions, entries from Web tables, chat conversations, etc. This makes GEM widely applicable. Our application to audio streams from Section 6.8 demonstrated that GEM is independent of the document structure and can handle highly noisy input data.

Ambiguous Entities. Entity names are ambiguous. However, in our approach we represent the entities with brief textual descriptions. It turns out that such short entity descriptions provide enough information to disambiguate entities with the same or similar names.

6.9.2 Future Work

Application to Entity Disambiguation. Since GEM can extract related information about a given entity and its seed text, it can be used to enhance tasks like entity disambiguation. In some cases the entity disambiguation task fails to correctly disambiguate a given mention because the input text does not provide enough information related to this mention. A possible solution could be to first expand the text which contains the mention of interest by using GEM, and then to disambiguate the mention based on the expanded text.

Improved Expansion of Gems for Novelty. The method for novelty expansion presented in Section 6.5.2 expands gems with more words from the left and from the right in order to capture novel information. However, after the expansion two gems can have non-empty intersection, which means that the words from the intersection form duplicate text fragments. Since, our goal is to extract gems with different information and as much novel information as possible, we need a remedy for such situations. If there are gems with non-empty intersection, we should (1) combine these gems into one gem, and (2) distribute to the left and to the right as many words as the number of the words in the intersection. We would iteratively check if there are gems with non-empty intersection until there are no more such gems. We tested how often expanded gems have non-empty intersection in our experiments from this chapter. Our results show that this occurs infrequently. However, to be able to apply GEM in various settings, we need to extend and adapt our algorithms.

Alternative for Novelty Expansion. An alternative approach to capture more novel information about the entity is to expand iteratively the seed text of the entity. The text sources would be processed in separate partitions. At each step, we would select gems from a given partition using our techniques from Section 6.4. After each step, we expand the seed text with (part of) the selected gems. While selecting gems

from the next partition we use the expanded seed text of the entity. In this way our extraction approach would rely less on the initial seed text and would potentially capture more novel information.

Consideration of Document Structure for GEM. Our approach ignores structures such as paragraphs or sentences in the candidate documents. This is why we lose the implicit semantic information given by these structures. In some cases if a paragraph is highly related to an entity, it could be useful to consider to complete paragraph, instead of parts of it. In our methods, we could utilize the paragraph boundaries during the computation of the relatedness function from Section 6.3. For example, the scores of the words in paragraphs, which are highly related to the entity, can be boosted. It is not clear if this approach would be more effective than our current methods. The reason is that often paragraphs, denoted by HTML <p> tags, are not well formed. Some paragraphs are too long with highly diluted topics, and others are very short, containing only a couple of words.

6.10 Summary

The work in this chapter is a contribution to aid knowledge communities in timely and convenient maintenance of knowledge about entities. Prior work related to this task assumes that input documents are well-formed text with sentence and paragraph markup. Our GEM method does away with this limitation and provides a suite of techniques that can cope with arbitrary input streams of tokens, including news streams or audio transcriptions from videos. The experimental results presented in this chapter demonstrate the viability of this novel approach. We believe that with the ongoing deluge of multimodal contents on the Web, in social media, and in enterprises, such departures from established paradigms are vital to cope with the ever-increasing pace of producing new information and knowledge.

Chapter 7

Conclusion

7.1 Summary

This thesis has first presented two different approaches for an automatic population of knowledge bases with images of entities. The first approach considered facts about each entity of interest from a knowledge base to construct a set of expanded queries posed to image search engines. The search results were merged and ranked based on the overlap of the images from the different expanded queries. Our second approach did not depend on ontological facts and did not construct expanded queries. Instead, it leveraged a seed description of the input entity, from which it extracted entity-characteristic keyphrases. The relevance of each candidate image was based on matches or partial matches of the keyphrases in the Web page that contained the image.

The thesis has also proposed a method for extracting concise text contents about a given entity, which can be recommended to contributors of knowledge bases for the expansion of the knowledge for the respective entity. The text fragments were extracted using a budget-constraint optimization problem without any assumptions on the structure of the text sources. In addition, we have demonstrated the viability of our approach by applying it to speech-to-text transcriptions.

7.2 Outlook

Despite our endeavors in this thesis, there are various research opportunities for future work. We have shown how to find images of entities, but we did not discuss their integration in knowledge bases. While this seems straightforward to do, the entity images can be further organized by subjects. For example, instead of simply assigning a bag of images to a given entity, we could organize the images by ontological facts or keyphrases which were used for their retrieval and ranking. Furthermore, it is interesting to develop an approach for searching images of entities in a knowledge base by using the entity-characteristic facts or keyphrases. The challenge here would be the use of phrases or facts during search which are different than those in the knowledge base.

Our methods focused on finding images of difficult entities. In our work, we defined as difficult the entities for which the search engines did not retrieve satisfactory image

results. Given a set of people or landmarks we tested for which one of them it is difficult to find good pictures. However, we do not know whether the same approach can be applied to other entity types, such as books, events, mathematical terms, awards, songs, etc. It is not clear for which type of entities we should aim at finding images that can be added to a knowledge base. The reason is that not all entity types can be well represented with images.

We applied our methods for extracting text contents for a given entity in three different settings, namely for news articles, with Web search documents, and for the retrieval of audio recordings related to the entity. Other application scenarios can be also studied. For example, our method could enhance the quality of methods for entity disambiguation or fact extraction. Since our method can expand a given text with related information or express a text in a slightly different way, every information extraction technique which can utilize this additional input can be applied. Our method can be used as a preprocessing step for such techniques.

Bibliography

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009a). Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, New York, NY, USA. ACM.

Agrawal, R., Gollapudi, S., Kannan, A., and Kenthapadi, K. (2012). Data mining for improving textbooks. *SIGKDD Explor. Newsl.*, 13(2):7–19.

Agrawal, S., Chakrabarti, K., Chaudhuri, S., Ganti, V., Konig, A. C., and Xin, D. (2009b). Exploiting web search engines to search structured databases. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 501–510, New York, NY, USA. ACM.

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

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007a). Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI, pages 2670–2676.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007b). Open information extraction from the web. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Beis, J. S. and Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1006.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517.

Bhogal, J., MacFarlane, A., and Smith, P. (2007). A review of ontology based query expansion. *Inf. Process. Manage.*, 43(4):866–886.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). DBpedia - A crystallization point for the Web of Data. *Journal of Web Semantics*, 7(3):154–165.

Borodin, A., Lee, H. C., and Ye, Y. (2012). Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st symposium*

on Principles of Database Systems, PODS '12, pages 155–166, New York, NY, USA. ACM.

Brants, T., Chen, F., and Tsochantaridis, I. (2002). Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, pages 211–218, New York, NY, USA. ACM.

Brook Wu, Y.-F., Li, Q., Bot, R. S., and Chen, X. (2006). Finding nuggets in documents: A machine learning approach. *J. Am. Soc. Inf. Sci. Technol.*, 57(6):740–752.

Büttcher, S., Clarke, C. L. A., and Lushman, B. (2006). Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 621–622, New York, NY, USA. ACM.

Callan, J. P. (1994). Passage-level evidence in document retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 302–310, New York, NY, USA. Springer-Verlag New York, Inc.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA. ACM.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the 24th Conference on Artificial Intelligence*, AAAI.

Carmel, D. and Yom-Tov, E. (2010). *Estimating the Query Difficulty for Information Retrieval*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers.

Carmel, D., Yom-Tov, E., Darlow, A., and Pelleg, D. (2006). What makes a query difficult? In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 390–397, New York, NY, USA. ACM.

Chakrabarti, K., Chaudhuri, S., Cheng, T., and Xin, D. (2011). Entitytagger: automatically tagging entities with descriptive phrases. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 19–20, New York, NY, USA. ACM.

Chaudhuri, S., Ganti, V., and Xin, D. (2009). Exploiting web search to generate synonyms for entities. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 151–160, New York, NY, USA. ACM.

Chen, H. and Karger, D. R. (2006). Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM*

SIGIR conference on Research and development in information retrieval, SIGIR '06, pages 429–436, New York, NY, USA. ACM.

Chen, N., Zhou, Q.-Y., and Prasanna, V. (2012). Understanding web images by object relation network. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 291–300, New York, NY, USA. ACM.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 26–33, Stroudsburg, PA, USA. Association for Computational Linguistics.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 659–666, New York, NY, USA. ACM.

Clarke, C. L. A., Cormack, G. V., and Lynam, T. R. (2001). Exploiting redundancy in question answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 358–365, New York, NY, USA. ACM.

Conroy, J. M., Schlesinger, J. D., and O'Leary, D. P. (2006). Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 152–159, Stroudsburg, PA, USA. Association for Computational Linguistics.

Crandall, D. J., Backstrom, L., Huttenlocher, D., and Kleinberg, J. (2009). Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770, New York, NY, USA. ACM.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 299–306, New York, NY, USA. ACM.

Cummins, R. and O'Riordan, C. (2009). Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 251–258, New York, NY, USA. ACM.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):5:1–5:60.

Daumé, III, H. and Marcu, D. (2006). Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 305–312, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE Computer Vision and Pattern Recognition*, pages 248–255.

Drosou, M. and Pitoura, E. (2010). Search result diversification. *SIGMOD Rec.*, 39(1):41–47.

Efron, M., Organisciak, P., and Fenlon, K. (2012). Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 911–920, New York, NY, USA. ACM.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fang, H. and Zhai, C. (2007). Probabilistic models for expert finding. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pages 418–430, Berlin, Heidelberg. Springer-Verlag.

Fellbaum, C. (1998). *WordNet: An Electronical Lexical Database*. The MIT Press, Cambridge, MA.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, IJCAI'99, pages 668–673, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

García-Silva, A., Jakob, M., Mendes, P. N., and Bizer, C. (2011). Multipedia: enriching DBpedia with multimedia information. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 137–144, New York, NY, USA. ACM.

Gollapudi, S. and Sharma, A. (2009). An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 381–390, New York, NY, USA. ACM.

Gurobi Optimization, Inc. (2012). http://www.gurobi.com.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 362–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Harabagiu, S. and Lacatusu, F. (2010). Using topic themes for multi-document summarization. *ACM Trans. Inf. Syst.*, 28(3):13:1–13:47.

Hauff, C., Azzopardi, L., and Hiemstra, D. (2009). The combination and evaluation of query performance prediction methods. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 301–312, Berlin, Heidelberg. Springer-Verlag.

He, B. and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *Proc. Symposium on String Processing and Information Retrieval*, pages 43–54. Springer Verlag.

Hearst, M. A. (1997). Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64.

Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hofmann, K., Tsagkias, M., Meij, E., and de Rijke, M. (2009). The impact of document structure on keyphrase extraction. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1725–1728, New York, NY, USA. ACM.

Jiang, X., Hu, Y., and Li, H. (2009). A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 756–757, New York, NY, USA. ACM.

Jing, Y. and Baluja, S. (2008). Pagerank for product image search. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 307–316, New York, NY, USA. ACM.

Kaszkiel, M. and Zobel, J. (2001). Effective ranking with arbitrary passages. *J. Am. Soc. Inf. Sci. Technol.*, 52(4):344–364.

Kennedy, L. S. and Naaman, M. (2008). Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 297–306, New York, NY, USA. ACM.

Knight, K. (1993). Building a large ontology for machine translation. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 185–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kumar, N. and Srinathan, K. (2008). Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *Proceedings of the eighth ACM symposium on Document engineering*, DocEng '08, pages 199–208, New York, NY, USA. ACM.

Lenat, D. B. (1995). Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38.

Leong, C. W. and Cucerzan, S. (2012). Supporting factual statements with evidence from the web. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 1153–1162, New York, NY, USA. ACM.

Li, H. and Yamanishi, K. (2003). Topic analysis using a finite mixture model. *Inf. Process. Manage.*, 39(4):521–541.

Li, L.-J., Wang, G., and Fei-Fei, L. (2007a). Optimol: automatic Online Picture collecTion via Incremental MOdel Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '07, pages 1–8.

Li, P., Jiang, J., and Wang, Y. (2010). Generating templates of entity summaries with an entity-aspect model and pattern mining. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 640–649, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li, Q., Candan, K. S., and Qi, Y. (2007b). Extracting relevant snippets from web documents through language model based text segmentation. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 287–290, Washington, DC, USA. IEEE Computer Society.

Li, Y., Geng, B., Yang, L., Xu, C., and Bian, W. (2012). Query difficulty estimation for image retrieval. *Neurocomputing*, 95:48–53.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lopez, V., Uren, V. S., Motta, E., and Pasin, M. (2007). AquaLog: An ontology-driven question answering system for organizational semantic intranets. *J. Web Sem.*, 5(2):72–105.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 233–242, New York, NY, USA. ACM.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 404–411.

Misra, H., Yvon, F., Jose, J. M., and Cappe, O. (2009). Text segmentation via topic modeling: an analytical study. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1553–1556, New York, NY, USA. ACM.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA.

Nakashole, N., Theobald, M., and Weikum, G. (2011). Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 227–236, New York, NY, USA. ACM.

Nakashole, N., Weikum, G., and Suchanek, F. (2012). PATTY: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1135–1145, Stroudsburg, PA, USA. Association for Computational Linguistics.

Navarro, G. and Raffinot, M. (2002). *Flexible pattern matching in strings: practical on-line search algorithms for texts and biological sequences*. Cambridge University Press, New York, NY, USA.

Nenkova, A. and McKeown, K. (2011). Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233.

OpenNLP. http://opennlp.apache.org/.

Petkova, D. and Croft, W. B. (2007). Proximity-based document representation for named entity retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 731–740, New York, NY, USA. ACM.

Quack, T., Leibe, B., and Van Gool, L. (2008). World-scale mining of objects and events from community photo collections. In *Proceedings of the 2008 international*

conference on Content-based image and video retrieval, CIVR '08, pages 47–56, New York, NY, USA. ACM.

Radlinski, F. and Craswell, N. (2010). Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 667–674, New York, NY, USA. ACM.

Radlinski, F. and Dumais, S. (2006). Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 691–692, New York, NY, USA. ACM.

Ravi, S. S., Rosenkrantz, D. J., and Tayi, G. K. (1994). Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vision*, 77(1-3):157–173.

Saari, D. G. (2000). The mathematics of voting: Democratic symmetry. *The Economist*, page 83.

Salembier, P. and Smith, J. R. (2001). Mpeg-7 multimedia description schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):748–759.

Salton, G., Allan, J., and Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, pages 49–58, New York, NY, USA. ACM.

Schenkel, R., Broschart, A., Hwang, S., Theobald, M., and Weikum, G. (2007). Efficient text proximity search. In *Proceedings of the 14th international conference on String processing and information retrieval*, SPIRE'07, pages 287–299, Berlin, Heidelberg. Springer-Verlag.

Schlaefer, N., Chu-Carroll, J., Nyberg, E., Fan, J., Zadrozny, W., and Ferrucci, D. (2011). Statistical source expansion for question answering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 345–354, New York, NY, USA. ACM.

Schroff, F., Criminisi, A., and Zisserman, A. (2007). Harvesting image databases from the web. In *Eleventh IEEE International Conference on Computer Vision*, pages 1–8.

Schroff, F., Criminisi, A., and Zisserman, A. (2011). Harvesting image databases from the web. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(4):754–766.

Serdyukov, P., Murdock, V., and van Zwol, R. (2009). Placing flickr photos on a map. In *Proceedings of the 32nd international ACM SIGIR conference on Research and*

development in information retrieval, SIGIR '09, pages 484–491, New York, NY, USA. ACM.

Serdyukov, P., Rode, H., and Hiemstra, D. (2008). Modeling expert finding as an absorbing random walk. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 797–798, New York, NY, USA. ACM.

Shtok, A., Kurland, O., Carmel, D., Raiber, F., and Markovits, G. (2012). Predicting query performance by query-drift estimation. *ACM Trans. Inf. Syst.*, 30(2):11:1–11:35.

Song, R., Taylor, M. J., Wen, J.-R., Hon, H.-W., and Yu, Y. (2008). Viewing term proximity from a different perspective. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 346–357, Berlin, Heidelberg. Springer-Verlag.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA. ACM.

Svore, K. M., Kanani, P. H., and Khan, N. (2010). How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 154–161, New York, NY, USA. ACM.

Taneva, B., Kacimi, M., and Weikum, G. (2010). Gathering and ranking photos of named entities with high precision, high recall, and diversity. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 431–440, New York, NY, USA. ACM.

Taneva, B., Kacimi, M., and Weikum, G. (2011). Finding images of difficult entities in the long tail. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 189–194, New York, NY, USA. ACM.

Taneva, B. and Weikum, G. (2013). Gem-based entity-knowledge maintenance. In *Proceedings of the 22nd ACM international conference on Information and knowledge management*, CIKM '13, New York, NY, USA. ACM.

Tao, T. and Zhai, C. (2007). An exploration of proximity measures in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 295–302, New York, NY, USA. ACM.

Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1958–1970.

Torralba, A., Russell, B. C., and Yuen, J. (2010). LabelMe: Online Image Annotation and Applications. *Proceedings of the IEEE*, 98(8):1467–1484.

Utiyama, M. and Isahara, H. (2001). A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 499–506, Stroudsburg, PA, USA. Association for Computational Linguistics.

van Leuken, R. H., Garcia, L., Olivares, X., and van Zwol, R. (2009). Visual diversification of image search results. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 341–350, New York, NY, USA. ACM.

van Zwol, R., Murdock, V., Garcia Pueyo, L., and Ramirez, G. (2008). Diversifying image search with user generated content. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, MIR '08, pages 67–74, New York, NY, USA. ACM.

Wan, X. and Yang, J. (2008). Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 299–306, New York, NY, USA. ACM.

Yagnik, J. and Islam, A. (2007). Learning people annotation from the web via consistency learning. In *Multimedia Information Retrieval*, pages 285–290.

Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., and Weikum, G. (2012). Natural language questions for the web of data. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 379–390, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yao, B., Yang, X., and Zhu, S.-C. (2007). Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks. In *Proceedings of the 6th international conference on Energy minimization methods in computer vision and pattern recognition*, EMMCVPR '07, pages 169–183, Berlin, Heidelberg. Springer-Verlag.

Zhai, C. (2008). *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Zhai, C. and Lafferty, J. (2006). A risk minimization framework for information retrieval. *Inf. Process. Manage.*, 42(1):31–55.

Zhang, X., Zhang, L., Wang, X.-J., and Shum, H.-Y. (2012). Finding celebrities in billions of web images. *IEEE Transactions on Multimedia*, pages 995–1007.

List of Figures

List of Tables

List of Algorithms