Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Science and Engineering

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Tomáš Jeníček**

Study programme: Open Informatics
Specialisation: Artificial Intelligence

Title of Diploma Thesis: **Canonical Views Extraction from Multimedia Databases Using Non-image Information**

Guidelines:

1. In Wikipedia, define the set of its "landmark" pages. Justify the definition, consider alternatives. The definition should be as broad as possible, i.e. rule out only objects where image based retrieval is poorly defined.
2. Consider one or more comprehensive collections of images.
3. Propose and implement an algorithm that obtains reliably a set of diverse images of the "landmarks";. Images taken at different time of day, time of year, viewpoint and resolutions should be included. Consider inside and outside views.
4. Reduce the set of images to canonical views by clustering.
5. The set of Wiki "landmark" pages and the canonical view set, with description, is one of the outputs of the thesis.
6. Build an image-based retrieval system able to retrieve any Wiki "landmark" from an image acquired in a broad range of conditions.
7. Define an evaluation protocol and measure the performance of the retrieval system.

Bibliography/Sources:

[1] Bryan C. Russell, Ricardo Martin-Brualla, Daniel J. Butler, Steven M. Seitz, Luke Zettlemoyer. 3D Wikipedia: Using online text to automatically label and navigate reconstructed geometry. 2012
[2] Andrej Mikulík, Filip Radenović, Ondřej Chum, Jiří Matas. Efficient Image Detail Mining. 2014
[3] Remco Veltkamp, Hans Burkhardt, Hans-Peter Kriegel, editors. State-of-the-art in content-based image and video retrieval. 2013

Diploma Thesis Supervisor: prof. Jiří Matas Ing., Ph.D.

Valid until the end of the winter semester of academic year 2017/2018

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.
-4-

prof. Dr. Michal Pěchouček, MSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, March 7, 2016

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering

Master's Thesis

Canonical Views Extraction from Multimedia Databases Using Nonimage Information

Bc. Tomáš Jeníček

Supervisor: prof. Ing. Jiří Matas, Ph.D.

Study Programme: Open Informatics

Field of Study: Artificial Intelligence

January 9, 2017

Aknowledgements

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted to prof. Jiří Matas for his patience as well as guidance and also for his support in the final stages of this project. I would like to express my gratitude towards my parents, my brother and my girlfriend for their tolerance which helped me in completing this project.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 9. 1. 2016 ....

Abstract

A comprehensive collection of complex data is beneficial and Wikipedia provides such collection. The ability to recognize any landmark on Wikipedia from an image and present its canonical views is impressive by itself. A comprehensive collection can provide a mean of measuring the false negative rate of image-based retrieval systems.

A set of 357 thousand Wiki Landmarks has been identified among all Wikipedia and Wikidata pages while utilizing 390 languages. An ontology representing a semantic description of Wiki Landmarks was formed by interlinking independent Open Data sources. A dataset of 1.1 million manually annotated Wiki images was retrieved and the main corpus of 131 million external images was obtained from five distinct online image databases - Google Images, Flickr, Yahoo Image Search, Bing Images and Yandex Image Search. All processed data are from 2016.

Images of each landmark consist of a set of diverse views together with metadata related to the scene. For each set of diverse views, canonical views were identified by clustering Wiki images. The result is a database with a complex description of each Wiki Landmark from both semantic and visual point of view. Based on this dataset, an image-based retrieval system able to retrieve any Wiki landmark was built and its evaluation protocol was defined. Taking the first result only, the average accuracy of the system was 48%.

Abstrakt

Ucelená kolekce komplexních dat je užitečná a Wikipedie takovou kolekci poskytuje. Schopnost rozpoznat jakýkoliv významný objekt na Wikipedii podle obrázku a prezentovat jeho kanonické pohledy je samo o sobě působivé. Ucelená kolekce může sloužit k měření chyby typu II u systémů pro vyhledávání pomocí obrazové informace.

Byla získána množina 357 tisíc významných objektů ze všech stránek Wikipedie a Wikidat za využití 390 jazyků. Propojením nezávislých zdrojů otevřených dat byla vytvořena ontologie představující sémantický popis významných objektů z Wiki. Dále byla sestavena množina 1.1 miliónu ručně anotovaných Wiki obrázků a soubor 131 miliónů externích obrázků z pěti různých internetových databází - Google Images, Flickr, Yahoo Image Search, Bing Images a Yandex Image Search. Všechna zpracovaná data jsou z roku 2016.

Obrázky každého významného objektu se sestávají z množiny odlišných pohledů společně s metadaty vztahující se ke scéně. Shlukováním Wiki obrázků obsahující odlišné pohledy byly identifikovány kanonické pohledy. Výsledkem je databáze s komplexním popisem každého významného objektu jak z pohledu sémantiky, tak z pohledu vizuální informace. S využitím těchto dat byl vytvořen systém vyhledávání pomocí obrazové informace, který je schopen vyhledat jakýkoliv významný objekt z Wiki, a pro který byl definovaný evaluační protokol. Při uvažování pouze prvního výsledku měl systém průměrnou přesnost 48%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, large annotated datasets are more valuable than ever. With the current trend of machine learning in computer vision, annotated data are becoming a very important part of the training phase. With the variety of computer vision algorithms, comparing them in a meaningful way helps moving forward in their precision. Another motivation for creating a comprehensive set of landmarks from Wikipedia was the ability to measure the false negative rate of large-scale specific object image retrieval systems.

Retrieving canonical views of specific objects is helpful. Canonical views represent a visual summary of the object and the image retrieval is the basis of a specific object recognition. The major contribution of this work is creating a large-scale image-based retrieval system that can retrieve canonical views and a Wikipedia description of any landmark that has a page on Wikipedia. To achieve this, a rich and well-defined dataset containing all landmarks on Wikipedia and diverse images for every one of them is created.

Landmarks form a subset of specific objects for which canonical views are well-defined. The reason behind this is that a landmark is generally plastic and has an invariable spatial context. Its spatial context is its surroundings constituted of other objects which, in a photo, appear as a background. Other specific objects such as paintings and applied arts objects have a changing spatial context and some of them are also flat, so they provide a single canonical view.

Wikipedia, with 5.3 million articles[1] just in the English-language edition (December 2016), presents one of the most interesting sources of information. During its 15 years of existence, 30 million contributors have created articles for everything that is anyhow interesting. The challenge is processing these information. An advantage of Wikipedia is that other projects reference to its articles which is leveraged in this work - Wikidata is used to augment the descriptive part of Wikipedia and Wikimedia Commons to augment the set of images from Wikipedia articles. To extend the number of images from Wikimedia Commons, five different online image databases - Google Images, Flickr, Yahoo Image Search, Bing Images and Yandex Image Search - are queried. All of this data form the Wiki Landmark dataset.

[1] Wikipedia articles are Wikipedia pages that has encyclopedic information on it. An example of pages that are not articles are redirect pages, disambiguation pages and the main page.

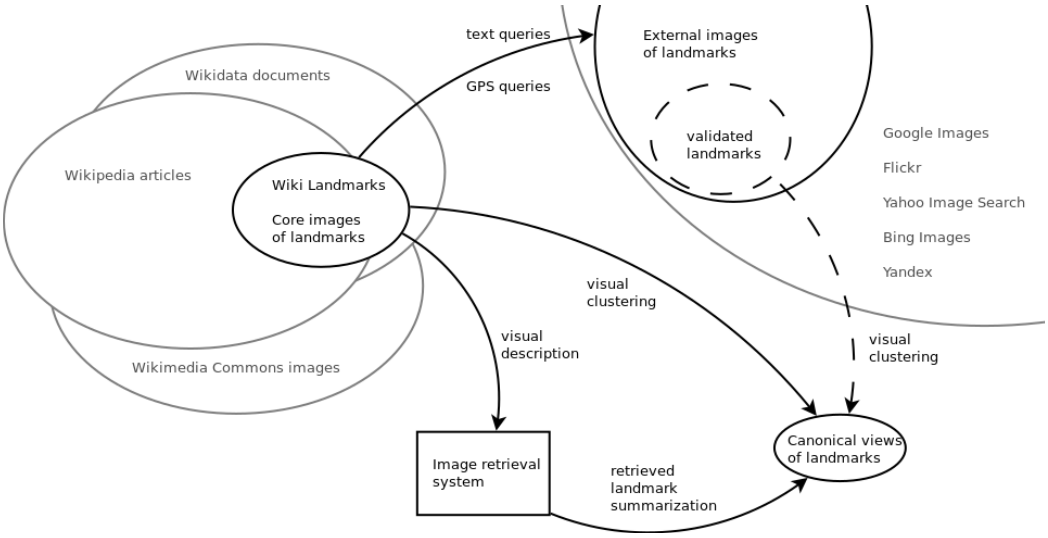


Figure 1.1: The structure of the process of data extraction.

The existing datasets are grey.

1.1 Landmark Object Definition

The landmark object definition is essential and yet not unambiguously obtainable. Dictionaries provide initial definitions which are consolidated, modified with respect to a computer vision and formulated in a way that can be applied to any dataset of structured data.

The dictionary definitions, as well as the accepted definition, contain objective and subjective criteria. An example of an objective criterion is that the landmark has an unchangeable position. These objective constraints can be formulated through knowledge bases that model these properties. Subjective criteria such as that a landmark must be distinguishable and uniquely identifiable must be ensured manually. The main advantage of choosing a project like Wikipedia as the main data source is that the community through the process of selection fulfilled the subjective part of the landmark definition. On Wikipedia, only distinguishable and uniquely identifiable objects have an article.

Differences between multiple definitions are discussed in Section 2 and the decisive definition D2 is provided in Section 2.2.

1.2 Wiki Landmark Pages

Wiki landmark pages are Wikipedia articles augmented with Wikidata documents that describe a landmark. The use of Wikipedia as an information resource is limited by the unorganized nature of Wikipedia articles which surely helped Wikipedia grow, but makes computer processing of contained information a non-trivial task. This is addressed in Section 3.1.

Despite the effort of its contributors, mistakes occur on Wikipedia. This, together with the error of the landmark article identification, was the motivation for Wikipedia augmentation with Wikidata. Linking Wikidata to Wikipedia addressed these issues and together provide a comprehensive set of Wiki Landmarks. This set is a superset of Wikipedia articles about landmarks. The Wikipedia augmentation is described in Section 3.2.

Measuring the false positive rate of the landmark identification can be done manually, but the false negative rate is hard to estimate since there is no exhaustive list of landmarks. To evaluate the Wiki landmark set completeness, datasets outside the Wiki domain are paired with the Wiki landmarks and the coverage percentage is used for a false positive rate estimate. This process is documented in Section 3.3.

1.3 Wiki Core Images

Wiki core images come from Wikimedia Commons and form a Wiki core imageset. Publicly available sources of specific-object image databases are generally not reliable and their relevance differ greatly depending on the query. Wikimedia Commons, maintaining over 35 million media files (December 2016), presents arguably the biggest manually annotated dataset publicly available at the moment. This source of images is used because both Wikipedia and Wikidata contain usually only one representative image per page. After interlinking with Wikidata, it provides a valuable set of diverse images for each landmark reliably. Metadata of images such as the description text or the page the image appeared on are stored together with the image. All details are described in Section 4.

1.4 External Images

The external imageset is an extension of the Wiki core imageset not depending on manual annotations. It is a result of running text-related and gps-related (Flickr only) queries on five different publicly available online image databases. All of the online databases provided the first page of results, but in theory can provide any number of images, exceeding the possibilities of processing, in an order of decreasing relevance. The motivation was to obtain a more diverse set of images by extending images from Wikimedia Commons.

The reasons for choosing specifically these five databases are that Google, Bing and Yahoo are the top three most popular search engines worldwide, Flickr is a popular personal image hosting service and Yandex is the most popular search engine in Russia. Many other services were considered, for example Baidu, the most popular search engine in China, and Pixabay which is a database of public domain images. Both of them were excluded because they do not provide direct links to the images, but rather reference to themselves. This means that for each result they provide, another request retrieving the actual image URL address must be performed.

The process of obtaining external images is detailed in Section 5.

1.5 Wiki Landmark Dataset

The *Wiki Landmark dataset* is the main output of this work. It consists of the set of *Wiki Landmark pages* (from Wikipedia and Wikidata) which is used to retrieve the *Wiki core* *imageset* (from Wikimedia Commons) and the *external imageset* (from five online databases). The relation between these components is indicated in Figure 1.1.

It is a large-scale dataset forming an ontology that describes the Wiki Landmarks in three ways - *semantically*, defining a hierarchy among landmarks and their categories; *descriptively*, providing a set of facts such as GPS coordinates and titles in different languages of the landmark; and *visually*, providing a set of images sharing the relation to the landmark.

Wikipedia, Wikidata and Wikimedia are community projects, so they evolve over time. New articles, documents and images are created, as illustrated in Figure 1.2 on Wikipedia, and existing articles and documents are getting more accurate. To address the data deterioration of completeness, a framework providing tools for data refresh is created. This framework addresses several other issues - changes in the landmark definition, adding more datasets containing landmarks and extending the internal and external image sources - which is all crucial for the long term maintenance. The added value over a simple maintenance is that data obtained using different setups can coexist in the database and can be simply compared.

All details related to the software architecture and the implementation point of view can be found in Section 8.

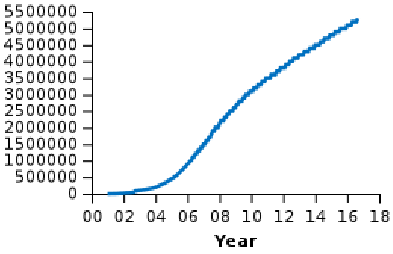


Figure 1.2: The evolution of the number of Wikipedia articles

From the official materials at
https://en.wikipedia.org/w/index.php?title=Wikipedia:Size_of_Wikipedia&oldid=757795678

1.6 Canonical View Identification

Canonical views serve as a mean of presenting large imagesets in a human-processable format as well as a method for speeding up the nearest neighbor algorithm. It is done by representing the imagesets by a small number of representative images. Also, using the same process, isolated views, usually considered outliers, can be identified among the imageset or, less strictly, images can be partitioned according to their representativeness. In this work, canonical views are utilized as an output of the image retrieval system. This, together with data from the Wiki Landmark dataset, provides a complex view on every Wiki landmark.

Clustering algorithms generally does not scale well. This is addressed by partitioning the search space of descriptors using the image annotations, so that only images relevant to one landmark are clustered together. The canonical views are identified as images closest to the cluster centers. The centroid-based mean shift and density-based DBSCAN algorithm are described, augmented and compared in Section 6.

1.7 Large-Scale Image Retrieval

At the top of the Wiki core imageset, an image-based retrieval system is built. The Wiki core imageset contains over 1 million images which was decisive when choosing an image description algorithm. A deep convolutional neural network for image retrieval [12] fine-tuned for landmarks [9] is used for the descriptor computation. The trained network compute global descriptors assigning an image a fixed-size vector. This simplified the image classification substantially reducing it to a nearest neighbor search in the space of visual descriptors. The main advantage of the network are low resource requirements and its biggest disadvantage is that the descriptor is not scale-invariant. This brings many problems because images on the Internet differ in size greatly.

The image retrieval system is able to retrieve any image acquired in a broad range of conditions. It is ensured by the properties of the visual descriptor and by the diversity of the Wiki core images. For the purpose of providing a baseline for consequent retrieval systems, an evaluation protocol is defined and the performance of the system is measured.

Identifying images that can serve as queries to measure error of the retrieval system is also challenging. Queries can be chosen from the core images, but these images are purposely diverse as people generally do not upload same-view images to Wikimedia Commons. Test queries are picked from the external imageset in which case multiple result images can be used for a multi-view query. This provides an interesting alternative to the traditional single-view queries. The description of the system and comparison between single-view and mutli-view queries is presented in Section 7.

1.8 Computational Limits

An extraordinary effort was taken to process the amount of data in every step. For the purpose of image retrieval and canonical view identification, only the core imageset was processed. The number of images in the external imageset (over 100 millions taking approximately 31TB) exceeded the available computing resources greatly. Furthermore, the external imageset contains around 50% images relevant to the landmarks in the top 20 results, so without an advanced validation, irrelevant images would bring an error to the image-based retrieval system and provide false canonical views. Such a validation would raise the computational resource demands significantly. A prototype of such an advanced validation could be based on an interesting approach of image validation through a 3D scene reconstruction described in [9]. The assumption is, nevertheless, that the algorithm has some images of the scene. This method is not a definite solution and the only reliable method of validating images this diverse is the Human-in-the-loop model.

An interesting way of external image validation is utilizing the information about which query retrieved the image. Queries in multiple languages were performed, so images are assigned multiple labels. Taking images with labels that differ filters out many issues such as text ambiguity or incorrect indexing on the database side. This presents a fast and reliable method of external image validation without the dependence on computer vision algorithms. In this work, the output of this external image validation is used to test the image retrieval system as described in Section 7.2.

1.9 Related Work

The idea of utilizing the knowledge base that Wikipedia presents appeared many times, but only a couple of times in terms of exploiting its complex entity description and never concerning images it provides. In [10], they extract names of different objects the landmark consists of from a Wikipedia article, download top 6 images from Google Images, try to identify the best image using text tags and match it into an already built 3D model. The result is an automatically created augmented reality where keywords in the article on Wikipedia are linked with images within an interactive 3D model. Since a 3D model must be provided for the tool, it is necessary to get a big number of different images and compute it using other tools, so it is applicable only for the most popular sights.

Wikipedia article information extraction have two main approaches - *natural language processing* and *infobox* (a fixed-format table) *parsing*. Natural language processing of Wikipedia articles is leveraged in [4] where the information extracted from the text of Wikipedia articles is used as a background knowledge for categorization of text fragments. In [7], the infobox parsing method is described and also a knowledge base DBpedia built solely using the data contained in infoboxes is presented. The original article of the infobox parsing method and the DBpedia knowledge base can be found in [1].

Building a comprehensive collection of images was already performed in the past, as a result of the ImageNet project [3]. It is based on the WordNet project [8] and provide a hierarchical structure among images. The biggest difference from this work is that in case of ImageNet, common objects are processed. The same is the idea of categorizing images into a structure that provides a complex view on the semantics behind them.

On the topic of image partitioning, two interesting papers have emerged. In [5], they compute the probability distribution of image geography location from a collection of 6 millions images. It is focused on used features for database retrieval and image set clustering in order to discard outliers. The other one is [2] where they organize a set of pictures based on text annotations, image features and temporal references. They work with a database of 33 million images where the photographer is known. The outcome is a set of segmented geographic locations with their representative images.

There are many works on the topic of canonical view extraction. In [6], they divide images into different locations and then cluster different views of the same object in order to get representative views of each object. The same goal is achieved in [11], but with an additional focus on assigning the views a textual description. Their approach relies on clustering too, but their visual feature choice is different.

Chapter 2

Landmark Object Definition

The first step towards identifying Wikipedia landmark pages is defining the landmark object. This definition is then adjusted to use Wikipedia article properties and to be applicable to other data sources, such as Wikidata, too. This allows implementing an algorithm that identifies landmarks reliably. In this section, multiple landmark definitions are provided and the decisive definition is formulated and discussed.

2.1 Landmark Definitions

Initial observations were the base of the definition formulation. The definition was formalized using dictionary definitions and finally the set of landmark properties was formally described.

2.1.1 Initial Observations

The landmark object is not easy to formally define. Objective criteria are applied as well as the subjective ones when identifying landmarks. This makes the border of the category subjectively biased and it is very hard to formulate reasons behind the decision whether an object is a landmark. The set of Wikipedia articles presented in Figure 2.2 and Figure 2.3 in the end of this section demonstrate the ambiguity with the intuitive landmark definition.

Generally, the following object categories reflect the inexact nature of a landmark definition. Examples of these object categories can be seen in Figure 2.3.

- Rivers, lakes and mountain ranges
- Parks
- Stations, chimneys or wayside shrines

With rivers, lakes and mountain ranges, probably the size is what determines whether they would be considered landmarks. In case of parks, the border is probably the size together with the uniqueness of the place. An example could be Japanese gardens, generally type of parks which most certainly are landmarks by all means. With stations, chimneys and wayside shrines, it is even more complicated. There exists stations, chimneys and shrines exemplars with architectonic and historic value, exemplars that are indistinguishable one from another, as well as everything in between. The border of the landmark category is probably undecidable here, even subjectively. For the purpose of landmark identification in this work, every category that contains landmarks, including all of mentioned is a landmark by definition D2.

By all definitions, landmarks are objects marking a point usable for navigation. In terms of navigation, local and global landmarks can be distinguished. Local landmarks can help with the position estimation only with the knowledge of their spatial context. An example of a local landmark is a uniquely-shaped stone near the road that can serve as an orientation point when being in one specific area and for a specific group of people aware of it. On the other side, global landmarks are globally unique objects that can act as landmarks without knowing the spatial context of the object. The main focus of this work is on global landmarks.

From the computer vision perspective, the landmark objects can be divided into objects having an outside view, inside view and both. Here, the definition of a landmark is constructed regardless of this division, so objects having images of the inside are also landmarks.

2.1.2 Dictionary Definition

Every landmark definition would be always subjectively biased because not all criteria are objective and could form a border of this object category. The reason behind this is that the definitions are defining landmarks as usable for navigation or having an extraordinary value. This criterion makes it impossible to create an objective definition of landmarks that would be processable by computers. Taken from Wikipedia [1]:

> A landmark is a recognizable natural or artificial feature used for navigation, a feature that stands out from its near environment and is often visible from long distances.
>
> In modern use, the term can also be applied to smaller structures or features that have become local or national symbols.

The Oxford dictionary provides a similar definition [2]:

> An object or feature of a landscape or town that is easily seen and recognized from a distance, especially one that enables someone to establish their location
>
> *North American:* A building or monument of historical importance
>
> *historical:* The boundary of an area of land, or an object marking this

The last definition comes from the Cambridge dictionary [3]:

[1] https://en.wikipedia.org/w/index.php?title=Landmark&oldid=746987484

[2] The decisive part of the definition were taken from https://en.oxforddictionaries.com/definition/landmark. A figurative meaning was excluded.

[3] The decisive part of the definition were taken from https://dictionary.cambridge.org/dictionary/english/landmark. A figurative meaning was excluded.

> A building or place that is easily recognized, especially one that you can use to judge where you are.

Because of the inexact nature of a landmark definition, papers working with the landmark object category do not define it and either make an assumption such as that everything on Flickr having images is some kind of a landmark [6] or avoid it by using a predefined list of landmarks [14].

2.1.3 Landmark Properties Definition

In this work, the landmark object must be formally defined because it is used extensively when identifying landmarks among Wikipedia articles and other data sources, even though the definition is subjective. Each object considered a landmark in this work must satisfy the following properties.

Landmark Property Definition D1

(1) Local uniqueness - unambiguously **distinguishable** from its surroundings

(2) Global uniqueness - uniquely **identifiable** among other landmarks

(3) With an **unchangeable position**

The local uniqueness condition excludes objects not having clear borders, such as forests or seas. The global uniqueness condition excludes common objects such as townhouses and traffic signs. Also, it excludes local landmarks, for example extraordinary trees and stones. These landmarks are distinguishable only with the context of the place and act as landmarks only for a specific group people, usually living nearby. The requirement for an unchangeable position filters out all objects (possibly works of art) where the position can change. These include paintings or applied art objects.

This is a complete definition of a landmark. This definition, nevertheless, includes objects not suitable for image based retrieval. These form a category of objects built specifically with the purpose of marking the land. Examples are highway signs and triangulation stations. These are landmarks from definition but without considering the unique identifier marked on them, they cannot be distinguished one from another. This means only people with the a priori knowledge of their identification can use them as an orientation point. Therefore these objects do not belong to the natural category of landmarks as perceived by every human without any knowledge. To address this, an additional point was added to the definition.

- Being of a **general interest**

Informally, this isolates objects usable for navigation, but not built for the purpose of it. The exception of this informal statement are lighthouses, being a typical example of a landmark.

2.2 Wiki Landmark Definition

When identifying landmarks among Wikipedia articles and other datasets containing landmarks, such as Wikidata, the defined properties must be converted into a specific set of conditions that decide which article fall into the landmark category. Wikipedia, from the essence of being a collaboratively edited encyclopedia, helps a lot with the subjective part of the definition. It is ensured that every specific object having an article on Wikipedia is already identifiable and have an added value. There are no Wikipedia articles about specific common objects. The properties this definition must ensure are that the objects have an unchangeable position (D1-(3)) and that they are well-bordered (part of D1-(1)). A landmark on Wikipedia and in other datasets must satisfy the following conditions.

Wiki Landmark Definition D2

(1) **have a title**

(2) **have GPS coordinates**

(3) **have an immutable location** (or more generally status)

(4) **is well-bordered**

Condition (1) is satisfied for all articles on Wikipedia and is purely formal, but necessary as the title is needed for consecutive steps, main of which is query retrieval (Section 5). Also, it cannot be presumed that in other datasets containing landmarks, such as Wikidata, all landmarks have a title. On Wikipedia, the title is the first underlined heading giving the article a name and also the article-differencing portion of a URL address.

Condition (2) states that a landmark must have GPS coordinates which ensures that the described entity is connected to a specific position. In a Wikipedia article, the GPS coordinates are either in the header of the article or in its infobox, as illustrated in Figure 3.1 that follows.

Condition (3) is necessary because there are articles having GPS coordinates but not being about a place, such as articles about organizations, people or events [4]. The immutable location or status property is demanded through an estimated article category.

Condition (4) was added specifically to exclude objects such as oceanic trenches and tectonic plates. These satisfy conditions (1), (2) and (3) but violate D1-(1).

To ensure conditions (3) and (4) on Wikipedia, the infobox template name was used as an estimate of the article category.

2.2.1 Wikipedia Infobox Templates

The process of a landmark identification in Wikipedia and Wikidata is described in Section 3.1 and Section 3.2.2 respectively. Because a Wikipedia article category is referred to in this definition, an overview of the Wikipedia landmark identification process is provided

[4] The complete list of categories consisting of articles having GPS coordinates but not being about a place can be found in Appendix A

together with examples relevant to the definition. The Wikidata landmark identification is very different from the process described in this section.

A Wikipedia article has an infobox which is a fixed-format table in the top right corner of the page. Each infobox has a template name, common to all infoboxes of a similar topic. Landmarks on Wikipedia are identified by defining landmark template names. The example of template names that are used in articles about landmarks is provided in Figure 2.1. It can be seen that the templates are not strictly categories of articles, but more of a tool for the editors helping them with repetitive work. A complete infobox description is provided in Section 3.1.

Military Memorial	shinto shrine	NRHP	Jain Temple
mill building	Monument	ancient site	UK feature
pre-columbian site	tibetan buddhist monastery	waterfall	wasserfall

Figure 2.1: Examples of infobox template names marking the landmark articles

On top of these template names, a hierarchy is built. For this purpose, the DBpedia ontology is used as a method of template categorization. The use of DBpedia is described in Section 3.1.1. Using these categories, every article under the "place" category was considered a landmark except for the following sub-categories.

- populated place (e.g. a settlement in Wales)
- road (e.g. Adelaide Street, Brisbane)
- celestial body (e.g. Icarus, an Apollo asteroid)
- crater (e.g. Eudoxus, a Martian crater)

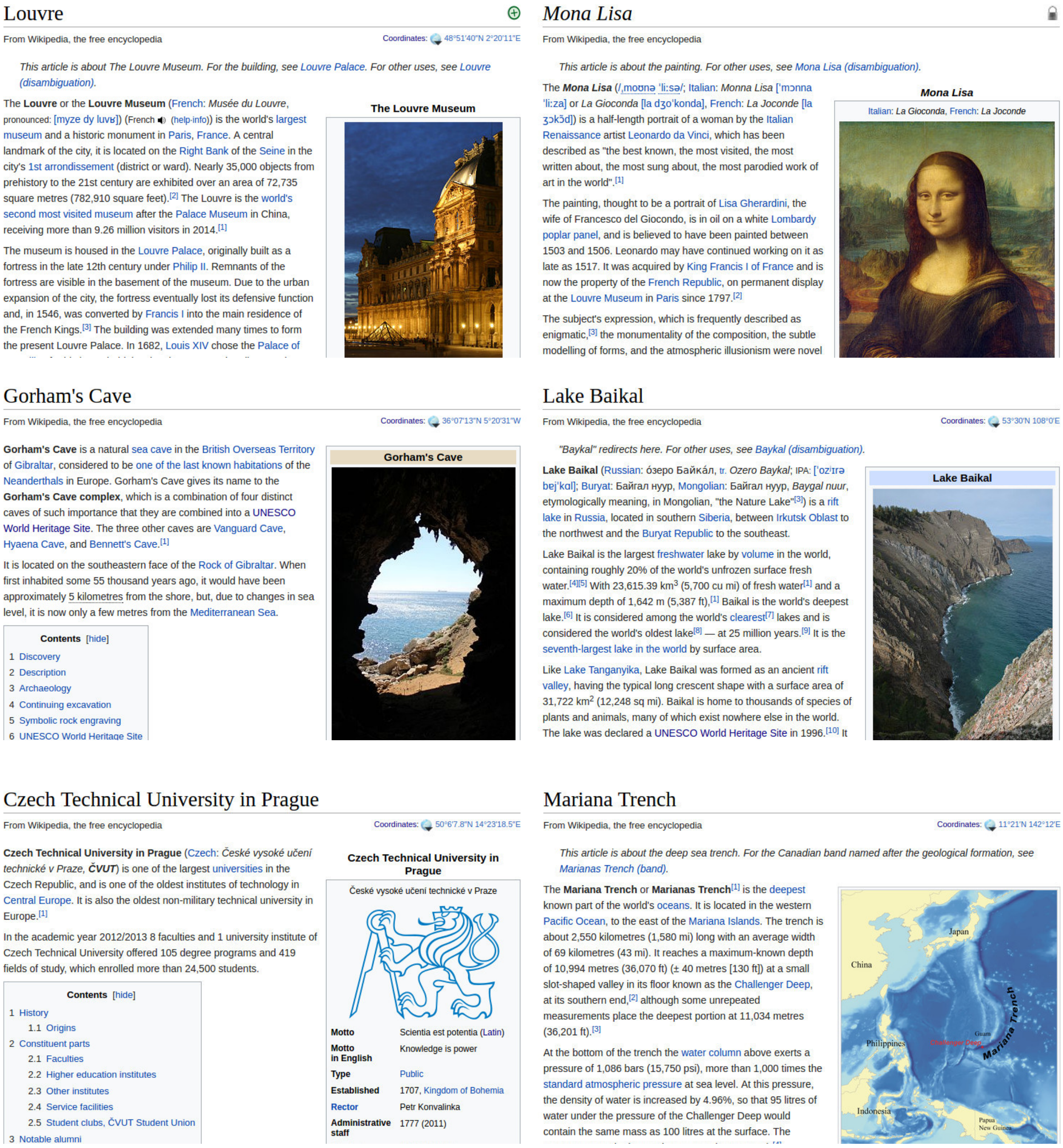
These sub-categories satisfy both definitions D1 and D2 but are the corner cases on which image-based retrieval is poorly defined. These are, therefore, excluded.

2.3 Discussion

Two definitions were presented. The semantic landmark definition in Section 2.1.3 and the Wikipedia landmark definition in Section 2.2. Even though one definition was formed from the other, the means through which they are applied differ. The semantic definition was determined from dictionary definitions and own observations. The Wikipedia definition was built on top of different Wikipedia article aspects. This causes a semantic gap between the two definitions which means that in some cases, these two definitions categorize objects differently and therefore are in a contradiction.

This phenomenon can be demonstrated on rivers. When working with river descriptions on Wikipedia, semantically multiple distinct objects are being described - the river itself and the source or mouth of the river. In some articles, the river itself is described with the information about its source and mouth. In other articles, the river is described as if it was a point, with a single GPS coordinates marking its approximately middle point. This causes that the second river is a landmark by definition D2 while the first one is not. Objects are considered landmarks according to their Wikipedia interpretation which can be sometimes misleading.

An example can be found in Figure 2.4. The Huerva river is a landmark, but Vltava is not based in its Wikipedia article representation. The river Huerva is so short (80 miles) that its complex structure was simplified to a point. The same mistake was encountered in case of some villages. This inconsistence is, nevertheless, something that the editors of Wikipedia unknowingly created when describing different objects. This can be considered a mistake and there is an expectation that this will be fixed in the future as the articles are becoming more and more comprehensive.



Louvre

From Wikipedia, the free encyclopedia

Coordinates: 48°51′40″N 2°20′11″E

This article is about The Louvre Museum. For the building, see Louvre Palace. For other uses, see Louvre (disambiguation).

The **Louvre** or the **Louvre Museum** (French: *Musée du Louvre*, pronounced: [myze dy luvʁ]) (French (help·info)) is the world's largest museum and a historic monument in Paris, France. A central landmark of the city, it is located on the Right Bank of the Seine in the city's 1st arrondissement (district or ward). Nearly 35,000 objects from prehistory to the 21st century are exhibited over an area of 72,735 square metres (782,910 square feet).[2] The Louvre is the world's second most visited museum after the Palace Museum in China, receiving more than 9.26 million visitors in 2014.[1]

The museum is housed in the Louvre Palace, originally built as a fortress in the late 12th century under Philip II. Remnants of the fortress are visible in the basement of the museum. Due to the urban expansion of the city, the fortress eventually lost its defensive function and, in 1546, was converted by Francis I into the main residence of the French Kings.[3] The building was extended many times to form the present Louvre Palace. In 1682, Louis XIV chose the Palace of

The Louvre Museum

Mona Lisa

From Wikipedia, the free encyclopedia

This article is about the painting. For other uses, see Mona Lisa (disambiguation).

The ***Mona Lisa*** (/ˌmoʊnə ˈliːsə/; Italian: *Monna Lisa* [ˈmɔnna ˈliːza] or *La Gioconda* [la dʒoˈkonda], French: *La Joconde* [la ʒɔkɔ̃d]) is a half-length portrait of a woman by the Italian Renaissance artist Leonardo da Vinci, which has been described as "the best known, the most visited, the most written about, the most sung about, the most parodied work of art in the world".[1]

The painting, thought to be a portrait of Lisa Gherardini, the wife of Francesco del Giocondo, is in oil on a white Lombardy poplar panel, and is believed to have been painted between 1503 and 1506. Leonardo may have continued working on it as late as 1517. It was acquired by King Francis I of France and is now the property of the French Republic, on permanent display at the Louvre Museum in Paris since 1797.[2]

The subject's expression, which is frequently described as enigmatic,[3] the monumentality of the composition, the subtle modelling of forms, and the atmospheric illusionism were novel

Mona Lisa

Italian: *La Gioconda*, French: *La Joconde*

Gorham's Cave

From Wikipedia, the free encyclopedia

Coordinates: 36°07′13″N 5°20′31″W

Gorham's Cave is a natural sea cave in the British Overseas Territory of Gibraltar, considered to be one of the last known habitations of the Neanderthals in Europe. Gorham's Cave gives its name to the **Gorham's Cave complex**, which is a combination of four distinct caves of such importance that they are combined into a UNESCO World Heritage Site. The three other caves are Vanguard Cave, Hyaena Cave, and Bennett's Cave.[1]

It is located on the southeastern face of the Rock of Gibraltar. When first inhabited some 55 thousand years ago, it would have been approximately 5 kilometres from the shore, but, due to changes in sea level, it is now only a few metres from the Mediterranean Sea.

Gorham's Cave

Contents [hide]

1 Discovery
2 Description
3 Archaeology
4 Continuing excavation
5 Symbolic rock engraving
6 UNESCO World Heritage Site

Lake Baikal

From Wikipedia, the free encyclopedia

Coordinates: 53°30′N 108°0′E

"Baykal" redirects here. For other uses, see Baykal (disambiguation).

Lake Baikal (Russian: о́зеро Байка́л, tr. *Ozero Baykal*; IPA: [ˈozʲɪrə bɐjˈkɑl]; Buryat: Байгал нуур, Mongolian: Байгал нуур, *Baygal nuur*, etymologically meaning, in Mongolian, "the Nature Lake"[3]) is a rift lake in Russia, located in southern Siberia, between Irkutsk Oblast to the northwest and the Buryat Republic to the southeast.

Lake Baikal is the largest freshwater lake by volume in the world, containing roughly 20% of the world's unfrozen surface fresh water.[4][5] With 23,615.39 km^3 (5,700 cu mi) of fresh water[1] and a maximum depth of 1,642 m (5,387 ft),[1] Baikal is the world's deepest lake.[6] It is considered among the world's clearest[7] lakes and is considered the world's oldest lake[8] — at 25 million years.[9] It is the seventh-largest lake in the world by surface area.

Like Lake Tanganyika, Lake Baikal was formed as an ancient rift valley, having the typical long crescent shape with a surface area of 31,722 km^2 (12,248 sq mi). Baikal is home to thousands of species of plants and animals, many of which exist nowhere else in the world. The lake was declared a UNESCO World Heritage Site in 1996.[10] It

Lake Baikal

Czech Technical University in Prague

From Wikipedia, the free encyclopedia

Coordinates: 50°6′7.8″N 14°23′18.5″E

Czech Technical University in Prague (Czech: *České vysoké učení technické v Praze, **ČVUT***) is one of the largest universities in the Czech Republic, and is one of the oldest institutes of technology in Central Europe. It is also the oldest non-military technical university in Europe.[1]

In the academic year 2012/2013 8 faculties and 1 university institute of Czech Technical University offered 105 degree programs and 419 fields of study, which enrolled more than 24,500 students.

Contents [hide]

1 History
1.1 Origins
2 Constituent parts
2.1 Faculties
2.2 Higher education institutes
2.3 Other institutes
2.4 Service facilities
2.5 Student clubs, ČVUT Student Union
3 Notable alumni

Czech Technical University in Prague

České vysoké učení technické v Praze

Motto	Scientia est potentia (Latin)
Motto in English	Knowledge is power
Type	Public
Established	1707, Kingdom of Bohemia
Rector	Petr Konvalinka
Administrative staff	1777 (2011)

Mariana Trench

From Wikipedia, the free encyclopedia

Coordinates: 11°21′N 142°12′E

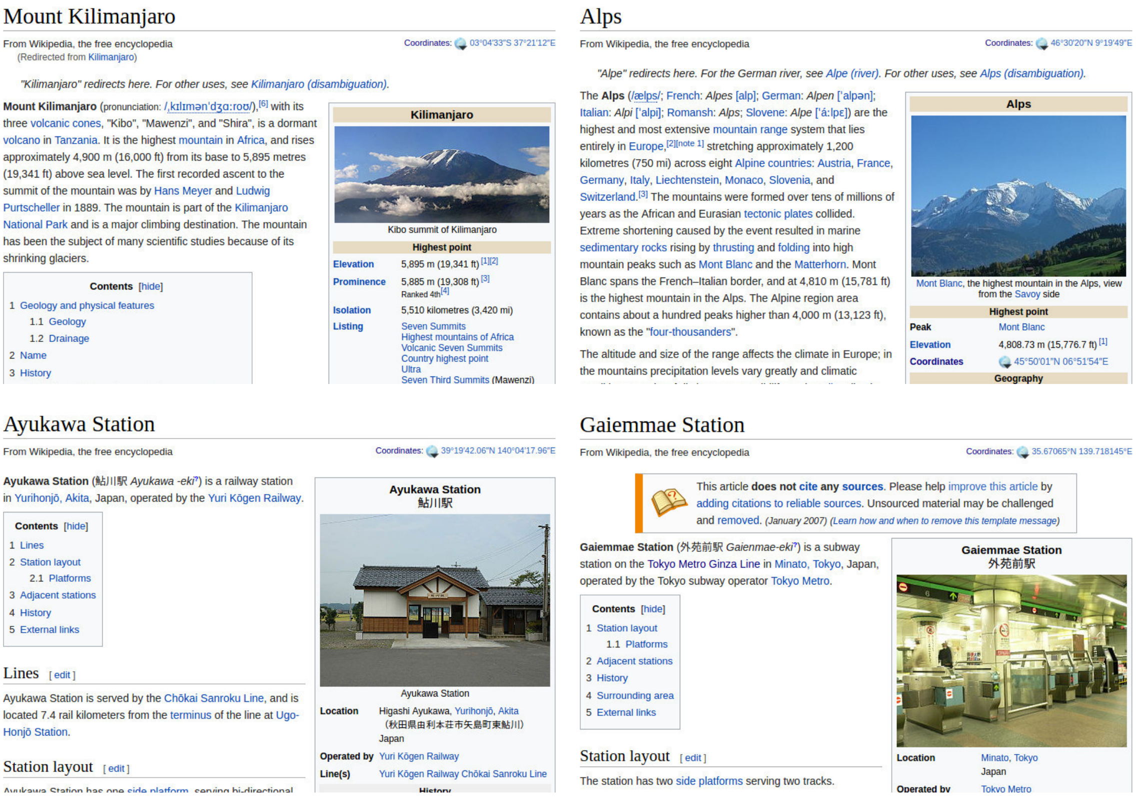
This article is about the deep sea trench. For the Canadian band named after the geological formation, see Marianas Trench (band).

The **Mariana Trench** or **Marianas Trench**[1] is the deepest known part of the world's oceans. It is located in the western Pacific Ocean, to the east of the Mariana Islands. The trench is about 2,550 kilometres (1,580 mi) long with an average width of 69 kilometres (43 mi). It reaches a maximum-known depth of 10,994 metres (36,070 ft) (± 40 metres [130 ft]) at a small slot-shaped valley in its floor known as the Challenger Deep, at its southern end,[2] although some unrepeated measurements place the deepest portion at 11,034 metres (36,201 ft).[3]

At the bottom of the trench the water column above exerts a pressure of 1,086 bars (15,750 psi), more than 1,000 times the standard atmospheric pressure at sea level. At this pressure, the density of water is increased by 4.96%, so that 95 litres of water under the pressure of the Challenger Deep would contain the same mass as 100 litres at the surface. The

Figure 2.2: Six Wikipedia articles, all having GPS coordinates. Articles 1, 3 and 4 are landmarks by definition D2.

Louvre is a landmark. Mona Lisa violates D2-(3) (and D1-(3)) by its position mutability. Gorham's Cave is a landmark despite it has inside views only. Lake Baikal is a landmark despite its size. Czech Technical University violates D2-(3) (and D1-(3)) by its position mutability. Mariana Trench violates D2-(4) (and D1-(1)) by its unclear borders. The classification is in contrast with the intuition.



Mount Kilimanjaro

From Wikipedia, the free encyclopedia
(Redirected from Kilimanjaro)

Coordinates: 03°04′33″S 37°21′12″E

"Kilimanjaro" redirects here. For other uses, see Kilimanjaro (disambiguation).

Mount Kilimanjaro (pronunciation: /ˌkɪlɪmənˈdʒɑːroʊ/),[6] with its three volcanic cones, "Kibo", "Mawenzi", and "Shira", is a dormant volcano in Tanzania. It is the highest mountain in Africa, and rises approximately 4,900 m (16,000 ft) from its base to 5,895 metres (19,341 ft) above sea level. The first recorded ascent to the summit of the mountain was by Hans Meyer and Ludwig Purtscheller in 1889. The mountain is part of the Kilimanjaro National Park and is a major climbing destination. The mountain has been the subject of many scientific studies because of its shrinking glaciers.

Contents [hide]

1 Geology and physical features
 1.1 Geology
 1.2 Drainage
2 Name
3 History

Kilimanjaro	
Kibo summit of Kilimanjaro	
Highest point	
Elevation	5,895 m (19,341 ft) [1][2]
Prominence	5,885 m (19,308 ft) [3] Ranked 4th[4]
Isolation	5,510 kilometres (3,420 mi)
Listing	Seven Summits Highest mountains of Africa Volcanic Seven Summits Country highest point Ultra Seven Third Summits (Mawenzi)

Alps

From Wikipedia, the free encyclopedia

Coordinates: 46°30′20″N 9°19′49″E

"Alpe" redirects here. For the German river, see Alpe (river). For other uses, see Alps (disambiguation).

The **Alps** (/ælps/; French: *Alpes* [alp]; German: *Alpen* [ˈalpən]; Italian: *Alpi* [ˈalpi]; Romansh: *Alps*; Slovene: *Alpe* [ˈáːlpɛ]) are the highest and most extensive mountain range system that lies entirely in Europe,[2][note 1] stretching approximately 1,200 kilometres (750 mi) across eight Alpine countries: Austria, France, Germany, Italy, Liechtenstein, Monaco, Slovenia, and Switzerland.[3] The mountains were formed over tens of millions of years as the African and Eurasian tectonic plates collided. Extreme shortening caused by the event resulted in marine sedimentary rocks rising by thrusting and folding into high mountain peaks such as Mont Blanc and the Matterhorn. Mont Blanc spans the French–Italian border, and at 4,810 m (15,781 ft) is the highest mountain in the Alps. The Alpine region area contains about a hundred peaks higher than 4,000 m (13,123 ft), known as the "four-thousanders".

The altitude and size of the range affects the climate in Europe; in the mountains precipitation levels vary greatly and climatic

Alps	
Mont Blanc, the highest mountain in the Alps, view from the Savoy side	
Highest point	
Peak	Mont Blanc
Elevation	4,808.73 m (15,776.7 ft) [1]
Coordinates	45°50′01″N 06°51′54″E
Geography	

Ayukawa Station

From Wikipedia, the free encyclopedia

Coordinates: 39°19′42.06″N 140°04′17.96″E

Ayukawa Station (鮎川駅 *Ayukawa -eki*?) is a railway station in Yurihonjō, Akita, Japan, operated by the Yuri Kōgen Railway.

Contents [hide]

1 Lines
2 Station layout
 2.1 Platforms
3 Adjacent stations
4 History
5 External links

Lines [edit]

Ayukawa Station is served by the Chōkai Sanroku Line, and is located 7.4 rail kilometers from the terminus of the line at Ugo-Honjō Station.

Station layout [edit]

Ayukawa Station has one side platform, serving bi-directional

Ayukawa Station 鮎川駅	
Ayukawa Station	
Location	Higashi Ayukawa, Yurihonjō, Akita （秋田県由利本荘市矢島町東鮎川） Japan
Operated by	Yuri Kōgen Railway
Line(s)	Yuri Kōgen Railway Chōkai Sanroku Line
History	

Gaiemmae Station

From Wikipedia, the free encyclopedia

Coordinates: 35.67065°N 139.718145°E

This article **does not cite any sources**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(January 2007) (Learn how and when to remove this template message)*

Gaiemmae Station (外苑前駅 *Gaienmae-eki*?) is a subway station on the Tokyo Metro Ginza Line in Minato, Tokyo, Japan, operated by the Tokyo subway operator Tokyo Metro.

Contents [hide]

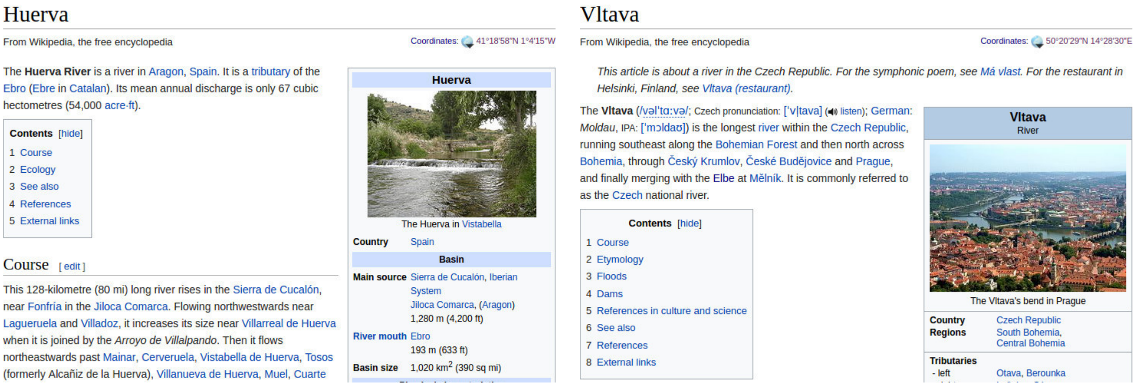
1 Station layout
 1.1 Platforms
2 Adjacent stations
3 History
4 Surrounding area
5 External links

Station layout [edit]

The station has two side platforms serving two tracks.

Gaiemmae Station 外苑前駅	
Location	Minato, Tokyo Japan
Operated by	Tokyo Metro

Figure 2.3: Pairs of Wikipedia articles with a similar topic which are all classified as landmarks by definition D2. Opinion on the two articles on the right would likely differ.



Huerva

From Wikipedia, the free encyclopedia

Coordinates: 41°18′58″N 1°4′15″W

The **Huerva River** is a river in Aragon, Spain. It is a tributary of the Ebro (Ebre in Catalan). Its mean annual discharge is only 67 cubic hectometres (54,000 acre·ft).

Contents [hide]

1 Course
2 Ecology
3 See also
4 References
5 External links

Course [edit]

This 128-kilometre (80 mi) long river rises in the Sierra de Cucalón, near Fonfría in the Jiloca Comarca. Flowing northwestwards near Lagueruela and Villadoz, it increases its size near Villarreal de Huerva when it is joined by the *Arroyo de Villalpando*. Then it flows northeastwards past Mainar, Cerveruela, Vistabella de Huerva, Tosos (formerly Alcañiz de la Huerva), Villanueva de Huerva, Muel, Cuarte

Huerva	
The Huerva in Vistabella	
Country	Spain
Basin	
Main source	Sierra de Cucalón, Iberian System Jiloca Comarca, (Aragon) 1,280 m (4,200 ft)
River mouth	Ebro 193 m (633 ft)
Basin size	1,020 km^2 (390 sq mi)

Vltava

From Wikipedia, the free encyclopedia

Coordinates: 50°20′29″N 14°28′30″E

This article is about a river in the Czech Republic. For the symphonic poem, see Má vlast. For the restaurant in Helsinki, Finland, see Vltava (restaurant).

The **Vltava** (/vəlˈtɑːvə/; Czech pronunciation: [ˈvl̩tava] (listen); German: *Moldau*, IPA: [ˈmɔldaʊ]) is the longest river within the Czech Republic, running southeast along the Bohemian Forest and then north across Bohemia, through Český Krumlov, České Budějovice and Prague, and finally merging with the Elbe at Mělník. It is commonly referred to as the Czech national river.

Contents [hide]

1 Course
2 Etymology
3 Floods
4 Dams
5 References in culture and science
6 See also
7 References
8 External links

Vltava River	
The Vltava's bend in Prague	
Country	Czech Republic
Regions	South Bohemia, Central Bohemia
Tributaries	
- left	Otava, Berounka

Figure 2.4: Wikipedia articles about two rivers. Huerva (left) is a landmark while Vltava (right) is not according to definition D2. Huerva is a landmark because its Wikipedia infobox contains the same fields (such as GPS coordinates of the center) as if it was a point.

Chapter 3

Wiki Landmark Pages

In Wikipedia, a community-driven encyclopedia, almost 30 million contributors are creating the largest knowledge base for humans. The challenge was making this knowledge base accessible for computers. The approach to identify landmarks in Wikipedia is based on exploiting the structured information from an infobox. An infobox is a fixed-format table present in almost every article. For this purpose, the DBpedia project parsing these infoboxes and defining a hierarchy among them was utilized. This led to a rapid extraction speed of landmarks and their descriptions.

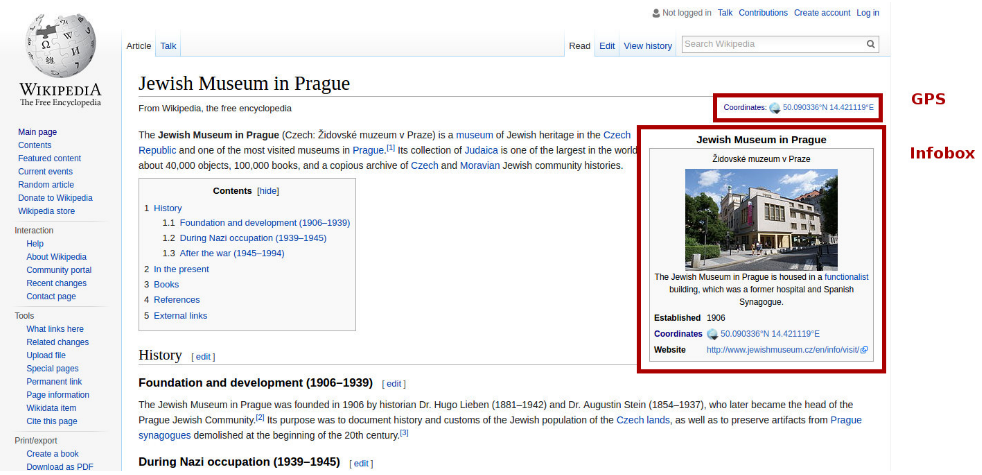
When using only one source of data, data quality issues must be addressed. DBpedia, despite the effort of contributors, does not provide a definite solution to Wikipedia processing. There are mistakes in DBpedia mappings (such as the Statue of Liberty not having GPS coordinates) and also Wikipedia itself is changing and mistakes can occur. Also, only landmarks having an English Wikipedia article were processed because of the computational resource limits. To address these issues, Wikipedia was augmented with Wikidata, another source of data.

The result of interlinking Wikipedia and Wikidata is a set of identified Wiki Landmark pages described by a computer-processable set of facts. Other independent sources of data are used to validate this set of pages. At the end, the Wiki Landmark pages are evaluated and results are presented.

3.1 Wikipedia Landmark Article Identification

Wikipedia itself has no implicit hierarchy - it contains an unorganized set of articles. There are separate pages listing all articles on a specific topic (e.g. List of Baroque residences), but these pages could not be used for the purpose of identifying landmarks, since only a small portion of articles is at some of these lists. As a solution, a so-called *infobox* was exploited as it presents the only piece of structured information in the whole article. It is a box located in the top-right corner of the page containing a summary of facts related to the article, as illustrated in Figure 3.1. Different groups of articles need different structure of this box, so to unify the formatting and reduce repetitive work of Wikipedia editors, each infobox must have a template with fields definition. This means that the infobox is defined only once and then reused on pages of similar topic with different field values only. Even though there are no strict guidelines on creating and using infobox templates, they can serve reliably to separate article sets with a similar field of focus and therefore the infobox template name can be considered an article tag. Also, as the infobox fields are defined in a template, structured data about an article can be parsed from the infobox without much effort. The result is a set of Wikipedia articles annotated with a tag and a set of facts related to an article.

The downside of this process is depending solely on the infobox. Articles without an infobox or with a key data missing in their infoboxes are not categorized as landmarks because of that. With the increasing quality of articles on Wikipedia, this is not considered an issue, as this process of extraction can be re-run any time with up-to-date data containing more complete articles and therefore fixing the issue. Also, this problem is mitigated using another source of data not depending on Wikipedia for landmark extraction.



Jewish Museum in Prague

From Wikipedia, the free encyclopedia

Coordinates: 50.090336°N 14.421119°E

The **Jewish Museum in Prague** (Czech: Židovské muzeum v Praze) is a museum of Jewish heritage in the Czech Republic and one of the most visited museums in Prague.[1] Its collection of Judaica is one of the largest in the world about 40,000 objects, 100,000 books, and a copious archive of Czech and Moravian Jewish community histories.

Contents [hide]

1 History
  1.1 Foundation and development (1906–1939)
  1.2 During Nazi occupation (1939–1945)
  1.3 After the war (1945–1994)
2 In the present
3 Books
4 References
5 External links

Jewish Museum in Prague

Židovské muzeum v Praze

The Jewish Museum in Prague is housed in a functionalist building, which was a former hospital and Spanish Synagogue.

Established	1906
Coordinates	50.090336°N 14.421119°E
Website	http://www.jewishmuseum.cz/en/info/visit/

History [edit]

Foundation and development (1906–1939) [edit]

The Jewish Museum in Prague was founded in 1906 by historian Dr. Hugo Lieben (1881–1942) and Dr. Augustin Stein (1854–1937), who later became the head of the Prague Jewish Community.[2] Its purpose was to document history and customs of the Jewish population of the Czech lands, as well as to preserve artifacts from Prague synagogues demolished at the beginning of the 20th century.[3]

During Nazi occupation (1939–1945) [edit]

Figure 3.1: A Wikipedia article with an infobox and GPS coordinates highlighted

In this case, GPS coordinates are both in the header of the article and in the infobox.

3.1.1 DBpedia as a Structured Wikipedia

In 2007, a project called *DBpedia* founded in the Free University of Berlin and the Leipzig University arose [1]. Its target is to extract Wikipedia information by exploiting data stored in infoboxes, building a semantic interpretation layer on top of Wikipedia articles.

The first version of Wikipedia landmark article identification was based on a raw Wikipedia parsing. It used a simple keyword matching of infobox template names where the landmark keywords were manually defined. DBpedia provides exactly this and also a hierarchy among the infobox templates and a reliable method of infobox field extraction.

To achieve this task, the project uses collaboratively edited mappings between Wikipedia infobox templates with their fields and DBpedia article classes with their properties. This allows to assign articles their categories together with properties such as title, GPS coordinates or date of foundation.

[1] The project is described in [7], its beginnings in [1]

Furthermore, the project also contains a community maintained *ontology* which is a set of relations among article categories (denoted as classes) and their descriptions creating a computer-processable knowledge base. This creates a hierarchy, so if an article is mapped to the class "Station" using a mapping showed in Listing 3.1, it is also in the class "Infrastructure" as stated in the class definition in Listing 3.2.

Listing 3.1: DBpedia template mapping example

```
{{ TemplateMapping
| mapToClass = Station
| mappings =
{{ PropertyMapping | templateProperty = name | ontologyProperty = foaf:name }}
{{ PropertyMapping | templateProperty = type | ontologyProperty = type }}
```

The first five lines of a station mapping linking the *Infobox:station* template with properties *name* and *type* to the *dbo:Station* class with properties *foaf:name* and *dbo:type*. The original mapping has more properties defined.

Listing 3.2: DBpedia class definition example

```
{{Class
| labels = {{label|en|station}}
| rdfs:subClassOf = Infrastructure
| comments = {{comment|en|Public transport station (eg. railway station,
metro station, bus station).}}
| owl:equivalentClass = wikidata:Q719456
}}
```

The *dbo:Station* class definition with its *rdfs:label* and *rdfs:comment* property and *rdfs:subClassOf* and *owl:equivalentClass* relation. The original class has labels and comments in multiple languages.

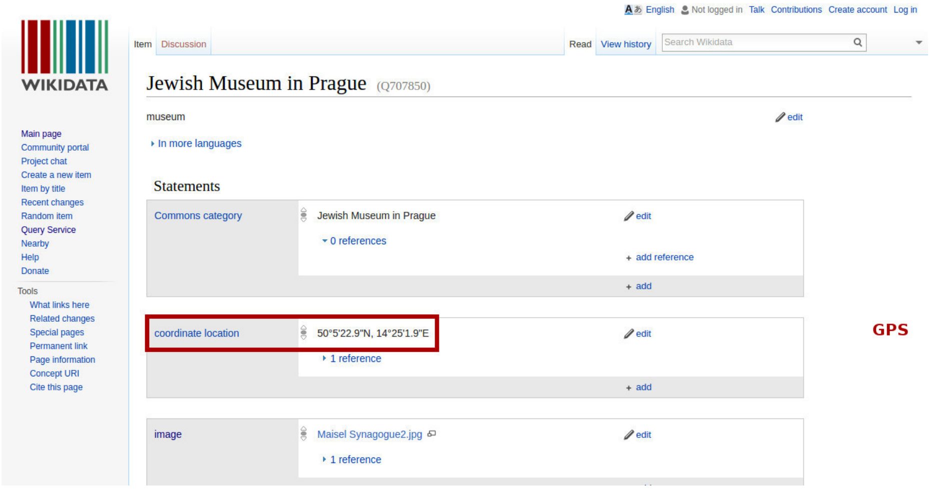
These linkings together with the extracted data fulfills the concept of a decentralized *Linked Data* connecting different *Open Data* sources on the web and together creating a *Semantic Web*, in all of which DBpedia plays a big role.

3.2 Wikipedia Augmentation with Wikidata

To address issues with mistakes in Wikipedia and DBpedia, another source of data is processed. The approach used in case of Wikipedia, categorization through a hierarchy, cannot be used for Wikidata. Instead, an approach based on computing the probability that a category contains landmarks is presented.

3.2.1 Wikidata Description

Wikidata, a project of the Wikimedia Foundation, is a collaboratively edited knowledge base about specific objects and abstract entities, usually associated with Wikipedia articles. It was started in 2012 and as of December 2016, most of its data originally come from Wikipedia, but other than that, it is a separate project where documents are edited independently of Wikipedia. In contrast with Wikipedia, it contains only structured data, and its goal is to serve as a reliable source of statements from which other projects, including Wikipedia, can benefit. Compared to DBpedia where the data are deduced from Wikipedia texts, the structured data is the primary output of the community effort. Wikidata keeps data about its entities in documents, each having a unique URL and containing a set of statements about that entity. An example of such document can be found in Figure 3.2. Despite its structured nature, the data from Wikidata, unlike data from DBpedia, does not form an *ontology* because it has no strict hierarchy and the statements can be in a contradiction [2]. Still, the Wikidata project provides a set of statements about different entities, linkable to Wikipedia, presenting an exploitable knowledge base of landmarks.



Jewish Museum in Prague (Q707850)

museum

In more languages

Statements

Commons category: Jewish Museum in Prague — 0 references

coordinate location: 50°5'22.9"N, 14°25'1.9"E — 1 reference

image: Maisel Synagogue2.jpg — 1 reference

Figure 3.2: A Wikidata document with GPS coordinates highlighted

A Wikidata document equivalent of the Wikipedia article showed in Figure 3.1.

3.2.2 Wikidata Landmark Document Identification

Wikidata does not provide a hierarchy that could be exploited to categorize its documents about entities. It provides only a concept of instances, so that one entity can be an instance of another entity. Example of such a relation is the Prague entity being an instance of the City entity. In this case, the City entity can be pronounced a tag of the Prague entity. For the purpose of categorization, ascendant entity names were considered tags of descendant entities.

Still, considering the size of Wikidata[3], it is not possible to manually annotate each entity that have another entity as its instance. But since both DBpedia and Wikidata are linked to Wikipedia articles, this can serve as a common ground when linking DBpedia and Wikidata together [4]. The pairs of linked entities form an intersection of DBpedia and

[2] From a strictly formal point of view, Wikidata does form an *ontology*, just not a satisfiable one.

[3] The number of entities having another entity as their instance is not easily computable because Wikidata does not distinguish the type of these two. Such a query would mean going through all entities and their class-instance relations.

[4] Since DBpedia version 2016-04, this linking is already part of the DBpedia project.

Wikidata datasets, therefore it is only a subset Wikidata, but it is enough to learn which tags falls into the landmark category.

For a Wikidata tag, Wikidata entity instances whose DBpedia counterpart is and is not a landmark are counted. From the number of landmark entities and non-landmark entities, the probability that a random Wikidata entity with that tag is a landmark is estimated [5]. For the computation, the *binomial proportion confidence interval* is estimated and its lower bound is thresholded as documented in Figure 3.3. The confidence level was set to 68% [6] and the threshold to 0.5. The interpretation is that on the confidence level of 68%, the next sample being a landmark is at least as much probable as the sample not being a landmark. On average, more than 84% of the samples will be landmarks for each tag.

$$
\begin{aligned}
n & & \text{number of samples} \\
n_l & & \text{number of samples being a landmark} \\
p_{ML} = \frac{n_l}{n} & & \text{ML probability of the next sample being a landmark} \\
E = \sqrt{\frac{p_{ML} * (1 - p_{ML})}{n}} * z & & \text{error at a confidence level expressed by z} \\
p_l \in < p_{ML} - E, p_{ML} + E > & & \text{binomial proportion confidence interval} \\
X_l = \{x \,|\, p_{ML} - E > \theta\} & & \text{thresholding the lower bound of the interval by } \theta
\end{aligned}
$$

Figure 3.3: A binomial proportion confidence interval estimation used to Wikidata landmark classification

It is not enough to use the maximum likelihood (ML) probability as the result is independent on the number of samples. This method is not the most precise estimation of the *binomial proportion confidence interval*, caused by the fact that it approximates a binomial distribution error with a normal distribution.

The result of this process is a reliable identification of Wikidata documents about landmarks and a pairing between a subset of these documents and Wikipedia articles. This can be viewed as a set of landmarks corresponding to Wikipedia articles, Wikidata documents or both and each described by a number of statements.

3.3 Measuring Error of Landmark Identification

To get an estimate of false positive and false negative rate, two approaches were used. False positive rate of a random sample of Wikipedia articles and Wikidata documents was measured. The resulting false positive rate was then computed using the equations from Figure 3.3, only, when estimating the ML probability, pseudo-counts were used as shown in Figure 3.4. This must have been done because no false positive samples were encountered in either case. It also reflects the expectation to see some false positives in the random sample.

To measure false negative rate, a different approach must have been chosen. The retrieved dataset of landmarks was linked with other independent datasets and the ratio of items not

[5] Strictly speaking, it is the probability of a Wikidata entity's DBpedia counterpart being a landmark.
[6] The corresponding z-value is 1

$$p_{ML} = \frac{n_l + 1}{n + 2}$$

Figure 3.4: An estimation of a ML probability using pseudo-counts

linked was measured.

3.3.1 Datasets Outside the Wiki Domain

To validate the landmark dataset on an independent source of data, five distinct datasets were obtained and for each item in these datasets, a matching Wikipedia landmark was found, if there existed.

- UNESCO
- EU Open Data - Natura 2000
- EU Open Data - Waterbase - River stations
- EU Open Data - Waterbase - Lake stations
- Czech Railways (CD) - train stations

UNESCO (United Nations Educational, Scientific and Cultural Organization) is an agency of United Nations and since 1978, it publishes regularly a list of landmarks and areas with an extraordinary value called the World Heritage List [7].

EU Open Data Portal is part of the European Union Open Data strategy where various datasets from the EU institutions and agencies are provided [8]. From this portal, Natura 2000, a dataset identifying terrestrial and marine areas of an interest, was obtained [9]. Furthermore, the dataset listing the river and lake stations in the EU [10] from an EU project mapping all Europe's waterbase status and quality was downloaded.

Lastly, a list of train stations operated by the Czech Railways national company was processed [11]. Another dataset was obtained by filtering the train station list keeping only stations having a building and providing some services, as only these probably present landmarks in the sense of being visually unique and distinguishable.

3.3.2 OpenStreetMap Project

Aside from the five datasets used to estimate false negative rate, OpenStreetMap (OSM) data were used to measure how many landmarks are already in its publicly available set of places and to provide an estimation of the extracted GPS coordinates error. The map data

[7] The version from 2016 at http://whc.unesco.org/en/list was used.

[8] The Open Data Portal URL is http://data.europa.eu/euodp/en/data.

[9] Specifically the version from 2016 updated by the data from 2015 was downloaded.

[10] Version denoted as v14 was processed.

[11] It was processed via a form at http://www.cd.cz/en/cd-online/stations-info/default.php in December 2016.

were downloaded [12], then ways (lines) and areas (polygons) together with points (nodes) with no information associated were filtered out, so that only places having a name were kept. Finally, for each Wikipedia landmark a matching OSM place was found, if there existed.

3.3.3 Algorithm of Linking with Wikipedia

The process of searching a matching item had three phases.

1. Find candidates (k-NN on GPS coordinates)
2. Sort candidates (distance in meters)
3. Find a match (label comparison)

Candidates were found using the k-NN algorithm [13] on their GPS coordinates. From the implementation point of view, a KD-tree was used to find the top candidates by their distance within a given radius. The resulting set of candidates was sorted, but as the distance in degrees differ greatly in different latitudes, their distance in meters was computed.

To compute the distance between two points defined by their GPS coordinates, a WGS84 spheroid model of Earth was used which should, at the sea level, have a maximum error of a centimeter in the estimated distance. This error, together with the error caused by the altitude of both points, which is usually different from the altitude at the sea level, and the difference in altitude between the points, is low enough for the purpose of sorting.

Matching places only by their mutual distance does not lead to good results as the differences in two sets of GPS coordinates of the same place but from different sources can differ greatly [14]. It is caused not only by errors in measurements but mainly by the lack of canonical position definition for some object types. This emerges especially in case of objects with a bigger area but without a canonical position definition such as lakes [15]. This was the motivation behind matching objects based on their label. Finding matches by their label is a process where the biggest problem of this linking arises. There are no canonical labels for landmarks and also, each dataset uses labels in a different set of languages.

3.3.4 Label Differences

The problem of label differences, caused by different languages, word order, dialects, or just word form, was to be addressed. First, multilingual labels in overall 390 languages were all used when matching labels. To overcome language nuances inside a language, the following approach minimizing the impact of different word forms and word order was proposed. For a label of a candidate to be considered for a match, it must have had at least three letters. Then, diacritics was stripped out from both labels together with trailing "s" and "n" characters and all consecutive whitespace characters were replaced by a single space. When any of the following conditions was satisfied, a match between labels was pronounced.

[12] A version from 2016-10-16 was used.

[13] The k-value (maximum number of candidates) was 50

[14] A median of the distance for different datasets is provided in Section 3.5 in Table 3.2.

[15] On the contrary, there are big objects (e.g. mountains) that usually have a canonical position associated with them (e.g. their highest or otherwise representing distinguishable point).

1. One label is a substring of the other
2. They have at least two words in common while each word has at least 3 letters

This ensured a robust matching between labels in different languages and from different datasets. Because of working with the full scale of 390 languages, this process proved to be of a higher success rate than a manual linking of labels when understanding English only.

3.3.5 Results

Both DBpedia and Wikidata were estimated a false positive rate of a landmark identification around 1.5% at the 95% level of confidence. False positive rate for the two used data sources is displayed in Table 3.1.

The results of linking independent datasets to the retrieved landmarks with details described in Section 3.3.3 are shown in Table 3.2. Results of linking the retrieved landmarks to OpenStreetMap data are provided in Table 3.3 where the median of a distance being *249m* provide a very good estimation of a GPS coordinates error.

Data source	Size of the sample	Observed false positives	Estimated false positive rate
DBpedia	200	0	**1.5%**
Wikidata	200	0	**1.5%**

Table 3.1: The false positive rate estimation for DBpedia and Wikidata at the 95% level of confidence

Data source	Number of objects	Coverage	Distance median
UNESCO	1,025	82%	91 m
EU - Natura 2000	27,313	22%	4,403 m
EU - River stations	6,264	33%	7,704 m
EU - Lake stations	3,028	32%	1,471 m
CD - building train stations	489	7%	34 m
CD - all train stations	3,855	3%	38 m

Table 3.2: The coverage of independent datasets by Wiki Landmarks

Objects in this case do not have to be landmarks.

Data source	Number of POI	Coverage	Distance median
OpenStreetMaps	$45 * 10^6$	**48%**	**249 m**

Table 3.3: Wiki Landmarks coverage by the OpenStreetMap data

POI denotes a point of interest which is a hypernym of a landmark.

3.4 Parsing Implementation

The first step towards building the dataset of landmarks was to obtain and process the two main sources of structured data. This demanded a specific approach since both datasets were large in size and advanced querying was to be performed on them. The DBpedia and Wikidata parsing was performed together using the same process, so it is described in this section for both sources of data.

3.4.1 Obtaining DBpedia and Wikidata Datasets

The data for both DBpedia and Wikipedia were obtained via a dump. Having a data from a dump instead of working with their current version brings many problems, the main of which is data staleness. Furthermore, as the two dumps were not from the same date (not even from the same month), it must had been ensured that Wikidata dump is more recent than DBpedia dump because Wikidata was referenced from DBpedia. The most recent dumps available were used - from April 2016 in case of DBpedia and from August 2016 in case of Wikidata. The motivation for using the dumps is that neither project provide an up-to-date data export and their online API is not usable for a data retrieval of this size and complexity.

Both dumps were in a TTL format where every line corresponds to a triplet which is a list of three elements. These elements represent the subject, verb and the object in this order. The subject with the verb is always represented by a URI uniquely identifying the resource. Object can be either a URI of a resource or a value of a string or other defined data type (mostly used are XML schema data types). This way, a triplet constitutes a statement about its subject.

3.4.2 Preprocessing DBpedia and Wikidata

These files must had been reduced because of their size. As not all of the statements were necessary to process, a simple preprocessing consisting of filtering out needless statements was applied. The results of this preprocessing, together with the dataset sizes, are presented in Table 3.4.

Data source	Size	Number of triplets
DBpedia	202 GiB	$1.5 * 10^9$
Wikidata	189 GiB	$1.4 * 10^9$
DBpedia and Wikidata merged	391 GiB	$2.9 * 10^9$
After pre-processing	**71 GiB**	$\mathbf{605 * 10^6}$

Table 3.4: The size of data for different data sources and phases of processing

3.4.3 Making the Datasets Indexed for Queries

After the pre-processing, all triplets were loaded into a database for further indexing. Two databases were tried for this purpose - Apache Jena Fuseki and OpenLink Virtuoso. They both work with data in a TTL format, provide a SPARQL API allowing for complex queries and can handle the size of the datasets. The production database OpenLink Virtuoso was chosen because of its performance and because of issues that Apache Jena had with invalid URL characters that were in both datasets. Virtuoso provided a decent performance on 71 GiB of data with having 8GB of RAM (which is mostly the bottleneck of these databases).

3.4.4 Data Retrieved

Various semantic data were obtained from DBpedia and Wikidata. These knowledge bases index their pages based on a URI which is a unique identifier of the entity in a similar format as URL address. For an entity, its URI, related images, pages and titles were kept. Images were identified using their URL addresses as well as pages. For pages, also a language of that page was kept, where applicable (for example Wikipedia articles have that information). As with titles, Wikipedia links between language mutations were exploited to retrieve titles in multiple languages. The title and its language was kept. Next, a short abstract in English corresponding to the first Wikipedia paragraph was kept. Finally, the categories of entities together with their hierarchy were retrieved. The specific queries used for retrieval of different sets of data are in Appendix C.

On top of these data, the linking between DBpedia and Wikidata was kept for quick querying. Together with these data, links to the original entities were stored, so that additional data retrieval is possible. This also allows merging the ontology built in this step with DBpedia and Wikidata through the Linked Data standard allowing anybody to query this ontology together with the other two data endpoints. This would create a computer vision contribution to the Semantic Web of Open Data.

3.4.5 Technical Problems

The main issues that had to be addressed were mistakes in both DBpedia and Wikidata dumps and the size of the data in a combination with limited computational resources. DBpedia had mistakes in its mappings, so for example the Statue of Liberty had no GPS coordinates associated with it. This was solved by adding another source of data - Wikidata. Wikidata dumps contained much more serious mistakes such as duplicated records and incomplete dumps. This must had been and was manually analyzed and fixed before the data could have been processed. The size of the data was partly solved by a rapid pre-processing, partly by choosing the appropriate database for each task but mostly by a careful analysis of the specific task, including testing with a small sample, and then adjusting both data, used software and implementation to provide maximum performance on available hardware [16] for each use-case.

3.5 Results of Wikipedia and Wikidata Processing

From DBpedia and Wikipedia together, *357 thousand distinct landmark entities* described by *1.1 million distinct labels* in *390 languages* [17] were retrieved while processing the total amount of 391 GiB of data containing 2.9 billions of triplets. For every landmark, its category was kept and the hierarchy among the total of *2904 categories* was extracted from DBpedia and Wikidata.

[16] A 5-year old computer with a 4-core i7 processor, 8GB RAM and 96GiB SSD

[17] The exhaustive list of processed languages together with their frequencies and assets is provided in Appendix B

In Table 3.5, different data source together with a manual approach of raw Wikipedia parsing [18] are compared. It can be seen that deduplication was an essential part of the processing pipeline as duplicates constitute 39% of landmarks and 61% of their labels. In Figure 3.5, the asset of each of the 100 most frequent languages can be seen.

The estimation of false positive and false negative rate of landmark identification in DBpedia and Wikidata is provided in Section 3.3.5.

Data source	Number of landmarks	Number of labels
Manual infobox analysis	$60 * 10^3$	$60 * 10^3$
DBpedia	$208 * 10^3$	$1 * 10^6$
Wikidata	$328 * 10^3$	$1.7 * 10^6$
DBpedia and Wikidata merged	$536 * 10^3$	$2.8 * 10^6$
Both merged an deduplicated	$\mathbf{357 * 10^3}$	$\mathbf{1.1 * 10^6}$

Table 3.5: Number of landmarks and labels for different datasets

Same labels appear in multiple images but are counted only once as a result of the deduplication. This causes that the number of all deduplicated labels is lower that the number of labels from Wikidata alone.

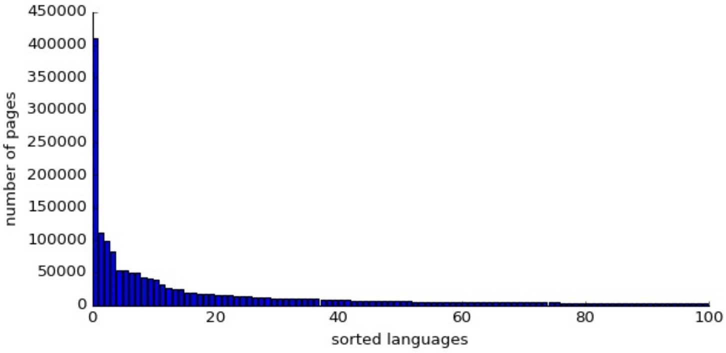


Figure 3.5: The number of Wiki Landmark pages in top 100 most frequent languages

[18] The manual approach of raw Wikipedia parsing was based on a simple keyword matching in infobox template names where the landmark keywords were manually defined.

Chapter 4

Wiki Core Images

Each Wikipedia article consists of a text and a media component. As Wikipedia aims to be primarily an encyclopedia, in 2004, Wikimedia Foundation launched a project providing a central repository for media content named Wikimedia Commons. All media files in this project have a free-use license and the project groups them into categories, comparable to Wikipedia articles. These categories present everything that people associate with a given topic. When considering images only, this is different from a content-based image categorization, as in Wikimedia Commons category, images of different objects sharing only the topic of the category appear. This fits in with the approach of breaking articles into described entities consisting of Wikipedia articles and Wikidata documents perfectly.

4.1 Semantic Interpretation of the Image Collection

From a semantic point of view, the collection of images related to a landmark do not describe the main object itself, but rather a semantic concept related to the landmark. This semantic concept includes everything people think that is closely related to the landmark such as signs with the landmark name, various details of the main object including inside views, or maps showing the landmark location.

From the computer vision point of view, multiple objects are being visually described, possibly with no visual link between them. An example is an inside and outside view where there cannot exist an image displaying both. These objects represent the same semantic concept. This concept was described textually on Wikipedia and Wikidata and it is described visually on Wikimedia Commons. And because people use an overlapping set of keywords to annotate images related to a concept, search engines show also images of a semantic concept, possibly also multiple concepts, rather than of one object. With regard to this, the visual description of a semantic concept is suitable for this work.

A consequence of working with semantic concepts is the fuzzy nature of the border of the concept. Specific objects have clear borders, but they can be part of different semantic concepts. An example could be paintings - it is a semantic concept itself, a piece of art, but it can be also part of the semantic concept of the museum it is in. Even though the painting itself is not a landmark and therefore should not appear in Wiki Landmarks, the museum is a landmark and when visually describing the interior of it, the painting represent a view of that landmark. The image of the painting is therefore part of the Wiki core imageset.

Only about 50% of images from Wiki core imageset (a subset of Wikimedia Commons) depict the main object of the landmark. This means that, if restricted to images of the object only, only half of the total nuber of relevant images could be utilized in the best-case scenario. Details can be found in the results section 4.5.1.

4.2 Wiki Image Sources

In a Wikipedia article, images related to it are included directly in the article. This presents the first source of images, so one image was retrieved for each Wikipedia article, if it had any. For this purpose, the image from the infobox is taken because Wikipedia articles are divided into sections which usually form a different topic from the semantic point of view (e.g. describing a person who built the castle). The image in the infobox was chosen by the community as most representative for the article and therefore it can be considered the most representative one. The DBpedia project was used as a method for working with already parsed Wikipedia data. A Wikidata document also contains an image directly in the document, but often more than one. In this case, the first image enlisted was taken as the most representative one and the others kept as related.

The next step is to retrieve all images related to a Wikidata document from Wikimedia Commons, so that a semantic concept defining a landmark consists of structured information together with visual data creating a complex description of that concept. This is done by linking together Wikidata documents and Wikimedia Commons categories and parsing all images from the categories. This was performed for every Wikidata document. For the linking, data directly from Wikidata were used and to retrieve the images from Wikimedia Commons, its website was queried, first for the category page containing the list of images in that category and second for the image detail pages. This was also a way of assigning an image to Wikipedia articles and Wikidata documents that did not contain any image. Wikipedia articles were not directly linked with Wikimedia Commons categories.

4.3 Image Metadata

When indexing an image on the Internet, text data are indexed together with the image itself. In this case, the text data are restricted to the "alt" attribute from the "image" HTML tag and to the "caption" text of the image which are both added to all images retrieved. The reason for this is to enable better matching between images from Wikimedia Commons and images retrieved using services like Flickr as the text information plays an essential role for the image categorization on their side.

The "alt" attribute of an "image" HTML tag contains the alternative text that should be used when the image cannot be processed. This is used extensively in web crawlers indexing these images and screen readers on the user side. The "caption" text is taken from Wikimedia Commons detail page directly and is also useful as it provides a condensed summary of that image provided by the person who uploaded the image.

4.4 Retrieval Implementation

For the image and metadata retrieval, an online Wikimedia Commons API was used. Also, image URL addresses were uncovered using the API. The reason for using an online API instead of a dump was that the volume of data downloaded (913 thousand images) was not significant compared to the total number of media files stored there (35 million files). Also, the download task is easily splittable and distributable which predetermines it to use the online API. Before the download started, Wikimedia API etiquette was studied, so that the download task complies with it. The download saturated the 100Mbps bandwidth, so from the software architecture point of view, this approach was better than downloading the whole Wikimedia Commons dump and then parsing it which would take about 38 times longer.

4.4.1 Data Retrieved from Wikimedia Commons

Together with the images themselves, metadata of these images were obtained. This includes the alt field of the HTML image tag, the image caption and the filename provided by the HTTP server. These three fields provide a textual description of the image. From the image itself, the following fields were extracted - width and height in pixels, size in bytes and mime type of the file. Also, the full response of the server together with the image EXIF information were kept for future processing. This allowed to repeat all downloads with a response signaling that for example the server is unavailable at the moment or there was some network error. This was performed multiple times to overcome errors caused by the network.

4.5 Results of Wiki Image Retrieval

From all sources, *1 million images* were retrieved totaling *2.6TiB* of data. From Wikimedia Commons alone, *913 thousand images* were downloaded for 45 thousand Wiki landmarks. Furthermore, metadata (namely 1.1 million image annotations) were added to 1.1 million images from all three Wiki projects. After enriching with images from Wikimedia Commons, there are 212 thousand landmarks on Wikipedia and Wikidata having at least one image. This forms the Wiki core imageset, a subset of Wikimedia Commons restricted to Wiki Landmarks.

Table 4.1 shows the number of images retrieved from different data sources. Even after all the linking, some Wikipedia articles and Wikidata documents could not have been assigned an image. Most of these articles are so called stub articles, containing only a very basic information about the topic [1]. The ratio of entities without any image from different sources is displayed in Table 4.2. The histogram of the number of images per article can be found in Figure 4.1 To have a better picture of relations between data from different sources, Table 4.3 shows counts for entities when using only some data sources.

The last table, Table 4.4, shows file types that were accepted and processed as images. The vnd.djvu and x-xcf image formats as well as images embedded in pdf files were excluded from processing.

[1] One example of a stub article is https://en.wikipedia.org/w/index.php?title=Nielsen_Airport&oldid=552692691

Data source	Number of landmarks	Number of images
DBpedia	$208 * 10^3$	$141 * 10^3$
Wikidata	$328 * 10^3$	$139 * 10^3$
Wikimedia Commons	$45 * 10^3$	$913 * 10^3$
All data sources merged	$518 * 10^3$	$1.1 * 10^6$
All sources merged an deduplicated	$\mathbf{357 * 10^3}$	$\mathbf{1.1 * 10^6}$

Table 4.1: Number of landmarks and images for different datasets

Data source	Without an image	With an image	Text-only entities ratio
DBpedia	$141 * 10^3$	$67 * 10^3$	32%
Wikidata	$132 * 10^3$	$196 * 10^3$	58%
All	$\mathbf{212 * 10^3}$	$\mathbf{145 * 10^3}$	**41%**

Table 4.2: Number of entities without any image for different datasets

The "All" data source denotes DBpedia, Wikidata and Wikimedia Commons all merged and interlinked.

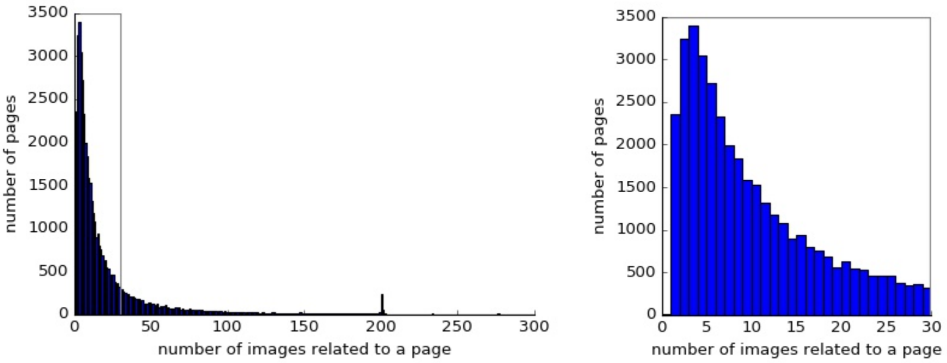


Figure 4.1: A histogram of the number of images per Wiki page up to 300 images.

The most common number of images per Wiki page is 3. There are 14 pages with more than 300 images. The number of 3500 pages corresponds to 7.8%. The explanation for the peak at 200 was not found. The author's opinion is that there was a limit for the number of images in a category that was later canceled. The specific values at the peak are: (199: 26), (200: 234), (201: 52).

Data source combination	Number of entities
dbpedia	28,516
wikidata	124,687
dbpedia-wikidata	158 648
wikidata-wikimedia	24 262
dbpedia-wikidata-wikimedia	20,652

Table 4.3: Number of entities for different combinations of data sources

There is no combination dbpedia-wikimedia because Wikimedia Commons was linked with Wikidata only.

Filetype	Number of landmark images
jpeg	$1 * 10^6$
png	$13 * 10^3$
tiff	$10 * 10^3$
svg	$3 * 10^3$
gif	$1 * 10^3$

Table 4.4: Number of landmark images for filetypes that appear on Wikimedia Commons

The following mime types were excluded from processing: application/pdf, application/x-empty, application/xml, audio/ogg, image/vnd.djvu, image/x-xcf, text/html, text/plain, video/ogg, video/webm.

4.5.1 Number of Images of the Main Object

It is a subjective task to estimate how many real landmarks are there in the set of Wiki Landmarks and it completely depends on the used landmark definition. False positive and false negative rates of Wiki Landmark identification are estimated in Section 3.5. To measure the Wiki core imageset diversity, the number of images depicting the main object was estimated among all images. For this task, 100 random landmark images from Wikimedia Commons were chosen. Landmarks depicted by the images were studied in order to identify the main object of the landmark. Then, the image was accepted when it depicted the main object. The set of accepted images are representative images of the landmark object and would be a valid depiction for a guide book. Among 100 random images, 51 were depicting the main object of the landmark. This means that for about 50% of images the image retrieval task is well defined.

Images were rejected based on two reasons - either it was an incorrect view or an inappropriate landmark object. The incorrect view could be of a specific detail of the landmark, for example a statue or one of its rooms, or of another object related to the landmark. The inappropriate landmark objects were of two types, objects not unique, such as an underground stations, and objects too spacious so that they cannot be depicted by a single image, such as parks, cemeteries or indoor museums.

Chapter 5

External Images of Wiki Landmarks

Landmarks got annotated by labels, GPS coordinates, text and images. Using these data, independent datasets were queried to provide a set of relevant images for each landmark. These images are in some way related, e.g. by a text label or GPS coordinates, but are not necessarily images of the same semantic concept which makes them irrelevant. It is caused either by an ambiguous query or by incorrect categorization on the database side. Still, these images provide an additional visual description of the object when used as a complement of the Wiki images.

5.1 Querying Publicly Available Sites

For the queries, five distinct services with a publicly available API were used. For the retrieval, the same approach as with searching images manually was chosen, entering the query to the search field on the main page and parsing the results displayed. Only the first page of results was parsed which was never more than 100 images. The following five services were queried.

- **Google** Images - *the most popular search engine worldwide*
- **Flickr** - *a popular personal image hosting service*
- **Yahoo** Image Search - *3rd most popular search engine worldwide*
- **Bing** Images - *2nd most popular search engine worldwide*
- **Yandex** Image Search - *the most popular search engine in Russia*

For all these services, text queries were performed. The input for the query was a title either of the Wikipedia article or the Wikidata document. For the title, used as the input, all languages were employed to retrieve as much relevant results as possible. Because the titles can be identical in many languages, a deduplication independent of used languages was performed. In case of Flickr, also spatial queries were performed using GPS coordinates associated with an article or document.

These services were chosen to provide a diverse and representative set of available search engines. Also, Baidu (the most popular search engine in China) and Pixabay (free-to-use license images) services were tested on a small sample. Pixabay did not pass because of a very few results for each queue and Baidu because it does not provide links of the images in the Internet but only links to its own servers. Pixabay suffers from the same issue too.

5.2 Metadata from Online Databases

The images are indexed by the databases together with metadata describing those images. These metadata could be obtained from the result list of each online database. The obtained metadata were as follows.

- Label - the input of a query
- Image URL - the download path of the image
- File name - a file name of the image file
- Size - an image size in bytes
- Width - an image width in pixels
- Height - an image height in pixels
- Page - the URL of the page the image is on
- Alt - the "alt" attribute of the "image" HTML tag

In case of spatial queries, GPS coordinates are stored instead of a label as the input of a query. Furthermore, response headers are stored together with the compressed raw response body for possible further analysis or re-parsing. The reason for this is that each query is very expensive in terms of time and minimizing the number of queries, both by deduplication of query inputs and storing query outputs is necessary.

5.3 Querying Implementation

For the purpose of a relevant images retrieval, five distinct services providing a publicly available API were used. The same API use all users searching images using these services. The query was put into the search field on the main page and results were processed from the resulting page. The results were parsed and data stored in the database. There were 65 results per query in average yielding a 299 million results in total, so data could not be saved in the db one after another while interlinking them. First of all, image URL addresses, sizes and related alt texts were extracted during one pass through all the data. These were then deduplicated and side-loaded into the database using aggregated INSERT queries inserting 10,000 rows at a time. Next, the records linking these newly inserted data with existing data in the database were created using a specific algorithm utilizing the possibility of SQL to return data as part of the INSERT statement. This way, many SELECT queries could be combined with INSERT queries reducing the total number of queries and allowing to bundle them to batches of 100 queries with an application-side processing and joining. This took the load of the database and allowed to process all 299 million results in a reasonable time of two weeks. Despite this effort, the database was the weak link in this step because of the too-strict formal model of the data in it. This would be an issue with any SQL database, so a different approach should be used in future instead.

5.3.1 Data Retrieved from Online Databases

From every online database, a set of images with their metadata together with the ordering of results was kept. The metadata fields that were retrieved are described in Section 5.2. The metadata retrieval replaced a client side processing for some use cases. Also, a wide range of errors was processed, so that the clients can flexibly react to different situations. To split query retrieval and query processing and also to allow later re-parsing of all responses, the full responses were saved. The response header was kept in the database to allow filtering according to header data. This allowed to re-schedule all queries where for example a network communication problem occurred.

5.4 Query Results Quality

The high number of independent online services is justifiable because the result set quality varies a lot. As an example, two queries were chosen. The first one is "Waldau" and the second one is "Cleveland Burke Lakefront Airport". The "Waldau" query is interesting because it was build based on a Wikidata document only [1] as it does not have a Wikipedia article at all. It is a name of a psychiatric hospital in Bern but the name "Waldau" is very ambiguous - it is not only a name for different places but also a name of an actor. Because of that, results of all queries are flooded with irrelevant images. In this case, only Flickr spatial query gave in some way relevant images. The results of this query can be found in Appendix D.

The full results for the query "Cleveland Burke Lakefront Airport" are provided in Appendix D. Interesting parts of the results are presented also in this section. The ground truth images can be seen in Figure 5.1. In this case, as can be seen in Figure 5.2, Flickr returned images of planes both being spatially and textually related to the airport object but none being relevant to the airport landmark. This was a common case with many Flick queries. Yahoo (Figure 5.3), on the other side, gave a mixture of relevant and irrelevant results for the queries in different languages. Two implementation issues and one semantic arises here.

The first implementation issue is that the first image given by Yahoo could not be retrieved after approximately a month. There were a lot of reasons why an image could not be retrieved. The two most common were that the image stopped existing and that there were some bot checking algorithm which Yahoo bot passed and the used bot did not. Second implementation issue can be seen on the third line of results of Yahoo query where two identical images appear. They have a different size and both are in a jpeg format which is a lossy compressor. Even though originally identical, that makes to differ a lot even when comparing pixel by pixel.

[1] Wikidata document for "Waldau" https://www.wikidata.org/wiki/Q2541082

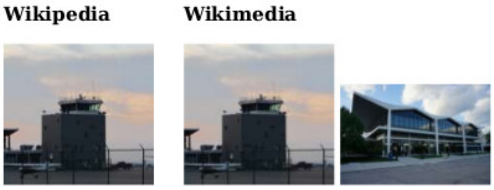


Figure 5.1: Wikipedia (left image) and Wikimedia (two right images) ground truth images for the Wikipedia article "Cleveland Burke Lakefront Airport"

The Wikipedia image is contained also in the Wikimedia image set because all files from Wikipedia are hosted on Wikimedia

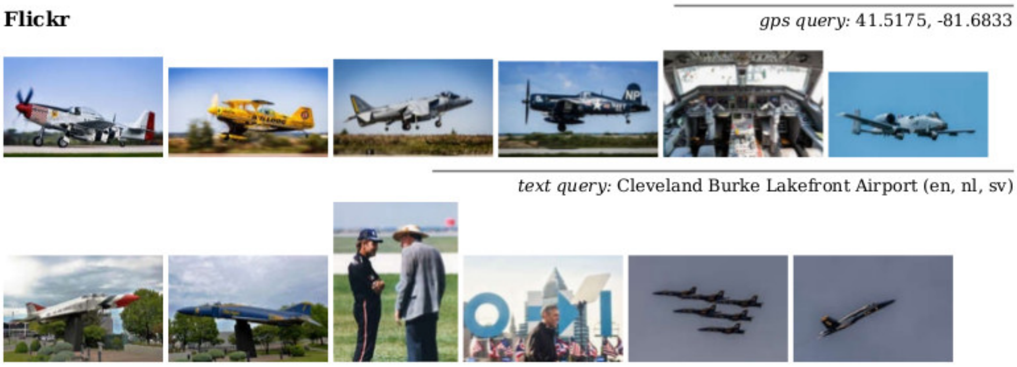

Flickr
gps query: 41.5175, -81.6833
text query: Cleveland Burke Lakefront Airport (en, nl, sv)

Figure 5.2: Top six Flickr results for a GPS query (top row) and text query (bottom row), both relevant to the Wikipedia article "Cleveland Burke Lakefront Airport"

The text query was performed only once even though it corresponds to the article title in three languages (English, Dutch and Swedish)

5.4.1 Relevance Undecidability

The semantic point of view is not always easily decidable, even for humans. The last row of Yahoo results shows query results for a Japanese text query, displaying four Formula racing results. It is not clear whether these are related to the airport or whether there was an ambiguity in the Japanese text query which meant not only an airport but also was somehow related to the Formula racing. The truth is there existed a Gran Prix of Cleveland [2] which was held on the Cleveland Burke Lakefront Airport annually. This information makes all the first sight irrelevant results in some way relevant to the airport. This shows the importance of working with the semantics of the objects, not only objects themselves.

[2] https://en.wikipedia.org/wiki/Grand_Prix_of_Cleveland

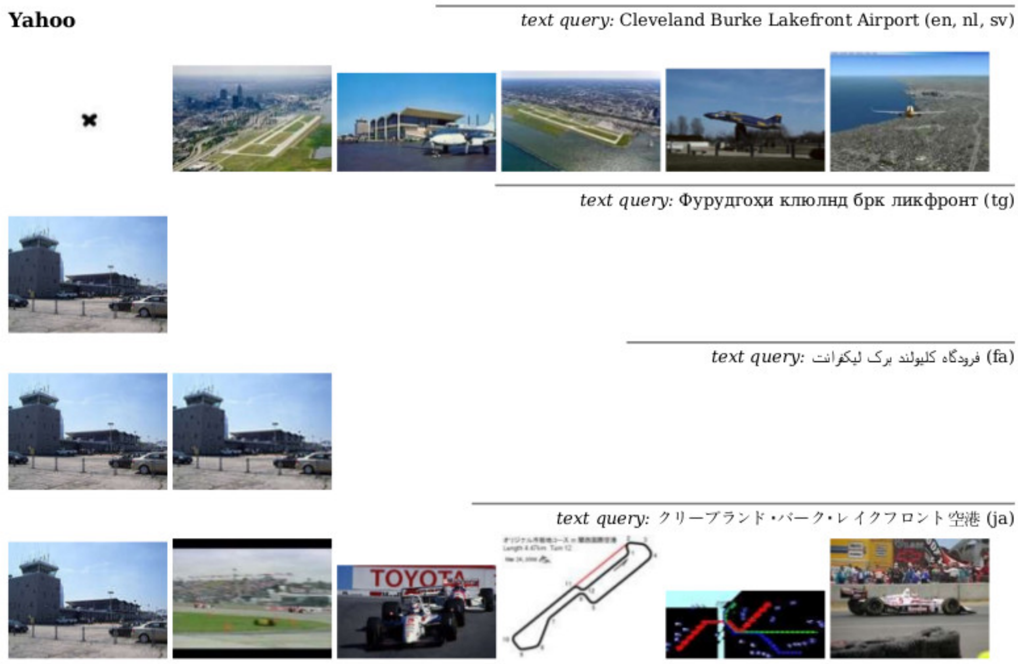

Yahoo
text query: Cleveland Burke Lakefront Airport (en, nl, sv)
✖
text query: Фурудгоҳи клюлнд брк ликфронт (tg)
text query: فرودگاه کلیولند برک لیکفرانت (fa)
text query: クリーブランド・バーク・レイクフロント空港 (ja)
TOYOTA

Figure 5.3: Top six Yahoo results for a text query in English, Dutch and Swedish (first row), in Tajik (second row), in Persian (third row) and in Japanese (last row), all relevant to the Wikipedia article "Cleveland Burke Lakefront Airport"

The first image link has been removed which is denoted by a black cross on a white background. In the third row, the same image is given in two different sizes both times in a JPEG format which is a lossy compressor causing the images to be indeed different, even when comparing them pixel-wise.

5.5 Results of External Image Keyword-based Retrieval

Data about the total number of *131 million distinct images* on *100 million distinct pages* annotated by *107 million distinct labels* in 390 languages were retrieved. This was achieved by executing 5.7 million queries from which 5.3 million queries were based on the title of the article in multiple languages and 372 thousand were spatial queries based on the GPS coordinates of a landmark. The result was five sorted sets of images with their metadata, all relevant to the label. Based on the metadata retrieved from the online databases, to download all images, 28TiB of disk space would be necessary.

The average number of results provided by each image database for one query together with the total number of results retrieved from that database is shown in Table 5.1. To illustrate the image quality difference, two histograms comparing the image dimensions for Wikimedia Commons and the rest of images on the Internet indexed by the image databases is presented in Figure 5.4. It is interesting to see how much more superior in quality are Wikimedia Commons images when compared to the Internet average.

The quality of results obtained from the image databases is illustrated in Figure 5.5. The difference between result quality between English and other languages is provided in

Image Database	Results demanded	Average results provided	Total results
Google Images	100	94.7	$100 * 10^6$
Flickr	100	55.3	$79 * 10^6$
Yahoo Image Search	60	41.6	$59 * 10^6$
Bing Images	35	28.6	$30 * 10^6$
Yandex Image Search	30	28.9	$31 * 10^6$
All aggregated	325	**249**	$\mathbf{299 * 10^6}$

Table 5.1: Number of results retrieved from the five online image databases.

In case of Flickr, not only text queries but also spatial queries were performed.

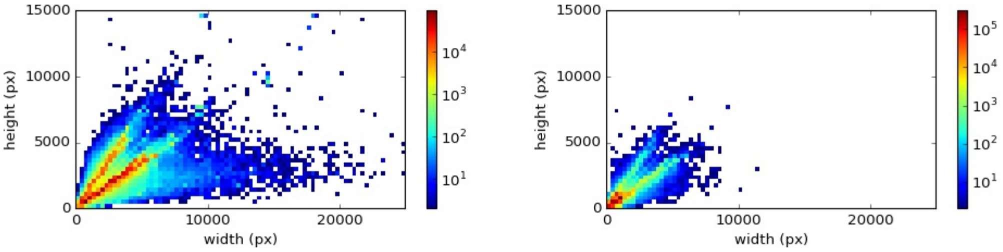


Figure 5.4: Image dimensions for the images from Wikimedia Commons (left) and the average on the Internet (right)

For the histogram of the average on the Internet, a random sample with the same size of 1.1 million was chosen. The histogram was limited to 25000px in width and 15000px in height, 170 images from Wikimedia Commons exceed one of these dimensions

Figure 5.6 [3]. Provided results were obtained by manually annotating 483 random queries. Participants were asked to place an image into one of five categories - duplicity of the core image, displaying the same object as the core image, related to the query, unrelated to the query and missing image. The plots display number of at least related images.

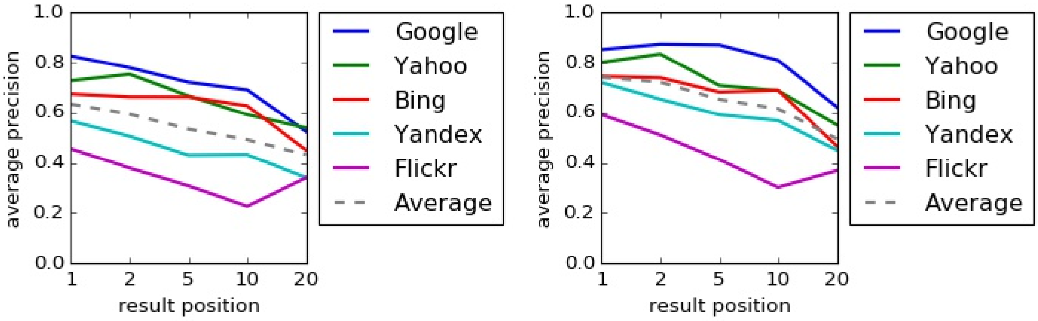


Figure 5.5: The results relevance for the five online image databases for all languages (left) and English only (right)

Images at "result positions" 1, 2, 5, 10 and 20 were manually annotated. The "average precision" is what percentage of images is relevant.

[3] For the purpose of this plot, the number of samples was the same for English and other languages

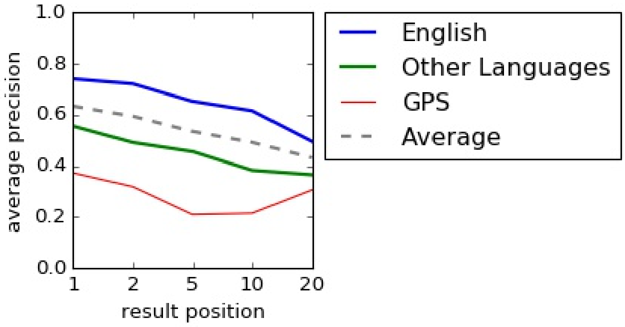


Figure 5.6: Results relevance for queries in English, non-English languages and GPS queries

Images at "result positions" 1, 2, 5, 10 and 20 were manually annotated. The "average precision" is what percentage of images is relevant.

Chapter 6

Canonical View Identification

For the purpose of a canonical view identification, the images are visually described and then clustered according to their description. For an image, a fix-length vector, denoted as visual descriptor, is computed first. This descriptor is based on the visual information contained in the image. The set of descriptors form together a descriptor space which has the property that visually similar images should be consistently placed close to each other. The next steps are image clustering in the descriptor space which groups visually similar images and canonical view identification which finds the most representative image in each group. The most representative image of a cluster corresponds to the center of the cluster.

Two imagesets were obtained as part of this work - the *Wiki core imageset* and the *external imageset*. The core imageset contains images manually linked to Wiki pages, the external imageset contains images related to Wiki pages through a common text label. Representing a Wiki page using a text label introduces ambiguity, described in detail in Section 5.4. This prevents from clustering the external imageset without further image validation. On the other side, the core imageset is already manually validated. For the purpose of canonical view extraction, only the core imageset is clustered. Every image from the core imageset corresponds to a specific Wiki Landmark which is exploited in the clustering - only images of the same landmark are clustered together. This partitions the descriptor space allowing for a very fast clustering and enhancing the precision of the clustering algorithms as the biggest number of outliers is not included in the clustering.

6.1 Visual Descriptor

The visual descriptor was computed by a deep convolutional neural network VGG [12] which was specifically fine-tuned for landmarks [9]. Simply speaking, the network computes for every image a 3D tensor where the first two dimensions correspond to the image dimensions and the third represents the set of activations for 512 feature maps. The descriptor is computed as a maximum activation on the image for each feature map. This yields a fixed-size vector of length 512 which is called Maximum Activations of Convolutions (MAC) vector.

The VGG network is a very deep convolutional network specifically designed for large-scale image classification. The fine-tuning of the network consisted in using automatically chosen image pairs among unannotated datasets for training the network. The image pairs were chosen through a geometric validation based on the Bag-of-words model. Before the image descriptor was computed, all images were downscaled to have 1024px on the longer edge while keeping their aspect ratio.

The biggest advantage of this visual description method is the speed of computation and memory requirements which are superior to the traditional bag-of-words method combined with the SIFT descriptor with a minimal loss in performance [1]. Mapping every image to the descriptor space of dimension 512 is advantageous - it simplifies the image retrieval task (nearest neighbor), query expansion (average of descriptors) and clustering of images based on their visual information (clustering in the space of descriptors). The biggest disadvantage of this descriptor is that it was not trained as scale-invariant which turned out to be a big downside in the image retrieval as the images on the Internet differ in dimensions greatly.

6.2 Clustering Method

Identification of canonical views was performed through clustering of visual description of images. Then, cluster centers correspond to canonical views. For the clustering, an algorithm with a varying number of clusters must have been used because the number of canonical views is not known beforehand. Furthermore, centroid-based clustering algorithms such as mean shift did not perform well for this use case. For image sets obtained from photo sharing services such as Flickr, it is a valid assumption that people use only a small number of most-popular viewpoints to capture each landmark, but for the image set from Wikimedia Commons, it is not a correct expectation. Through the community-driven process of selection, intentionally diverse images are presented for each landmark.

A density-based clustering algorithm DBSCAN was used to cluster the image descriptor space. Compared to the centroid-based clustering algorithms, it does not expect the clusters to be convex-shaped with a single cluster center. It utilizes the fact that areas of high density are separated by areas of low density. This enables the cluster to follow the chain of images as the camera rotates around the object, moves towards the object or follows some other trajectory, for example a hiking trail near the object. In the DBSCAN algorithm, if a point is close enough to any point of the cluster, it is part of the cluster. This iterates until there are no points mutually close enough and not incident to the same cluster. [2]

DBSCAN was compared to mean shift on a set of 50 random landmarks, the results were manually evaluated and it performed equally or better than mean shift on every sample [3]. In the case of shifting view, it outperformed mean shift substantially. This phenomena is illustrated in Figure 6.1 where the mean shift cluster centers are provided for a sample query [4]. In case of DBSCAN, all the images are from a single cluster with the most representative image being displayed in Figure 6.2. In case of mean shift, the cluster centers are positions

[1] Source: [9] section 6.

[2] From the image retrieval point of view, it is a continuous query expansion with one initial query image having a number of result images closer than a specified threshold serving as query images themselves.

[3] For the comparison, multiple parameters were evaluated for both algorithms and the best performing were chosen.

[4] In this case, the phenomena is further amplified by the fact that the visual descriptor used is not scale-invariant.

on a hiking trail where multiple images were taken. These are more or less random points on the trail, not canonical views of the object from the computer vision perspective. Setting the mean shift bandwidth parameter higher partially solves this issue because it merges some of the clusters, but it merges other unrelated clusters too. It is caused by the fact that the presented images in Figure 6.1 are indeed distant in the descriptor space.

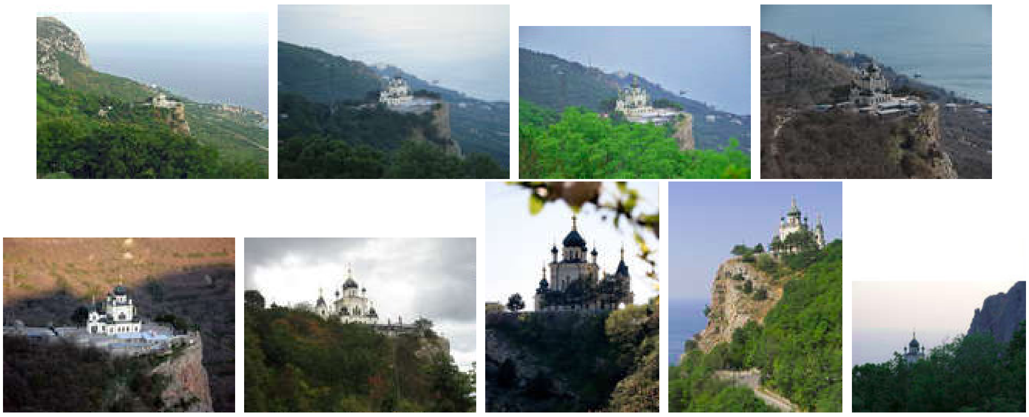


Figure 6.1: Illustration of mean shift cluster centers of images taken from a hiking trail around the Church of Resurrection of Christ in Foros

Cluster centers are sorted, so that the hiking trail the images are taken from is apparent. 26 images were clustered into 9 clusters. In DBSCAN, this all forms a single cluster with the center displayed in Figure 6.2.

Figure 6.2: A single cluster center as a DBSCAN result corresponding to the same set of images as Figure 6.1

6.2.1 DBSCAN Algorithm Augmentation

The DBSCAN algorithm distinguishes two types of cluster membership - core and non-core. The core samples can expand the cluster themselves, the non-core are assigned to the cluster at the end of the algorithm and therefore cannot contribute to the cluster expansion. This leads to two parameters for the DBSCAN algorithm - the "radius" and the "minimum number of neighbor points". The "radius" defines when points are close enough, and the "minimum number of neighbor points" specifies the number of points that must be close enough to a point to be a core sample. Because the average number of images forming a cluster is relatively small for this image set, the distinction of core and non-core samples was suppressed by setting the "minimum number of neighbor points" to 2 with the consequence that all points are core samples. This allowed to reach distant views with only one connecting image which is frequently the case in the Wikimedia Commons image set.

To emulate the non-core sample behavior, at the end of clustering, clusters were expanded - not classified points were assigned to the closest cluster when their distance to the nearest point of the cluster was under a specified threshold. This cluster expansion was performed for the mean shift algorithm too in order to compensate the sensitivity to the bandwidth parameter and to enhance its results.

Because a cluster in DBSCAN can have any shape, it is not a straight-forward task to decide where the center of the cluster lies. For the purpose of taking the most representative image, the cluster equilibrium was computed by averaging all members of the cluster and then the image from the cluster closest to the equilibrium was taken as the representative one.

This algorithm presents only one of the possibilities of getting canonical views from clusters. A better approach would be to measure the standard deviation of all points of a cluster and when exceeding a threshold, multiple canonical views would be obtained from a single cluster. This would ensure that even landmarks having so many images that the uninterrupted chain of images is covering a substantial area around the landmark would have multiple canonical views. Multiple diverse views from a single DBSCAN cluster can be obtained by simply taking N images which differ most - are most distant in the descriptor space. Landmarks having that many images in an uninterrupted chain were a corner case for this image set, as shown in Figure 6.3 in the Results section, so this approach is not used to get canonical views.

6.3 Results of Image Clustering

The number of 1.2 million core images related to 208 thousand Wiki landmarks was clustered. From these images, 86% is part of a cluster with at least one another image and clusters have 6.8 image on average. The cluster sizes are described in more detail by Figure 6.3. An example of canonical views of a clustered landmark - the Kirkstall Abbey - is provided in Figure 6.4. The corresponding outliers (clusters with only one image) are provided in Figure 6.5. Another example of canonical views is provided in Figure 6.6.

A DBSCAN clustering result of the Kirkstall Abbey is provided in Figure 6.4. When the same set of images was clustered using the mean shift algorithm, the last two clusters were merged with the first one yielding 4 clusters as opposed to 6. At the same time, only 23 images were clustered as opposed to 38 in case of DBSCAN. Increasing the bandwidth parameter leads to more clusters merged and decreasing it leads to less images clustered. Even though clustering evaluation is generally hard, subjectively all clusters from the figure provide a different view of the object and merging any three clusters is not justifiable. As there were no false positive samples in the clustering result of DBSCAN, decreasing the number of clustered images is also a loss in performance.

Only the core imageset was clustered because of limited resources and the necessity to validate the external images which would eventually require more resources. The external imageset was considered and a random sample was chosen and processed. The results of clustering the external imageset together with the internal one are provided in Figure 6.7

and Figure 6.8. In the first figure, possibly any of the images can be correct, but probably not all of them. In the second figure, all images are incorrect except for the one from Wikipedia.

The fact that the image descriptor is not scale-invariant caused that images with less than about 400px on the longer edge were matched incorrectly and must have been filtered out. The exact same images in two different sizes were more distant in the descriptor space than pairs of unrelated images. When kept, small images introduced an error to canonical views by creating redundant clusters. This was most apparent when the external imageset containing images from the whole Internet was used.

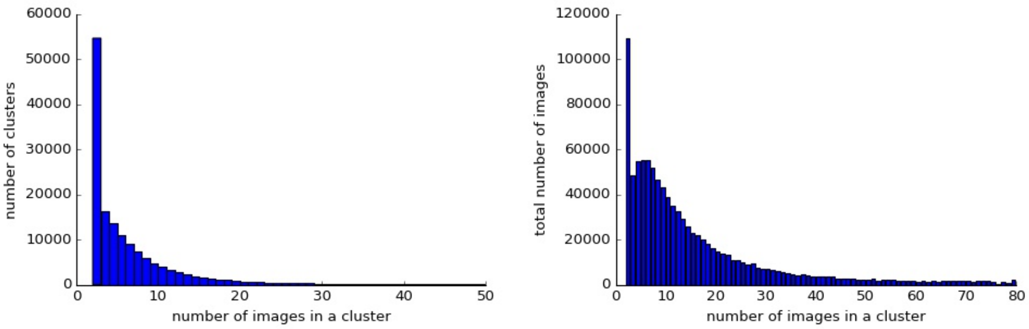


Figure 6.3: A histogram of cluster sizes (left) and an image distribution among cluster sizes (right)

The average number of images per cluster is 6.8 (excluding single-image clusters).

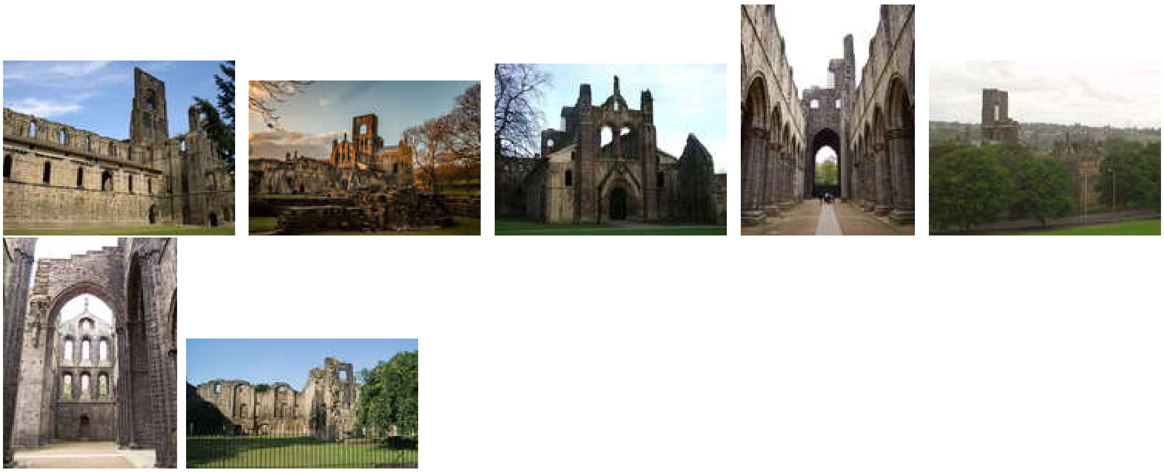


Figure 6.4: Illustration of all canonical views of the Kirkstall Abbey

These 7 cluster centers represent 38 similar images. When clustered using mean shift, the last two clusters were merged with the first one and at the same time clustering only 23 images.

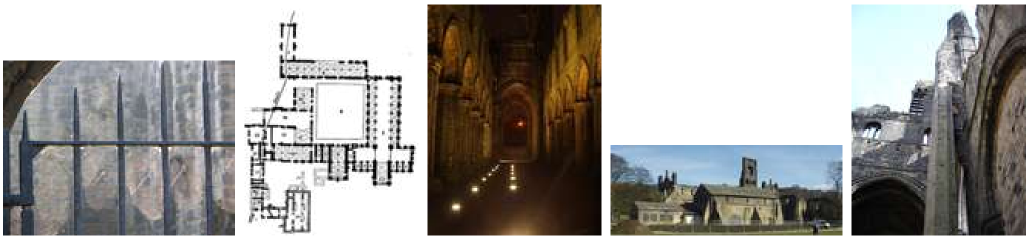


Figure 6.5: Illustration of DBSCAN outliers of the Kirkstall Abbey

These 5 outliers are a subset of the total 34 outliers. Outliers are considered clusters with only one image.

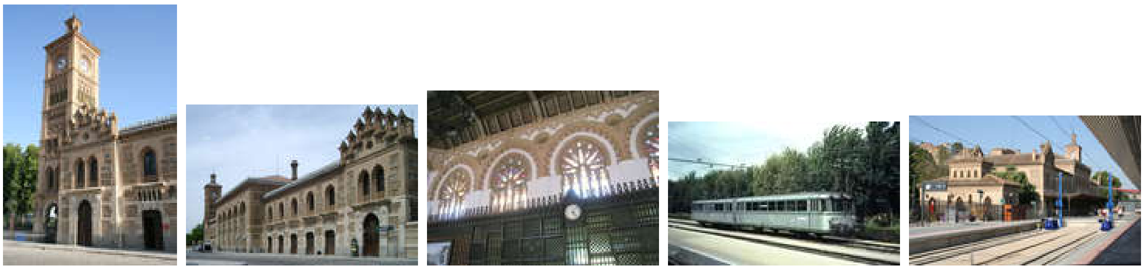


Figure 6.6: Illustration of all canonical views of the Toledo railway station

These 5 cluster centers represent 16 similar images.

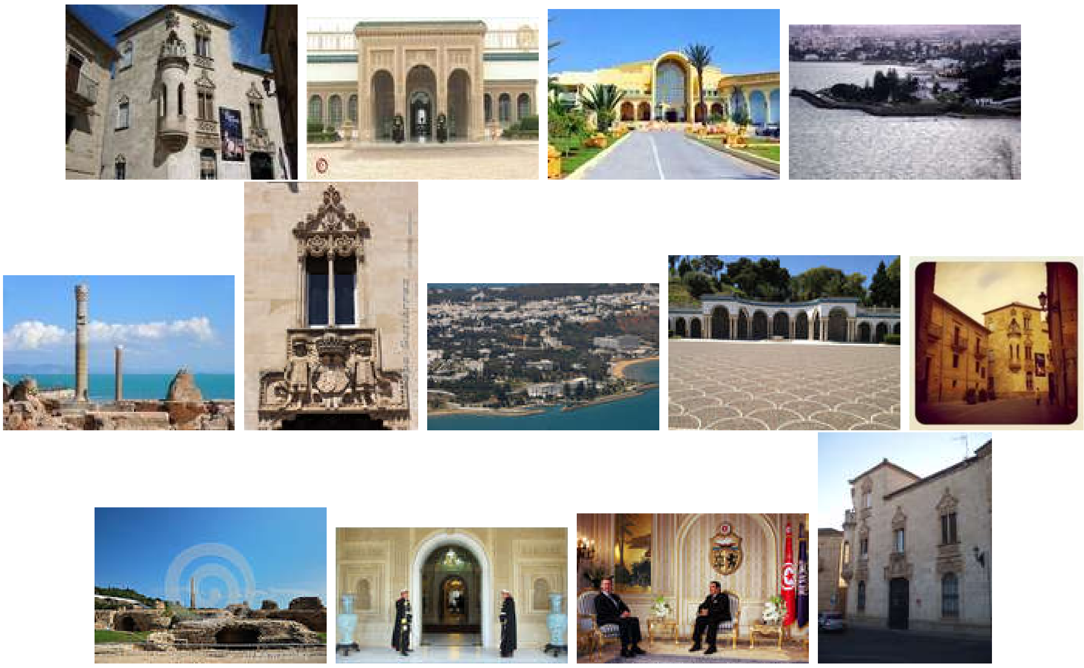


Figure 6.7: Illustration of canonical views of the Presidential palace of Carthage when using also the external imageset

Only clusters 4, 7 and 12 contain an image from the internal imageset. Any image provided here is potentially an image of the palace. Only clusters with more than two images are shown.

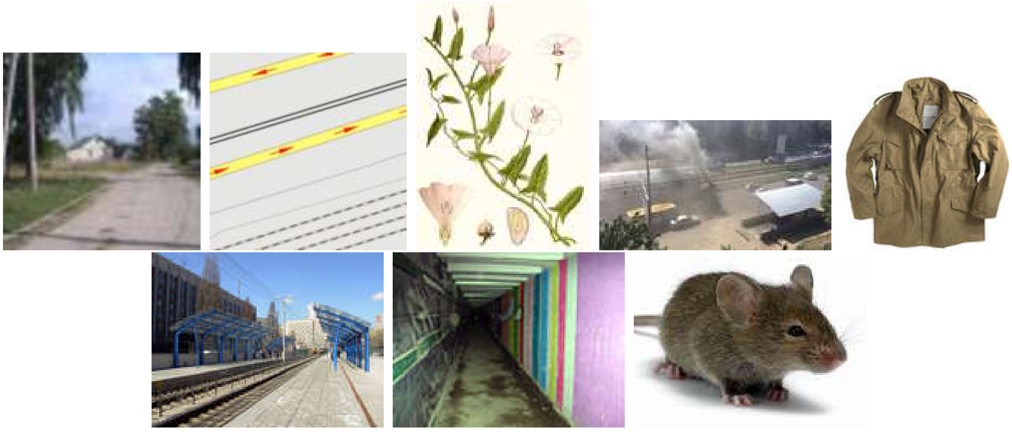


Figure 6.8: Illustration of canonical views of the Polyova (Kiev Light Rail) when using also the external imageset

Only cluster 6 contains an image from the internal imageset. This is also probably the only correct canonical view of the station (image 4 and 7 is undecidable). Only clusters with more than two images are shown.

Chapter 7

Large-Scale Image Retrieval

In this section, the evaluation method is formulated, the query image choice is described and results for different image retrieval systems are presented. For the image retrieval itself, a simple nearest neighbor with the Euclidean distance is used because the network VGG was trained using this distance. The same visual descriptor as in Section 6.1 is used.

Two different metrics are used to measure the system performance. One is measuring the average precision for the top N results. The second metric is described as a part of the evaluation method in the following section. For the specific object recognition which retrieves canonical views of any Wiki landmark, a simple voting system among retrieved results is implemented. The landmark with the most votes among the top 10 results is the recognized Wiki landmark. In case that two landmarks have the same number of votes, the position of the top result among them decides. This creates a robust system that, even if the first result is incorrect, can still correctly recognize the landmark.

7.1 Evaluation method

For the image retrieval task, only the Wiki core imageset is used because images there are manually annotated, so no further validation of the images is necessary. The images are annotated, but only with their Wiki category (topic), not the object they capture. This makes it hard to estimate the total number of relevant images that should be retrieved for a query. This makes it imprecise to estimate false positive rate. Furthermore, because of the Wiki category overlap, images of the same object can be found under multiple categories. Because retrieving an image of the same object but from a different category is not a mistake for the image retrieval system, this makes it hard to estimate true negative rate as well. Lastly, it cannot be ensured that duplicate categories will not exist, even though this phenomena was not observed.

Because precision and recall could not be measured, another evaluation metric is proposed. A query is considered successfully solved, if at least one out of top N images is correctly recognized. Then, the average number of queries successfully solved is measured for different N values. This evaluation metric is justified in [13] through other literature and denoted as "common at N". As all images from both Wiki core imageset and external imageset are interlinked, the true relevance of images is known. This metric makes query expansion redundant.

The main advantage of the metric is that it is nondecreasing, so the upper bound as well as the lower bound can be deduced. This provides a valuable insight into the image retrieval error components.

7.2 Query Image Choice

Random images from the external imageset were chosen for the set of query images. Because these images do not necessarily depict the same object as related images from the core imageset, the following criteria were applied to the query images.

- It must have the shorter edge at least 250px - *the descriptor is not scale-invariant*
- The category of the query image contains at least three images - *to rise the probability that there is an image of the same object in the core imageset*
- It is the first result from Google, retrieved by an English label - *to ensure maximum possible relevance based on data from Section 5.5*
- **The image must be a result of two textually unrelated queries**

The relevance of the images that the online databases offer depends on many factors, as shown in Section 5.5. To maximize the probability of getting a reasonable image for all Wiki Landmarks, the first result from Google of an English query was taken. This, nevertheless, did not protect from taking a completely unrelated image when multiple objects share the same label, as discussed and shown in Section 5.4. For every Wiki Landmark, titles in all languages were used to obtain relevant images from online databases. When the same image was retrieved using two distinct text labels, it means that the result is not dependent on the label itself, but rather on what the label represent. The reachability of the same image using two unrelated labels is a consequence of the search engine indexing the same image with two different labels. This can happen either on sites that offer multilingual labels for their images (e.g. Wikimedia Commons) or on two unrelated sites in two languages showing the exact same illustrative image. The difference of the labels is measured by the Levenshtein distance and it must be at least 7. Both query results referencing the same image must be among the top ten results of the two queries.

This condition fully replaced any need for utilizing the visual information of external core images. That did not introduce any additional complexity to the query image choice process and it is independent of any computer vision method which is beneficial as the performance of a computer vision algorithm is being measured.

7.3 Results of Image Retrieval

The image retrieval system was tested on the core imageset. The results are presented in Figure 7.1 and serve as a lower bound for the external imageset queries. The results are modest, mostly because of the fact that Wikimedia Commons presents intentionally diverse views. To compensate this, categories without at least 3 images were filtered out. The impact of the minimum number of images in a category is presented in the same figure.

To provide an upper bound for the external imageset queries, a multi-view image retrieval is performed and up to top 20 results for each query view are taken. This is demonstrated in Figure 7.2 with the best achievable result marked. This best achievable result is based on the manual annotation of query results from Section 5.5.

The results for a single-view large-scale image retrieval system are presented in Figure 7.3. These results are compared with the vanilla multi-view image retrieval system from the previous paragraph - for every view, the single top result is taken which yields at most 20 result images for each query. Taking the first result for at most 20 views outperformed taking 20 results of a single, even though carefully chosen, view.

As this dataset is to be used to measure false positive rate of the image retrieval systems, a standard graph of average precision is provided in Figure 7.4. The single-view queries from the external imageset are used.

The specific object recognition algorithm was able to successfully recognize a landmark in 55% of cases. This was tested using a sample of randomly chosen 500 queries.

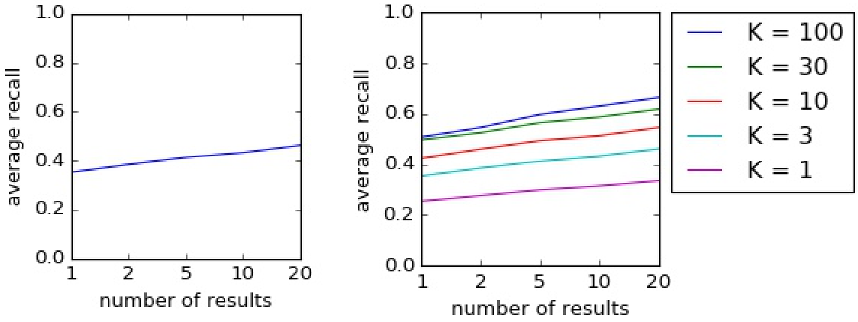


Figure 7.1: Average recall ("common at N") of queries from the core imageset. All images are from a category with at least 3 images (left) or K images (right).

Filtering out categories without at least K images raises the probability that multiple images of the same object exists in that category. But it does not guarantee that the specific object of the chosen image will have multiple images. A random sample of 2000 images was used.

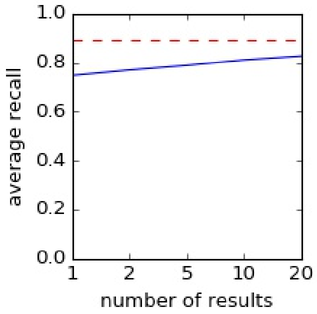


Figure 7.2: Average recall ("common at N") of multi-view queries from the external imageset

The top red line marks the best achievable result, based on the manual evaluation from Section 5.5. For each "number of results", at most 20 queries were performed and the "number of results" taken from each of them yielding at most 20 images for every result position. Results are based on 5240 randomly chosen queries.

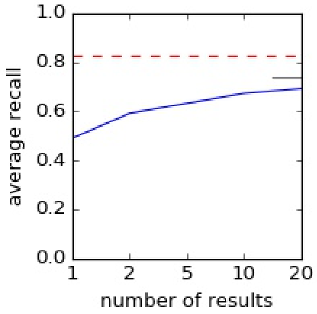


Figure 7.3: Average recall ("common at N") of single-view queries from the external imageset

The top red line marks the best achievable result, based on the multi-view query results from Figure 7.2. The gray line marks the results for the vanilla multi-view image retrieval where at most 20 queries were performed and the top result of each of them was taken. Results are based on 500 randomly chosen queries.

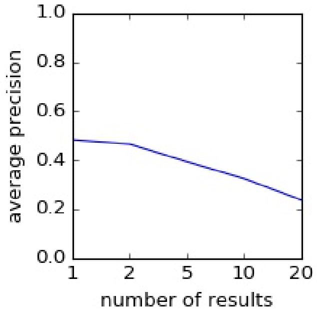


Figure 7.4: Average precision of single-view queries from the external imageset

Results are based on 500 randomly chosen queries.

7.3.1 Error Estimation of Different Factors

The performance of the system was influenced by many factors. Namely it were mistakes in the online image databases and therefore in the external imageset, the error caused by the image descriptor and the query image choice algorithm which caused incorrect images to be used as queries. The proportionality between these errors was estimated using the metric described in Section 7.1 and for the number of results equal to 20. Even though providing a valuable insight, all these results are relevant only with respect to this setup.

For the external imageset error quantification, data from the manual annotation from Section 5.5 were used. The difference between the data and an ideal state where for every set of the top 20 results, at least one image of the query object is provided, was 11 percent.

The image descriptor error was estimated by comparing manually annotated data to the multi-view query image retrieval system. The difference was 7 percent.

The error caused by the query image choice algorithm was modeled by the difference between the multi-view average recall and the single-view average recall which was 16 percent.

7.4 Sample Output of the Image Retrieval System

Examples of the output of the image retrieval system are provided in Figure 7.5 and Figure 7.6. In the second, it is worth noting that even objects generally hard for the image retrieval provide reasonable results. More examples of output are provided in Appendix E.

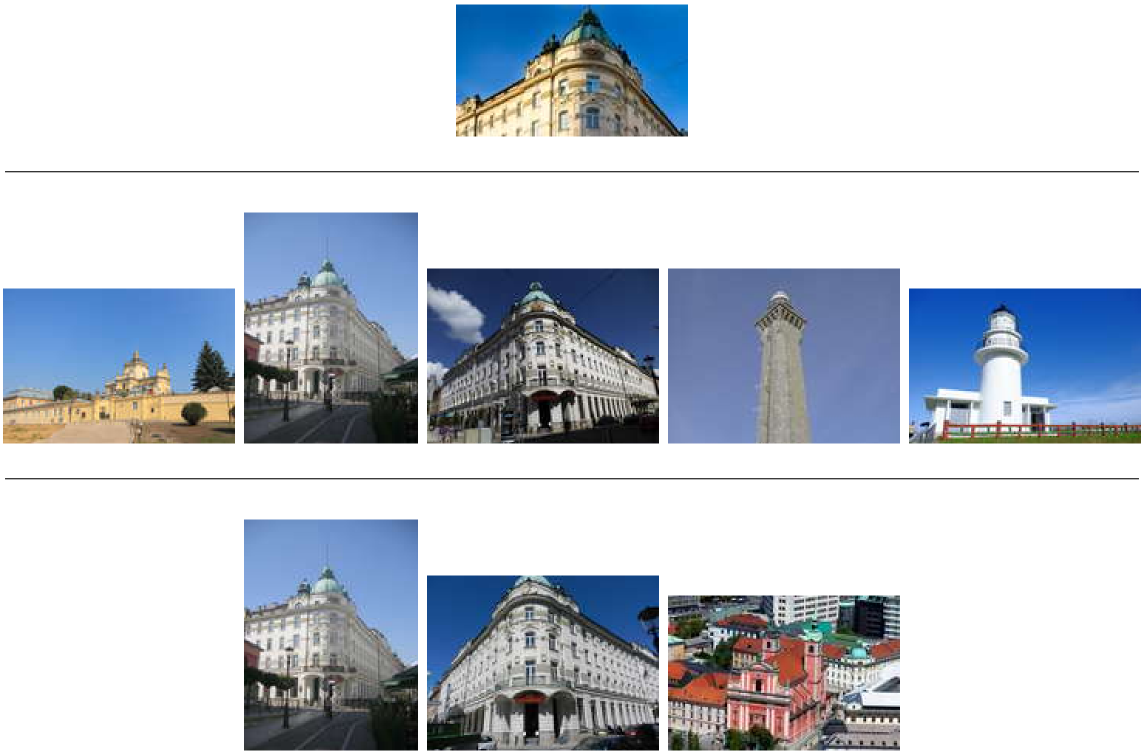


Figure 7.5: The image retrieval system sample output for Grand Hotel Union with a query image (top row), retrieved results (middle row) and the canonical views for the correctly identified landmark (bottom row).

From the results (middle row), the results 2 and 3 are correct, so the landmark is correctly recognized and the true canonical views are displayed. The correctly recognized Wikipedia article is https://en.wikipedia.org/wiki/Grand_Hotel_Union

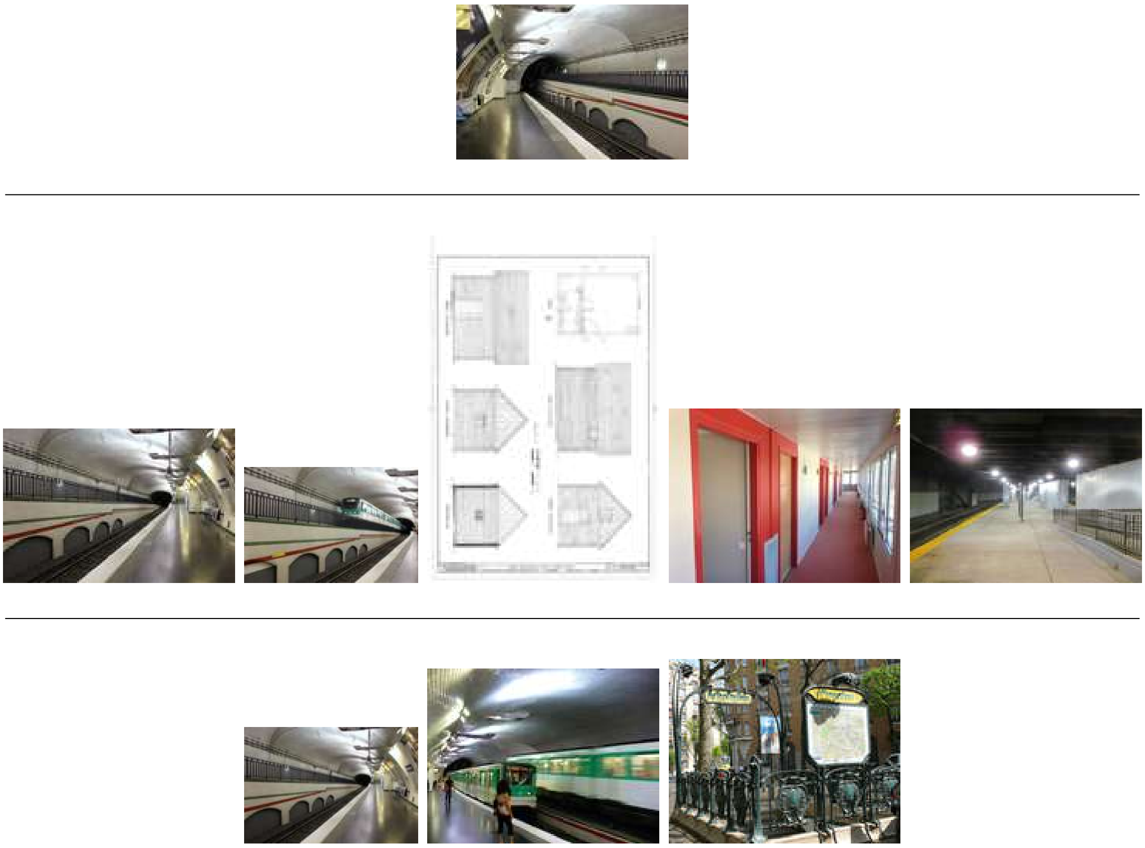


Figure 7.6: The image retrieval system sample output for Mirabeau (Paris Métro) with a query image (top row), retrieved results (middle row) and the canonical views for the correctly identified landmark (bottom row).

From the results (middle row), the results 1 and 2 are correct, so the landmark is correctly recognized and the true canonical views are displayed. It is worth noting that even objects generally hard for the image retrieval provide reasonable results. The correctly recognized Wikipedia article is https://en.wikipedia.org/wiki/Mirabeau_(Paris_M%C3%A9tro)

Chapter 8

Software Architecture

Putting all pieces together was the most time-consuming part of the processing. To save some of this time to others interested in implementing this pipeline, the implementation part of the work is briefly documented in this section. There is an emphasis on maintainability and repeatability of my work in this section.

This section presents the following topics - updating part of data, all data re-processing, what data are retrieved, SQL backend deployment, employed software design patterns and used technologies on the server side. The topics are presented in the order of their depth. The topics document updating different parts of data, different tools of the system, system maintenance and some implementation details such as duplicate prevention. The last section addresses a couple of issues that occurred together with their solutions.

The canonical view identification and image retrieval implementation are not addressed here, as they did not present a challenge from the implementation point of view. Only topics related to the framework for dataset retrieval and maintenance are described.

8.1 Dynamics of the Data

The whole processing pipeline is designed to be re-ran any time. It is possible to stop any task and when started again, only the unprocessed data will be scheduled for processing. This is useful not only when the processing ends with an error for various reasons, but also when the data slightly change, for example after patching them, so that there is no need to process the whole batch again. This is the major advantage of master server architecture. This also made possible the next design feature that allows working with data from many sources in many revisions. It is possible to mix data sources while deduplicating identical data, so that the intersection of data sets is processed only once. Also, it allows to update the data, mark them as a new revision and re-run the whole processing pipeline while processing only the difference between old and new data. This is an essential part of the architecture because performing a query and computing an image descriptor are both expensive operations worth minimizing.

Updating the data schematically consists of the following steps.

1. download DBpedia or Wikidata set of dumps of desired revision

2. mark the revision of the data so that it is propagated in the database
3. re-run the parsing
4. re-run the queries

Description of the "parsing" or "queries" step is provided in the next paragraphs.

8.2 The Processing Pipeline

To perform different tasks with the data, the processing was split into separate steps of a processing pipeline. All steps can be performed simultaneously, even though it is not always desirable. It is recommended to run the steps parsing the input data first, then manually check the parsed data and then run the querying steps. Some steps depend on others, but the steps can still be performed simultaneously, one generating data while other consuming them. The following sequence satisfy all dependencies.

Parsing steps:

1. **DBpedia** - retrieve categories
2. **DBpedia** - retrieve entities
3. **Wikidata** - retrieve categories
4. **Wikidata** - retrieve entities (depends on step 3)
5. **Wikidata** - link data (depends on step 2 and 4)
6. **Wikimedia** - link data (depends on step 2 and 4)
7. **Wikimedia** - enrich image data (depends on step 2, 4 and 6)

Querying steps:

1. **Plan** queries
2. **Link** queries with their source (depends on step 1)
3. **Execute** queries (depends on step 1)
4. **Parse** queries (depends on step 3)

This sequence should demonstrate the steps performed to obtain the dataset and provide the first source of information when replicating the work performed here - each step corresponds to one program to be run. The queries used for the retrieval of different sets of data corresponding to the first 4 parsing steps are explicitly listed in Appendix C.

8.3 Retrieved Data Summary

For each page about a landmark in Wikipedia and Wikidata, its title in a subset of 390 languages and GPS coordinates were kept. In case of Wikipedia, also a short abstract corresponding to the first paragraph the article and the article URL address were retrieved. Furthermore, the estimated page category together with the hierarchy among categories was stored. These data were retrieved from interlinking two data sources - DBpedia and Wikidata - and landmarks appearing in both were marked identical.

The set of Wiki core images was retrieved from Wikimedia Commons where people upload images into categories. These categories were mapped to Wiki landmarks and images contained in a category were associated with the landmark. Together with images, the alt fields and the image captions were downloaded. The alt field is contained in the HTML image tag and presents a valuable text information about that image.

To extend the Wiki core images, related images were retrieved from five different public online image databases. For the queries performed against them, titles in all 390 languages were used, together with GPS coordinates in case of Flickr. After all the queries were performed, in a distributed manner, image URL addresses, alt fields, page URL addresses, images dimensions and sizes in bytes were extracted from the responses. These data were kept for 131 million images.

8.4 Central Storage Backend

To handle the data size while keeping a certain level of processing speed, all steps were heavily parallelized. The degree of parallelization depended on a task's resource usage patterns. To guarantee that each task is not processed more than once, a central master server in the role of coordinator, or broker, was utilized. Also, a massive deduplication was demanded to process the smallest amount of data possible in each step. This was necessary because the ratio of duplicated files was extraordinary high. It was caused by the fact that multiple independent sources of data were used and that they overlap significantly. Also, because of other bottlenecks such as network speed, many processing steps of the pipeline had to be performed simultaneously. The master server architecture allowed that without any overhead as steps very rarely overlapped in terms of shared resources on the database side.

For the master server, a PostgreSQL database server was chosen and the specific SQL schema used is illustrated in Figure 8.1. The visualization of the SQL schema corresponds to different phases of processing being as follows.

1. Input data parsing (blue color)
2. Query performing (yellow color)
3. Linking parsed data and query results (purple color)

Input data parsing (blue color) corresponds to the DBpedia, Wikidata and Wikimedia Commons processing. In this step, the categories, where applicable, were kept including

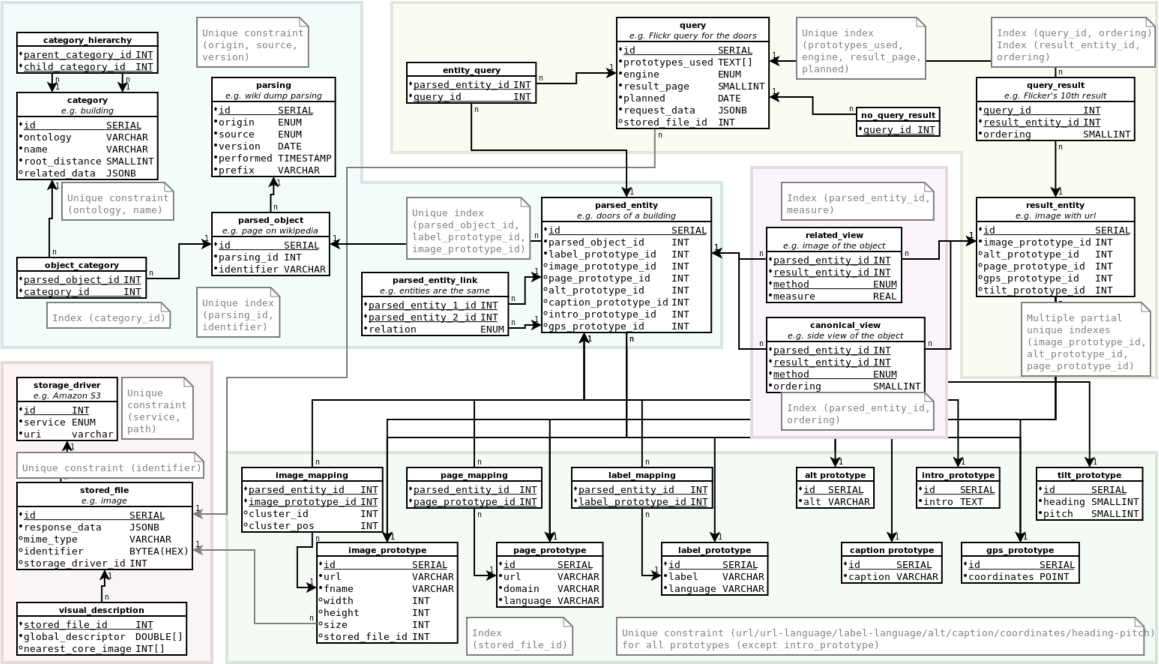


Figure 8.1: The SQL scheme containing all data and coordinating workers.

their hierarchy. The result of this processing was stored in the table "parsed_entity". Data from the table "parsed_entity" served as an input for queries which represents the contents of the next step, query performing (yellow color). This step planned, executed and parsed queries of different online services. The result of the executed query parsing was stored in the "result_entity" table. The tables "parsed_entity" and "result_entity" were linked together according to the results of a clustering and these links were put into two tables - "related_view" and "canonical_view" (purple color). The first one links together views of the same object, the other one reduces the views to a set of canonical views.

Furthermore, the following data structures stored as tables in the database were used:

- image_prototype - an image
- page_prototype - a page of that image
- label_prototype - a label linked to that image
- alt_prototype - an "alt" attribute of an "image" HTML tag found on the page
- caption_prototype - a caption of that image on the page
- intro_prototype - a short text affiliated with the image
- gps_prototype - gps coordinates linked to that image
- tilt_prototype - a special information about the camera tilt in both horizontal axis and vertical plane used only in StreetView processing [1]

[1] StreetView processing was not performed in this work

Tables "*_mapping" such as "image_mapping" (green color) are only n-n relations so that one parsed entity can have multiple prototypes of that type assigned. Finally, the following tables took care of data storage (orange color).

- stored_file
- storage_driver

A storage driver could be a service (e.g. S3 or XtreemFS) or a directory path on the host. The storage drivers also defined a specific structure of stored files.

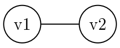
The downside of the formally strong model is caused by the deduplication requirement. Because of that, two queries are necessary for every row insert - SELECT and INSERT query. This slowed down the row inserts massively for the data size. It was compensated using aggregating INSERT queries into batches executed at once, but this is to be avoided in the oncoming architecture.

8.4.1 Downsides of a Flat Data Representation

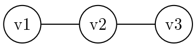
To comply with the relational database architecture, data were stored in a normalized form as rows in tables. In this format, modeling certain relations is a difficult task. In this case, Wiki Landmarks came from three independent sources mutually interlinked and therefore forming graphs. Every row in the table "parsed_entity" corresponds to one Wiki Landmark from one source. The table "parsed_entity_link" then models the interlinking between Wiki Landmarks from different sources. When seen from the graph theory perspective, "parsed_entity" represents vertices and "parsed_entity_link" edges of each graph. This makes certain tasks performed on graphs much more difficult. Also, different approaches have varying computational complexity. As an example, a simple graph counting was chosen.

Using one SQL query, only the degree of vertexes can be deduced. It is unnecessary to reconstruct the whole graph as the degree of vertexes is enough to get the number of graphs. The following graph combinations are valid. The valid graph combinations are determined by the process of data obtaining.

deg(v1) = 0



deg(v1) = 1, deg(v2) = 1



deg(v1) = 1, deg(v2) = 2, deg(v3) = 1

$$\boldsymbol{number\_of\_graphs = |\{v|deg(v) = 0\}| + |\{v|deg(v) = 1\}|/2}$$

where $|\{v|deg(v) = 0\}|$ is the cardinality of the set of vertices having their degree equal to zero.

This means a single SQL query can count the number of graphs using the table of edges only and replace a costly graph reconstruction process.

Iterating through graphs is even more complicated. To address this, a temporary table "meta_entity" was introduced creating a pre-computed view on data in "parsed_entity". Every row in "meta_entity" have an *id* column uniquely identifying a graph and a *parsed_entity_id* column referencing the corresponding record in "parsed_entity" table. Furthermore, it has columns "chosen", "processed" and "clustered" to set different flags related to choosing a random sample and working with it. The quickest way of creating this table is to import all records from "parsed_entity" and then modifying the *id* column according to the "parsed_entity_link" table.

8.5 Software Design Patterns

As the data amount grew, more and more sophisticated approaches had to be used. A master-worker software architecture was implemented for tasks where low amounts of structured data had to be saved, e.g. downloading images. Master acted as a broker, or a coordinator, and was implemented using an SQL database which offered locking capabilities. This ensured that workers had been synchronized at the level of data access, so that multiple workers could not edit the same data without knowing about each other. On the other side, image download and writing to hard disk was running in parallel on the workers. This ensured that the whole bandwidth of 100Mbps was utilized and that there was no bottleneck on the implementation side.

For structured data write intensive tasks, a different approach had to be chosen. An example is retrieving metadata where the retrieval had to be processed and written to the database. Compared to the image download, the task was much more data writing intensive on the database side. The approach of multiple workers running independently of each other could not be used because the locking of shared resources becomes quickly a bottleneck of every distributed system. A map-reduce inspired approach was used instead. The processing was split into two parts. The map part was the part being parallelized and contained metadata download and parsing. The reduce part took the processed data and putted them into the database in the quickest way possible (aggregating INSERT queries so that all values are inserted into a table at once). This reduced the overhead of locking on the database side and allowed to utilize the whole bandwidth of 100Mbps without any bottleneck on the implementation side.

Because the database was at all times utilized heavily, it was not desirable to execute long running queries. Also the RAM of the computer was a limiting factor because when used by the application, it could not be used by the database nor IO cache with a big impact on processing speed. To accommodate these needs, a map-reduce epoch-driven approach was introduced. This was inspired heavily by distributed systems where master-master replication is necessary. All data to be processed were split into a separate map-reduce epochs. These epochs did not share any data and were independent of each other, thus allowing the processing of them to overlap. Each time the reduce part of an epoch was finished, another epoch was planned. Multiple epochs could be processed at the same time, so one epoch was never waiting for another. This allowed to process all data in chunks and thus using the minimal RAM possible, corresponding to a couple of epochs only and not stressing the database with exhausting SQL queries. All of which without any processing speed impact.

These two architectures were a result of a series of benchmarks and profiling different parts of the processing pipeline and are considered the best solution for the tasks.

8.6 Used Technologies

A variety of technologies was used to process and keep the data. The following list contains the most heavily utilized ones, together with a brief description of them.

- Distributed file systems
  - GlusterFS
  - CephFS
  - XtreemFS
- Databases
  - OpenLink Virtuoso
  - PostgreSQL
  - Redis

- Deployment platform
  - Docker

GlusterFS is a distributed drop-in replacement for a NFS share server and in this pipeline was used to store big files such as dumps and database backups. There is no overhead of running GlusterFS when compared to NFS because it is a truly distributed solution without any master or metadata servers. CephFS is a distributed file system solution for LAN containing both master and metadata servers. This makes it much faster for the file metadata retrieval tasks as it do not have to access the underlying file system and these queries are answered from the database on the metadata server. CephFS was used for quickly changing data in a combination with various synchronization programs. XtreemFS is a very interesting project and presents essentially the only implementation of a WAN distributed file system available. It was used to share data on the file system level between workers in different geographic locations.

OpenLink Virtuoso is a multipurpose database used as a NoSQL triplet database to provide data from DBpedia and Wikidata dumps that were in a TTL format. PostgreSQL is a traditional SQL database used as the main data store and worker coordinator for this project. Redis is a NoSQL key-value store used solely as a cache of repeating PostgreSQL queries.

The main deployment platform used was Docker. This tool utilizes container virtualization to provide repeatable assembly of software (called containers), perfectly suitable for encapsulation of both software programs and used implementation. This can be then distributed in a manner ensuring the exact same environment and code base on each host.

8.7 Issues of the Implementation Phase

A number of problems arose as the amount of data was being processed from which four most severe and time consuming were chosen to be described together with used solutions.

- Invalid data
- Big number of small files
- PostgreSQL indexes breakage
- Regular data corruption

Getting invalid data was a big issue. This issue appeared in essentially every step of processing and was mostly related, but not limited to, incorrect encoding and standard disobeying. Incorrect encoding was a big issue as 390 languages were processed. This was tightly related to disobeying standards mostly related to incorrect URL and URI addresses which is common on the Internet, but should not appear in datasets from DBpedia and Wikidata. Because of this issue, measures had to be taken. This included data validation every time an external source of data was used. Also Apache Jena Fuseki software could not have been used as it was not robust enough to deal with this issue.

Big number of small files is a problem for every single file system. In this case, 6 millions of in average 30kiB files were stored on a Btrfs file system on top of cheap high-capacity hard disks. Various issues were related to that. First of all, the performance of some operations such as listing all files in a folder or, less obviously, getting metadata of a single file, was heavily influenced by the number of files in that folder. After some research, the issue seems to be the inode tree is no longer able to fit in the memory. This was solved by splitting the files into folders, based on the first three letters of their file name, so that at most 4096 files were in one of 4096 folders. The next issue was tightly related to the Btrfs file system. The free space running low in a combination with a rapid usage pattern change can leave the file system in a condition when no more files can be added even though a free space is then created there, in this case over 30%. It is a consequence of the file system being copy-on-write. The issue was finally solved by extending the file system with another device, a USB flash drive, and removing it after a balance operation on the file system which fixed the issue effortlessly. Despite that, it presented a big slowdown in progress because this issue is not documented anywhere.

The PostgreSQL indexes breakage was related to the way of stopping the PostgreSQL daemon. It can and does, when stopped not gently enough, lead to data inconsistent writes which yields to data loss but more importantly to index breakage causing unique constraint to fail and duplicated rows to be saved. This was a serious issue and the only solution was to plan every restart of the PostgreSQL daemon in advance and give it time to shutdown itself gently. The immediate fix consisted of all data deduplication and index rebuild. This operation is very slow because of links between tables that have to be adjusted.

With the amount of data being processed, data corruption occurred regularly. This was amplified by using cheap high-capacity hard disks and moving data between them in order to utilize all possible space and by using cheap portable drives to transfer the data. Even though file systems such as Btrfs that should prevent silent data corruption were used, data corruption occurred so often it had to be counted with in every processing step. Solving the issue meant adjusting all software to handling possibly corrupted data and defining recovery strategies, with the most radical solution being re-downloading the file from the Internet.

8.8 Enhancing the Software Architecture

This implementation solution was designed with the emphasis on future extensibility and scalability as a result of a number of analysis performed. Despite the effort, the magnitude of the data exceeded all expectations, so the advantages of a master-server and data deduplication on the server side became quickly the weak link. The proposal for the next generation solution would be using a NoSQL database which can utilize sharding, distributing the load across multiple servers, without any impact on querying from the user point of view. The keys in the NoSQL database would correspond to the aggregation of columns of unique constraints in every table which would solve deduplication of data smoothly. This approach would be scalable and presents a solution to all problems such as data integrity presented in this section while having esentially no bottlenecks.

Chapter 9

Conclusion

In this work, the landmark object was formally defined and the definition used to identify landmark pages on Wikipedia and through them also in Wikidata. The ambiguity of the definition was discussed. These Wiki Landmarks were interlinked with other projects, most notably UNESCO and OpenStreet Maps which allowed measuring the Wiki landmark coverage.

For the set of 357 thousand Wiki Landmarks, two imagsets were obtained - the Wiki core imageset from Wikimedia Commons (1.1 million images and 1.1 million labels in 390 languages) and the external imageset from five distinct online image databases (131 million images and 107 million text tags from 100 million web pages using 5.7 million queries). Both imagesets contain a diverse set of images together with their metadata for each Wiki Landmark. From the set of 357 thousand Wiki Landmarks, 212 thousand have an image in the Wiki core imageset and all of them have an image in the external core imageset.

The Wikipedia data from April 2016, Wikidata data from August 2016 and Wikimedia Commons images from November 2016 were processed. The queries against Google Images, Flickr, Yahoo Image Search, Bing Images and Yandex Image Search services were performed in November 2016. The tools for data refresh were provided and described.

The core imageset was reduced to canonical views by clustering the visual descriptors, computed for the images. After a comparison with the mean shift algorithm, the DBSCAN clustering algorithm was chosen to cluster core images for all Wiki Landmarks. The annotated nature of Wiki core images helped to partition the space of descriptors, so that clusters could have been found quickly using an augmented version of the standard DBSCAN algorithm implementation. Canonical views were identified as the closest images to the cluster equilibrium.

An image-based retrieval system able to retrieve canonical views and Wikipedia description of any Wiki Landmark was built from the core imageset. The ability to retrieve any image acquired in a broad range of conditions was ensured by the properties of the visual descriptor and by the diversity of the Wiki core images. An evaluation protocol was defined and two sets of input query images were carefully chosen. The performance of two versions of the system - a single-view and multi-view - was measured and compared. The single-view version, when taking the first result only, had an average accuracy of 48%.

9.1 Possible Extensions

The interlinking between multiple separated projects is presented in this work. These include Wikipedia, Wikidata, Wikimedia Commons and OpenStreet Maps. The next step is to enrich the projects based on the linked data. This applies to Wikipedia and Wikidata where the GPS coordinates can be validated through the OpenStreet Map project, to Wikimedia Commons where unrelated images together with duplicate categories can be identified through described computer vision methods, and to OpenStreet Map places which can be directly linked to Wikipedia articles and augmented through them.

The landmark definition and Wiki page landmark identification, forming the base of the linkings between projects, can be fine-tuned based on this work. In this work, problems with the definition are described, mistakes in Wikipedia illustrated and the issues with a semantic gap postulated. Solving these points would mean a higher quality for the Wiki Landmark datataset. Images from both imagesets could be further validated to ensure even higher quality.

An interesting non-computer-vision algorithm for image validation is presented. It utilizes the fact that images have multiple different labels assigned as they were retrieved through queries in multiple languages. When used in a combination with a computer vision validation, it can offer both speed and robustness.

In case of the image-based retrieval system, the potential for improvement is great. A vanilla multi-view image-based retrieval system outperformed the presented single-view system. Solving the multi-view image retrieval task would mean another step towards visually described semantic concepts, an ImageNet-like database but with specific objects. In a combination with the presented dataset, it could be utilized in numerous applications, such as web page topic recognition using referenced images only or location estimation from a smartphone camera.

Lastly, applying more advanced specific-object recognition methods to the imagesets, relations between images can be modeled and that can substantially augment the existing knowledge databases working with text and semantic descriptions only.

Bibliography

[1] AUER, S. – LEHMANN, J. What have innsbruck and leipzig in common? extracting semantics from wiki content. In *European Semantic Web Conference*, s. 503–517. Springer, 2007.

[2] CRANDALL, D. J. et al. Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, s. 761–770. ACM, 2009.

[3] DENG, J. et al. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, s. 248–255. IEEE, 2009.

[4] GABRILOVICH, E. – MARKOVITCH, S. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*. 2009, 34, s. 443–498.

[5] HAYS, J. – EFROS, A. A. IM2GPS: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, s. 1–8. IEEE, 2008.

[6] KENNEDY, L. S. – NAAMAN, M. Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th international conference on World Wide Web*, s. 297–306. ACM, 2008.

[7] LEHMANN, J. et al. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*. 2015, 6, 2, s. 167–195.

[8] MILLER, G. A. WordNet: a lexical database for English. *Communications of the ACM*. 1995, 38, 11, s. 39–41.

[9] RADENOVIĆ, F. – TOLIAS, G. – CHUM, O. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. *arXiv preprint arXiv:1604.02426*. 2016.

[10] RUSSELL, B. C. et al. 3d wikipedia: Using online text to automatically label and navigate reconstructed geometry. *ACM Transactions on Graphics (TOG)*. 2013, 32, 6, s. 193.

[11] SIMON, I. – SNAVELY, N. – SEITZ, S. M. Scene summarization for online image collections. In *2007 IEEE 11th International Conference on Computer Vision*, s. 1–8. IEEE, 2007.

[12] SIMONYAN, K. – ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 2014.

[13] TORII, A. et al. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, s. 1808–1817, 2015.

[14] TSAI, C.-M. et al. Extent: Inferring image metadata from context and content. In *2005 IEEE International Conference on Multimedia and Expo*, s. 1270–1273. IEEE, 2005.

Appendix A

Wikipedia Non-Landmark Categories Having GPS coordinates

The following list contains all categories of Wikipedia articles that can have GPS coordinates but are not landmarks. The GPS coordinates in these cases do not have to mark the location of a place, but for example can mark a location at one specific point in time (e.g. of a battle).

All these categories have the *dbo:* prefix in the DBpedia ontology.

Agent (= entity that acts - e.g. Organization, Person) - location of residence

Award - location of price award

ChemicalSubstance - location of discovery

Device - artifact discovery location

EthnicGroup - area of the ethnic group

Event (e.g. an earthquake) - location of that event

Food - place of origin

Holiday (e.g. a festival) - location of the event

MeanOfTransportation - place where the vehicle is operated or museum where is kept

Species - location of the specie native land

SportCompetitionResult - location of the sport competition

SportsSeason (e.g. results for a team during a season) - team base that season

UnitOfWork (e.g. a case decided by a court) - location of the event

Work (= a result of work - e.g. a sculpture or an atomic bomb) - its location

Name (e.g. a name echelon of rulers) - location of their ruling

Appendix B

Processed Languages

The list of every processed languages from Wikipedia and Wikidata is presented in a form of a table. It has the following columns.

Lang - a language code corresponding to either the IETF language tag, the ISO code or a PHP language code, in that order

Language name - an English name of the language (contructed languages are marked by a star after the language name)

Labels - the number of processed labels corresponding to the number of landmarks having a Wikipedia article or Wikidata document in that language

Uniq - the number of labels that appear in this language only - if ommited, it would equal to the number of lossed labels

Interestingly, the language frequency of landmark descriptions differ from the language frequency of Wikipedia articles more then expected (it is not even remotely proportional). Also, it is worth noting that some languages appear multiple times - the reason behind this is that programmers need to define not only the language but also used charset, so the same language written in different ways have mutliple language codes.

The only language that was excluded from the processing was the Gothic language with the language code *got* because it contained characters that were not in the unicode character set of the programs used for processing the labels. Also, according to Wikipedia, it is mostly extinct by the 8th or 9th century [1].

[1] https://en.wikipedia.org/wiki/Gothic_language

The specific SQL query used to retrieve the counts for languages is as follows.

```
SELECT label_prototype.language, COUNT(*) AS Labels, count AS Uniq
FROM label_prototype
LEFT OUTER JOIN (
        SELECT l1.language, COUNT(l1.id) FROM label_prototype l1
        LEFT OUTER JOIN label_prototype l2
                ON l1.label = l2.label AND l1.language != l2.language
        WHERE l2.id IS NULL GROUP BY l1.language
) t ON label_prototype.language = t.language
GROUP BY label_prototype.language, count
ORDER BY COUNT(*) DESC;
```

Lang	Language name	Labels	Uniq
en	English	410136	279093
fr	French	112138	55260
nl	Dutch	98286	34717
de	German	82086	31216
zh	Chinese	54230	41596
ja	Japanese	53815	52703
es	Spanish	50035	29472
sv	Swedish	49884	11852
it	Italian	42446	20800
pl	Polish	41466	21464
ru	Russian	38808	32127
ceb	Cebuano	31731	6308
nn	Norwegian Nynorsk	26442	5569
pt	Portuguese	24775	12251
nb	Norwegian Bokmål	24528	3500
fa	Persian	19015	18717
cs	Czech	18668	6842
ca	Catalan	17242	7350
ko	Korean	16644	16605
zh-hant	traditional Chinese	16570	5752
hu	Hungarian	15794	6784
fi	Finnish	15183	6352
uk	Ukrainian	14770	11941
eo	Esperanto*	14351	7493
ms	Malay	13778	9399
da	Danish	13089	2990
tg	Tajik	12518	12485
no	Norwegian	11342	740
ar	Arabic	11104	10895
zh-hans	simplified Chinese	10939	4010
vi	Vietnamese	10718	6178
id	Indonesian	10637	1089
zh-hk	traditional Chinese (Hong Kong)	10538	332
ro	Romanian	10459	5051
en-gb	British English	10278	28
eu	Basque	10018	3861
en-ca	Canadian English	9550	56
sk	Slovak	9213	2324
hr	Croatian	8550	1627
zh-cn	simplified Chinese (Mainland China)	8343	5838
et	Estonian	7634	2676
lt	Lithuanian	7613	6099
cy	Welsh	7008	2349
he	Hebrew	6728	3356
ka	Georgian	6642	6326
tr	Turkish	6620	4699
gl	Galician	6579	1030
sl	Slovenian	6161	1580
bg	Bulgarian	6132	4460
de-ch	German (Switzerland)	5785	345
oc	Occitan	5765	872
ga	Irish	5764	1520
sr	Serbian	5590	2908
pms	Piedmontese	5521	695
af	Afrikaans	5325	1092
in	Indonesian	5173	195
sr-latn	Serbian (Latin)	5154	737
br	Breton	4985	754
pt-br	Portuguese (Brazil)	4973	53
be	Belarusian	4944	4263
sco	Scots	4928	434
is	Icelandic	4820	647
lb	Luxembourgish	4647	608
el	Greek	4518	4413
cv	Chuvash	4314	1471
sh	Serbo-Croatian	4141	477
gsw	Swiss German	4107	263
nds	Low Saxon	4101	294
bar	Bavarian	4063	178
an	Aragonese	4061	319
gd	Scottish Gaelic	4041	346
io	Ido*	3980	462
sw	Swahili	3960	353

Lang	Language name	Labels	Uniq
ast	Asturian	3907	228
th	Thai	3868	3858
scn	Sicilian	3816	224
vec	Venetian	3796	222
iw	Hebrew	3765	425
rm	Romansh	3676	116
ia	Interlingua*	3672	121
co	Corsican	3659	64
li	Limburgish	3650	91
vls	West Flemish	3638	113
mk	Macedonian	3620	2641
de-at	German (Austria)	3614	1
vo	Volapük*	3608	45
nds-nl	Dutch Low Saxon	3602	62
hy	Armenian	3582	3557
sc	Sardinian	3567	61
nap	Neapolitan	3545	62
wa	Walloon	3545	53
mg	Malagasy	3519	30
lij	Ligurian	3514	45
pcd	Picard	3505	35
fur	Friulian	3503	49
fr-x-nrm	Norman	3492	39
ie	Interlingue*	3490	10
frp	Franco-Provençal	3463	30
min	Minangkabau	3456	12
wo	Wolof	3446	12
zu	Zulu	3433	2
kg	Kongo	3428	2
ur	Urdu	2988	2647
pnb	Western Punjabi	2818	2542
hi	Hindi	2641	2415
lv	Latvian	2590	2085
ta	Tamil	2468	2464
la	Latin	2435	1791
zh-tw	Chinese (Taiwan)	2283	149
hbs	Serbo-Croatian	2220	204
az	Azerbaijani	2208	1924
zh-sg	Chinese (Singapore)	1906	6
be-tarask	Belarusian (Taraškievica)	1734	1122
sr-cyrl	Serbian (Cyrillic)	1650	54
yue	Yue Chinese	1555	371
bs	Bosnian	1516	422
mr	Marathi	1450	1314
bn	Bengali	1401	1370
kk	Kazakh	1325	583
fy	Western Frisian	1313	762
qu	Quechua	1294	903
ml	Malayalam	1251	1250
pa	Eastern Punjabi	1116	1114
uz	Uzbek	1037	722
jv	Javanese	969	408
sq	Albanian	846	569
xmf	Mingrelian	760	467
lmo	Lombard	755	377
tl	Tagalog	748	495
war	Waray	744	472
ne	Nepali	665	561
ba	Bashkir	660	385
tt	Tatar	612	440
ky	Kirghiz	570	336
yi	Yiddish	550	327
fo	Faroese	512	62
te	Telugu	498	496
mn	Mongolian	496	414
nan	Min Nan	492	387
gan	Gan	467	221
arz	Egyptian Arabic	430	314
kn	Kannada	395	394
gu	Gujarati	391	387
kk-cyrl	Kazakh (Cyrillic)	353	5
kk-latn	Kazakh (Latin)	349	346
kk-arab	Kazakh (Arabic)	349	185
rw	Kinyarwanda	349	170
my	Burmese	345	343
kl	Greenlandic	338	23

Lang	Language name	Labels	Uniq
wuu	Wu	336	136
si	Sinhalese	331	330
als	Alemannic	328	16
ku	Kurdish (Kurmanji)	312	176
ckb	Kurdish (Sorani)	301	293
ilo	Ilokano	298	237
ji	Yiddish	274	77
os	Ossetian	255	153
mt	Maltese	255	198
hak	Hakka	246	245
hsb	Upper Sorbian	242	134
sa	Sanskrit	232	214
ce	Chechen	228	132
ps	Pashto	216	197
mrj	Hill Mari	213	157
nah	Nahuatl	208	69
am	Amharic	204	202
kab	Kabyle	200	45
sah	Sakha	199	150
su	Sundanese	193	111
se	Northern Sami	193	77
mzn	Mazandarani	189	158
new	Newar	187	158
pap	Papiamentu	184	96
bh	Bihari	184	20
vmf	German (Main-Franconian)	178	2
zh-mo	Chinese (Macau)	174	0
sgs	Samogitian	168	143
jam	Jamaican Patois	167	20
kk-cn	Kazakh (China)	159	0
kk-kz	Kazakh (Kazakhsta)	159	0
kk-tr	Kazakh (Turkey)	159	0
zh-my	Chinese (Malaysia)	157	0
bm	Bambara	150	4
frc	Cajun French	148	0
prg	Prussian	146	0

Lang	Language name	Labels	Uniq
rgn	Romagnol	146	0
vep	Vepsian	145	110
mhr	Meadow Mari	144	92
gan-hans	simplified Alekano	143	59
km	Khmer	143	141
gan-hant	traditional Alekano	141	2
ang	Anglo-Saxon	139	96
mis-x-rip	miscellaneous language	137	64
bo	Tibetan Standard	137	136
lad	Ladino	135	110
vro	Võro	128	48
tk	Turkmen	117	106
as	Assamese	117	89
dsb	Lower Sorbian	115	59
ht	Haitian	115	49
gv	Manx	114	21
hif	Fiji Hindi	114	79
or	Oriya	113	108
stq	Saterland Frisian	110	41
ku-arab	Kurdish (Arabic)	110	104
ku-latn	Kurdish (Latin)	109	5
lzh	Literary Chinese	109	20
nv	Navajo	106	104
gn	Guarani	102	73
zea	Zeelandic	101	53
szl	Silesian	101	49
kw	Cornish	97	60
udm	Udmurt	91	37
diq	Zazaki	91	58
pfl	Palatinate German	87	56
frr	North Frisian	86	43
mwl	Mirandés	86	43
ace	Acehnese	83	62
ay	Aymara	82	26
yo	Yoruba	82	62

Lang	Language name	Labels	Uniq
csb	Kashubian	81	62
azb	Southern Azerbaijani	80	62
sd	Sindhi	78	73
mai	Maithili	76	17
so	Somali	72	52
tg-latn	Tajik (Latin)	68	62
dv	Divehi	67	66
rue	Rusyn	66	35
pam	Kapampangan	65	21
ug	Uyghur	64	63
ext	Extremaduran	62	40
eml	Emilian-Romagnol	61	50
bcl	Central Bicolano	58	20
bxr	Buryat	57	43
na	Nauruan	57	13
it-x-tara	Tarantino	54	29
kbd	Kabardian	51	39
pdc	Pennsylvania German	50	21
rup	Aromanian	50	19
lo	Lao	49	46
arc	Aramaic	46	45
krc	Karachay-Balkar	46	27
ny	Chichewa	45	39
kv	Komi	43	20
bpy	Bishnupriya Manipuri	42	22
cdo	Min Dong	42	39
mo	Moldovan	42	9
av	Avar	41	32
ty	Tahitian	38	3
ln	Lingala	37	17
bjn	Banjar	34	15
mi	Maori	34	16
chy	Cheyenne	33	30
grc	Ancient Greek (to 1453)	32	27
kaa	Karakalpak	31	16
lez	Lezgian	30	11
bho	Bhojpuri	27	4
haw	Hawaiian	26	9
koi	Komi-Permyak	26	12
crh-latn	Crimean Tatar (Latin)	25	15
jv-x-bms	Javanese (Banyumasan dialect)	24	2
jbo	Lojban*	22	21
sn	Shona	21	12
bi	Bislama	20	17
cu	Old Church Slavonic	20	17
cbk-x-zam	Zamboanga Chavacano	20	5
tw	Twi	20	5
myv	Erzya	19	13
ltg	Latgalian	18	14
tn	Tswana	18	17
ab	Abkhazian	16	14
ady	Adyghe	16	8
pag	Pangasinan	16	5
za	Zhuang	16	14
fit	Tornedalen Finnish	14	0
pi	Pali	14	1
nov	Novial*	13	3
rn	Kirundi	13	2
gom	Goan Konkani	12	2
mdf	Moksha	12	4
ak	Akan	11	6
glk	Gilaki	11	6
ha	Hausa	11	6
lbe	Lak	11	8
tyv	Tuvan	11	7
ik	Inupiak	10	8
om	Oromo	10	4
pih	Norfolk	10	6
st	Sesotho	10	6

Lang	Language name	Labels	Uniq	Lang	Language name	Labels	Uniq
tet	Tetum	10	4	sg	Sango	2	0
to	Tongan	10	3	tg-	Tajik (Cyrillic)	2	1
xh	Xhosa	10	6	cyrl			
ff	Fula	9	7	ti	Tigrinya	2	1
pnt	Pontic	9	5	ve	Venda	2	1
ch	Chamorro	8	3	ary	Moroccan Spoken	1	0
chr	Cherokee	8	7		Arabic		
got	Gothic	8	7	kbd-	Kabardian (Cyril-	1	0
ki	Kikuyu	8	3	cyrl	lic)		
ks	Kashmiri	8	6	ruq-	Megleno-	1	0
lrc	Northern Luri	8	6	cyrl	Romanian (Cyril-		
srn	Sranan	8	1		lic)		
gag	Gagauz	7	2	avk	Kotava*	1	0
ig	Igbo	7	2	ng	Ndonga	1	0
rmy	Romani	7	3	lfn	Nova, Lingua	1	0
tpi	Tok Pisin	7	2		Franca Nova*		
bug	Buginese	6	0	ii	Nuosu	1	0
ee	Ewe	6	1	lzz	Laz	1	0
dz	Dzongkha	6	5	ks-	Kashmiri (Ara-	1	0
nso	Northern Sotho	6	1	arab	bic)		
sma	Southern Sami	6	5	ruq-	Megleno-	1	0
ss	Swati	5	2	latn	Romanian (Latin)		
tum	Tumbuka	5	2	tru	Turoyo	1	0
ts	Tsonga	5	4	hil	Hiligaynon	1	0
xal	Kalmyk	5	4	tt-	Tatar (Cyrillic)	1	0
brh	Brahui	4	3	cyrl			
iu	Inuktitut	4	2	hz	Herero	1	0
arn	Araucanian	3	2	niu	Niuean	1	0
sm	Samoan	3	2	liv	Livonian	1	0
sei	Seri	3	1	loz	Lozi	1	0
pdt	Plautdietsch/Mennonite	3	0	ug-	Uyghur (Latin)	1	0
	Low German			latn			
fj	Fijian	2	1	tcy	Tulu	1	0
ho	Hiri Motu	2	1	crh-	Crimean Tatar	1	0
ike-	Inuktitut (Syllab-	2	0	cyrl	(Cyrillic)		
cans	ics)			rif	Tarifit	1	0
ike-	Inuktitut (Latin)	2	1	cps	Capiznon	1	0
latn				anp	Angika	1	0
kr	Kanuri	2	1	aln	Gheg Albanian	1	0
lg	Luganda	2	0	kj	Kuanyama	1	0
mis-	Toki Pona	2	1	aa	Afar	1	0
x-				ruq	Megleno-	1	0
tokipona					Romanian		

Lang	Language name	Labels	Uniq
en-x-simple	Simple English*	1	0
dtp	Central Dusun	1	0
bcc	Southern Balochi	1	0
vot	Votic	1	0
ug-arab	Uyghur (Arabic)	1	0
mus	Muscogee	1	0
egl	Emilian	1	0
shi	Tachelhit	1	0
krj	Kinaray-a	1	0
sat	Santali	1	0
shi-latn	Tachelhit (Latin)	1	0
kiu	Kirmanjki	1	0
sli	Lower Selisian	1	0
qug	Kichwa/Northern Quechua	1	0
ks-deva	Kashmiri (Devanagari)	1	0
ko-kp	Korean (DPRK)	1	0
lus	Mizo/Lushai	1	0
shi-tfng	Tachelhit (Tifinagh)	1	0
tly	Talysh	1	0
inh	Ingush	1	0
jut	Jutish/Jutlandic	1	0
bqi	Bakthiari	1	0
de-x-formal	Formal German	1	0
cho	Choctaw	1	0
khw	Khowar	1	0
tt-latn	Tatar (Latin)	1	0
sdc	Sassarese	1	0
kri	Krio	1	0
hif-latn	Fiji Hindi (Latin)	1	0
mh	Marshallese	1	0
arq	Arabic, Algerian Spoken	1	1
cr	Cree	1	0

* A constructed language

Appendix C

SPARQL Queries for Wiki Landmark Retrieval

This section provides queries that were used to retrieve landmarks and their categories from DBpedia and Wikidata. The queries are in a SPARQL format which is the SQL equivalent for knowledge databases. These queries were not ran against the public API provided by DBpedia and Wikidata but the dumps of both were downloaded and a local mirror without any execution limits was built. For this purpose, the OpenLink Virtuoso Software was used.

First, Wiki Landmarks are retrieved from DBpedia (Listing C.1) together with category hierarchy (Listing C.2). The DBpedia category hierarchy retrieval is not necessary for consecutive steps. Next step is to identify which categories contain landmarks and how many of them (Listing C.3) and which categories contain non-landmarks and how many of them (Listing C.4). The last step was to use this information to retrieve Wiki Landmarks from Wikidata (Listing C.5).

In the queries, two variables are used: $CATEGORY_NAME for the name of a category the data are retrieved for and $PARENT_CATEGORY for the ancestor category name in the category traversal.

Listing C.1: DBpedia Wiki Landmarks Query

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo:  <http://dbpedia.org/ontology/>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?object as ?identifier_uri, ?en_label, ?image, ?latitude,
                ?longitude, ?page, ?comment,
        GROUP_CONCAT(DISTINCT CONCAT(?label, "@", LANG(?label)); separator=" ;; ")
                as ?labels,
        GROUP_CONCAT(DISTINCT ?type; separator=" ;; ") as ?types
        WHERE {

        ?object rdf:type dbo:Place .
        ?object rdfs:label ?en_label .
        ?object geo:lat ?latitude .
        ?object geo:long ?longitude .
        FILTER ( LANG(?en_label) = 'en' )

        OPTIONAL { ?object foaf:depiction ?image. } .
        OPTIONAL { ?object foaf:isPrimaryTopicOf ?page. } .
        OPTIONAL { ?object rdfs:comment ?comment. FILTER
                ( LANG(?comment) = 'en' ) } .
        OPTIONAL { ?object owl:sameAs ?object_in_language. ?object_in_language
                rdfs:label ?label. FILTER( LANG(?label) != 'got' ) } .
        OPTIONAL { ?object rdf:type ?type } .

        OPTIONAL { ?no1 rdf:type dbo:PopulatedPlace. FILTER (?object = ?no1) } .
        OPTIONAL { ?no2 rdf:type dbo:CelestialBody. FILTER (?object = ?no2) } .
        OPTIONAL { ?no3 rdf:type dbo:Crater. FILTER (?object = ?no3) } .
        OPTIONAL { ?no4 rdf:type dbo:Road. FILTER (?object = ?no4) } .
        FILTER ( !BOUND(?no1) && !BOUND(?no2) && !BOUND(?no3) && !BOUND(?no4) )
}
```

Listing C.2: DBpedia Category Hierarchy Query

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?category WHERE {
        ?category rdfs:subClassOf $PARENT_CATEGORY .
}
```

Listing C.3: Wikidata Positive Categories Query

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo:  <http://dbpedia.org/ontology/>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wdt: <http://www.wikidata.org/entity/>

SELECT COUNT(?wikidata) as ?count, ?category, ?label WHERE {

        ?dbpedia rdf:type dbo:Place .
        ?dbpedia geo:lat ?latitude_dbpedia .
        ?dbpedia geo:long ?longitude_dbpedia .

        OPTIONAL { ?no1 rdf:type dbo:PopulatedPlace. FILTER (?dbpedia = ?no1) } .
        OPTIONAL { ?no2 rdf:type dbo:CelestialBody. FILTER (?dbpedia = ?no2) } .
        OPTIONAL { ?no3 rdf:type dbo:Crater. FILTER (?dbpedia = ?no3) } .
        OPTIONAL { ?no4 rdf:type dbo:Road. FILTER (?dbpedia = ?no4) } .
        FILTER ( !BOUND(?no1) && !BOUND(?no2) && !BOUND(?no3) && !BOUND(?no4) )

        ?dbpedia owl:sameAs ?wikidata .
        ?wikidata rdf:type <http://www.wikidata.org/ontology#Item> .
        ?wikidata wdt:P31c ?category .
        OPTIONAL { ?category rdfs:label ?label. FILTER (LANG(?label) = 'en') } .

}
```

Listing C.4: Wikidata Negative Categories Query

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbo:  <http://dbpedia.org/ontology/>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX wdt: <http://www.wikidata.org/entity/>

SELECT COUNT(?wikidata) as ?count, ?category WHERE {
        {
                ?dbpedia rdf:type owl:Thing .
                OPTIONAL { ?no1 rdf:type dbo:Place. FILTER (?dbpedia = ?no1) } .
                FILTER ( !BOUND(?no1) )

                ?dbpedia owl:sameAs ?wikidata .
                ?wikidata rdf:type <http://www.wikidata.org/ontology#Item> .
                ?wikidata wdt:P31c ?category .
        } UNION {
                ?dbpedia rdf:type dbo:PopulatedPlace .

                ?dbpedia owl:sameAs ?wikidata .
                ?wikidata rdf:type <http://www.wikidata.org/ontology#Item> .
                ?wikidata wdt:P31c ?category .
        } UNION {
                ?dbpedia rdf:type dbo:CelestialBody .

                ?dbpedia owl:sameAs ?wikidata .
                ?wikidata rdf:type <http://www.wikidata.org/ontology#Item> .
                ?wikidata wdt:P31c ?category .
        } UNION {
                ?dbpedia rdf:type dbo:Crater .

                ?dbpedia owl:sameAs ?wikidata .
                ?wikidata rdf:type <http://www.wikidata.org/ontology#Item> .
                ?wikidata wdt:P31c ?category .
        } UNION {
                ?dbpedia rdf:type dbo:Road .

                ?dbpedia owl:sameAs ?wikidata .
                ?wikidata rdf:type <http://www.wikidata.org/ontology#Item> .
                ?wikidata wdt:P31c ?category .
        }
}
```

Listing C.5: Wikidata Wiki Landmarks Query

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX schema: <http://schema.org/>
PREFIX wdt: <http://www.wikidata.org/entity/>

SELECT DISTINCT ?object as ?identifier_uri, ?en_label, ?latitude, ?longitude,
        GROUP_CONCAT(DISTINCT ?image; separator=" ;; ") as ?images,
        GROUP_CONCAT(DISTINCT CONCAT(?label, "@", LANG(?label)); separator=" ;; ")
                as ?labels,
        GROUP_CONCAT(DISTINCT CONCAT(?page, "@", ?page_lang); separator=" ;; ")
                as ?pages,
        GROUP_CONCAT(DISTINCT ?type; separator=" ;; ") as ?types
        WHERE {

        ?object rdf:type <http://www.wikidata.org/ontology#Item> .
        ?object wdt:P31c wdt:$CATEGORY_NAME .
        ?object wdt:P31c ?type .
        ?object rdfs:label ?en_label .
        ?object wdt:P625c ?coors .
        FILTER ( LANG(?en_label) = 'en' )
        # Mandatory - must be checked manually (the OPTIONAL keyword leads to a better
                query execution plan in virtuoso)
        OPTIONAL { ?coors <http://www.wikidata.org/ontology#latitude> ?latitude. } .
        OPTIONAL { ?coors <http://www.wikidata.org/ontology#longitude> ?longitude. } .

        OPTIONAL { ?object wdt:P18c ?image. } .
        OPTIONAL { ?object rdfs:label ?label. } .
        OPTIONAL { ?page schema:about ?object. ?page rdf:type
                <http://www.wikidata.org/ontology#Article>.
                OPTIONAL { ?page schema:inLanguage ?page_lang. } } .
}
```

Appendix D

Sample Keyword Queries to Retrieve External Images

Two sample queries ran against five online databases are presented here - "Cleveland Burke Lakefront Airport" and "Waldau". The black cross on a white background denotes that there was an error during image retrieval causing the image download to fail.

In the "Cleveland Burke Lakefront Airport" query, the results of the Japanese text query for Bing and Yahoo can be considered relevant because these are the images from the event "Gran Prix of Cleveland" [1] held at the airport.

The "Waldau" query is also interesting. It is a name of a psychiatric hospital in Bern but the name "Waldau" is very ambiguous - it is not only a name for different places but also a name of an actor. Because of that, results of all queries are flooded with irrelevant images. Flickr spatiall query seems to be the only one that provides in some way relevant images.

[1] https://en.wikipedia.org/wiki/Grand_Prix_of_Cleveland

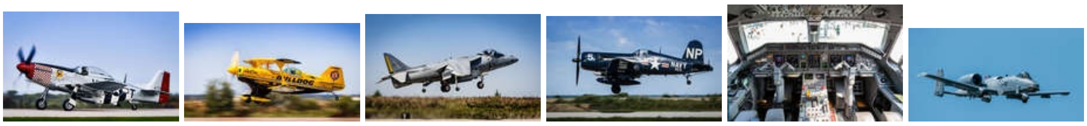
Sample Query 160881 - Cleveland Burke Lakefront Airport

Wikipedia

Wikimedia

Flickr

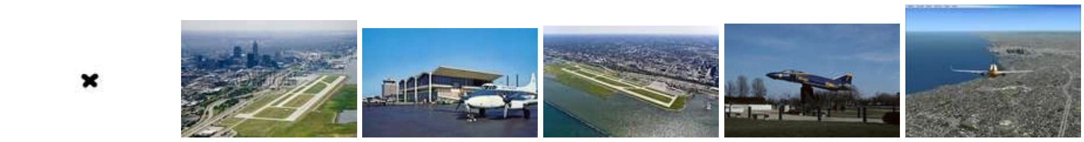
gps query: 41.5175, -81.6833



text query: Cleveland Burke Lakefront Airport (en, nl, sv)

Yahoo

text query: Cleveland Burke Lakefront Airport (en, nl, sv)



text query: Фурудгоҳи клюлнд брк ликфронт (tg)

text query: فرودگاه کلیولند برک لیکفرانت (fa)

text query: クリーブランド・バーク・レ イクフロント空港 (ja)

Bing

text query: Cleveland Burke Lakefront Airport (en, nl, sv)

text query: Фурудгоҳи клюлнд брк ликфронт (tg)

text query: فرودگاه کلیولند برک لیکفرانت (fa)

text query: クリーブランド・バーク・レ イクフロント空港 (ja)

Yandex

text query: Cleveland Burke Lakefront Airport (en, nl, sv)

text query: クリーブランド・バーク・レ イクフロント空港 (ja)

text query: 克里夫蘭Burke湖畔機場 (zh)

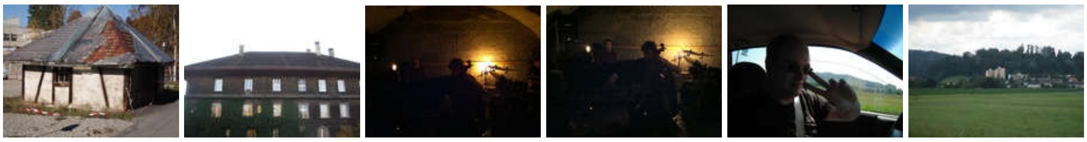
Sample Query 180825 - Waldau

Wikipedia

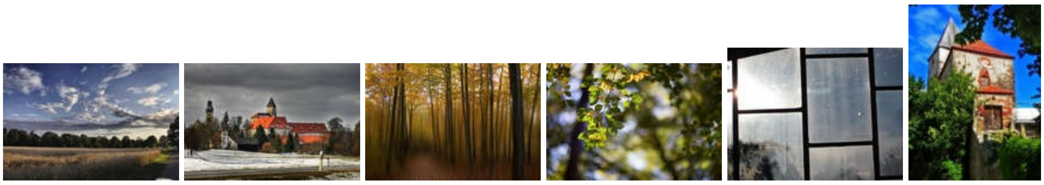
Wikimedia

Flickr

gps query: 46.9653, 7.48504



text query: Waldau (it, en, de)



text query: Вальдау (ru)

Yahoo

text query: Waldau (it, en, de)

text query: hôpital Waldau (fr)

text query: Βαλντάου (el)

text query: Вальдау (ru)

Bing

text query: Waldau (it, en, de)

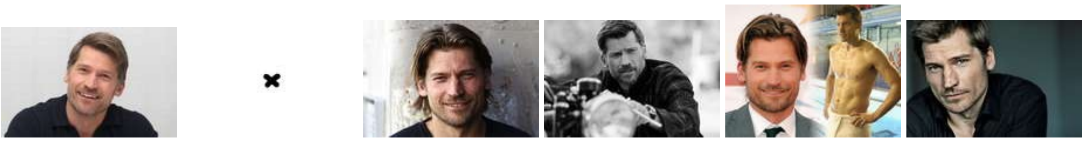
text query: hôpital Waldau (fr)

text query: Βαλντάου (el)

text query: Вальдау (ru)

Yandex

text query: Waldau (it, en, de)



text query: hôpital Waldau (fr)

text query: Βαλντάου (el)

text query: Вальдау (ru)

Appendix E

Image Retrieval System Output

Three more examples of the image-based retrieval system output are shown. The top row in the output illustration is the query image, retrieved results are in the middle row and the recognized canonical views are in the bottom row.

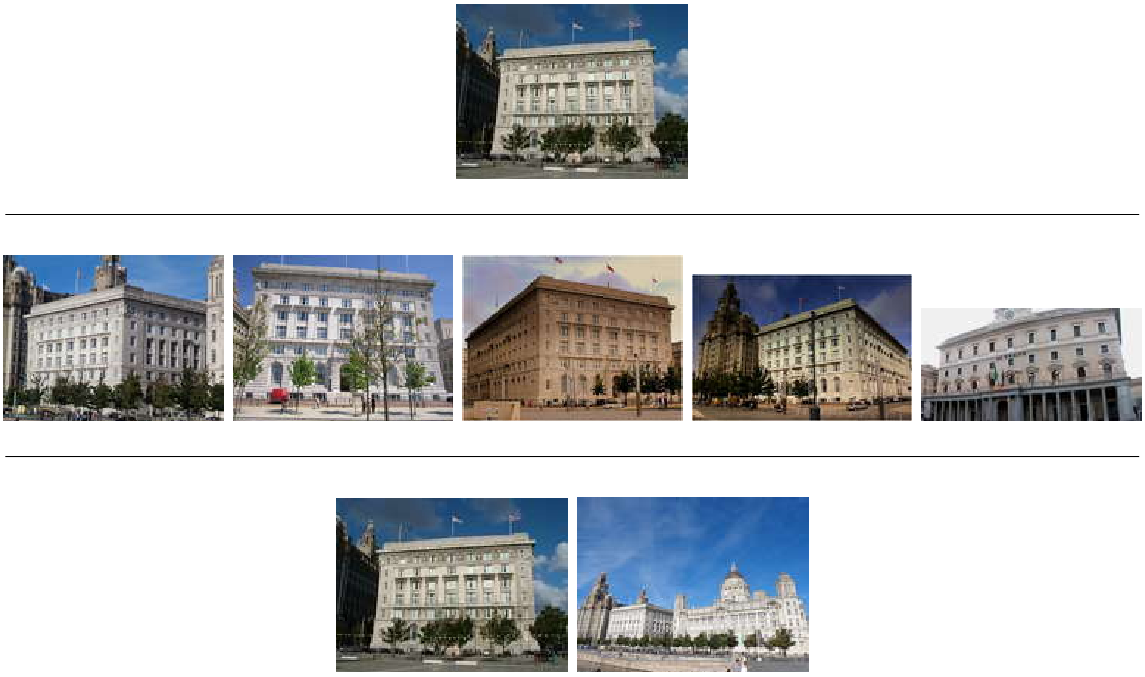


Figure E.1: The image retrieval system sample output for Cunard Building

The first 4 results are correct. The correctly recognized Wikipedia article is https://en.wikipedia.org/wiki/Cunard_Building

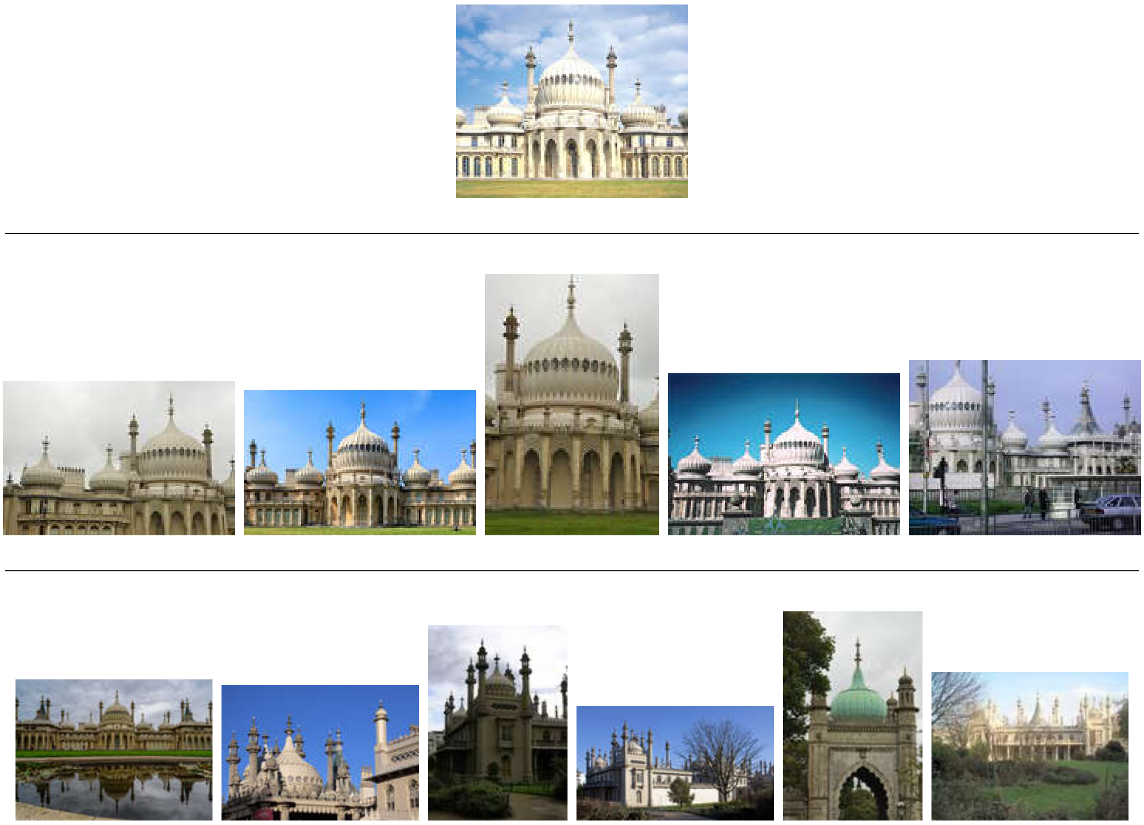


Figure E.2: The image retrieval system sample output for Royal Pavilion

All retrieved results are correct. The correctly recognized Wikipedia article is https://en.wikipedia.org/wiki/Royal_Pavilion

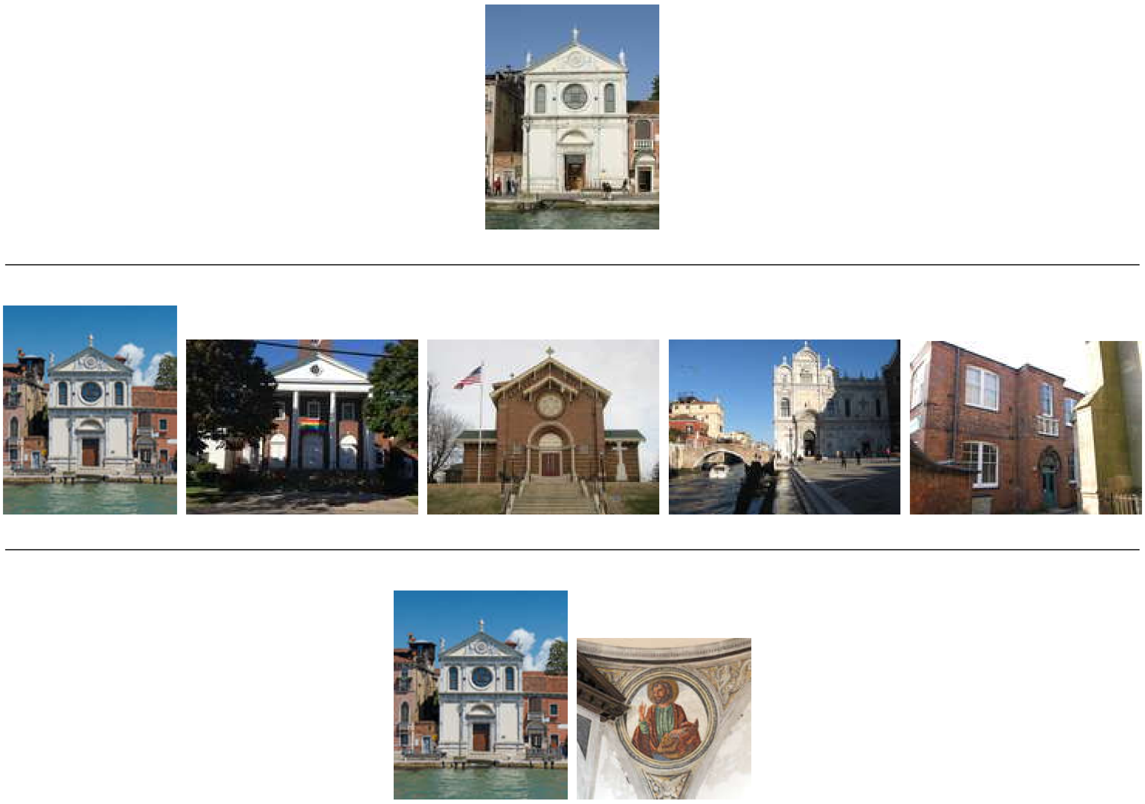


Figure E.3: The image retrieval system sample output for Santa Maria della Visitazione

Only the first result is correct. It is the only outside view of that landmark available on Wikimedia Commons. The correctly recognized Wikidata document is https://www.wikidata.org/wiki/Q2223139

Appendix F

CD Contents

The following appendix lists directories on the enclosed CD. Directories to the maximum level of 2 are listed.

```
doc/                              documentation folder
        dp assignment/            diploma thesis assignment
        dp report/                diploma thesis source files
        implementation/           implementation documentation
        literature/               categorized literature
        notes/                    various documented implementation topics
        shouts/                   illustrative problems with solutions
        sp report/                preceding software project report
        visualizations/           used plots and other illustrative material
DP.pdf                            diploma thesis in the pdf format
fabfile/                          configuration for the fab deployment tool
matlab/                           source files for the matlab prototyping
python/                           source files for the python implementation
        doc/                      scripts for illustrative material generation
        scenarios/                modules performing a single processing step
        scripts/                  single purpose scripts
        src/                      modules and classes forming the core
        test/                     unit and integration tests
source_files/                     other source files such as SQL schema
subprojects/                      source files for smaller isolated projects
        databases/                tools for utilizing 3−rd party databases
        osm/                      tools for parsing OpenStreet Map dumps
```