



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IMPROVING WEB MULTIMEDIA INFORMATION RETRIEVAL
USING SOCIAL DATA

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN CIENCIAS, MENCIÓN COMPUTACIÓN

TERESA JACQUELINE BRACAMONTE NOLE

PROFESOR GUÍA:
BÁRBARA POBLETE LABRA

MIEMBROS DE LA COMISIÓN:
BENJAMIN BUSTOS CÁRDENAS
AIDAN HOGAN
VANESSA MURDOCK

Este trabajo ha sido parcialmente financiado por Fondef D09I-1185,
CONICYT-PCHA/Doctorado Nacional/2013-63130260, Apoyo a estadías corta de la
Escuela de Postgrado de la U. de Chile, y el Núcleo Milenio CIWS.

SANTIAGO DE CHILE
2018

Resumen

Buscar contenido multimedia es una de las tareas más comunes que los usuarios realizan en la Web. Actualmente, los motores de búsqueda en la Web han mejorado la precisión de sus búsquedas de contenido multimedia y ahora brindan una mejor experiencia de usuarios. Sin embargo, estos motores aún no logran obtener resultados precisos para consultas que no son comunes, y consultas que se refieren a conceptos abstractos. En ambos escenarios, la razón principal es la falta de información preliminar.

Esta tesis se enfoca en mejorar la recuperación de información multimedia en la Web usando datos generados a partir de la interacción entre usuarios y recursos multimedia. Para eso, se propone mejorar la recuperación de información multimedia desde dos perspectivas: (1) extrayendo conceptos relevantes a los recursos multimedia, y (2) mejorando las descripciones multimedia con datos generados por el usuario. En ambos casos, proponemos sistemas que funcionan independientemente del tipo de multimedia, y del idioma de los datos de entrada.

En cuanto a la identificación de conceptos relacionados a objetos multimedia, desarrollamos un sistema que va desde los resultados de búsqueda específicos de la consulta hasta los conceptos detectados para dicha consulta. Nuestro enfoque demuestra que podemos aprovechar la vista parcial de una gran colección de documentos multimedia para detectar conceptos relevantes para una consulta determinada. Además, diseñamos una evaluación basada en usuarios que demuestra que nuestro algoritmo de detección de conceptos es más sólido que otros enfoques similares basados en detección de comunidades.

Para mejorar la descripción multimedia, desarrollamos un sistema que combina contenido audio-visual de documentos multimedia con información de su contexto para mejorar y generar nuevas anotaciones para los documentos multimedia. Específicamente, extraemos datos de clicks de los registros de consultas y usamos las consultas como sustitutos para las anotaciones manuales. Tras una primera inspección, demostramos que las consultas proporcionan una descripción concisa de los documentos multimedia.

El objetivo principal de esta tesis es demostrar la relevancia del contexto asociado a documentos multimedia para mejorar el proceso de recuperación de documentos multimedia en la Web. Además, mostramos que los grafos proporcionan una forma natural de modelar problemas multimedia.

Abstract

Retrieving multimedia content is one of the most common daily tasks users perform while surfing on the Web. Current Web multimedia search engines have increased the accuracy of search results, and now provide a more user-friendly experience. Nevertheless, some engines still fail at getting accurate query results that are not so popular, and queries that carry abstract concepts. In both scenarios, the main reason is the lack of background information.

This thesis addresses multimedia information retrieval by using data generated from the interaction between users and multimedia resources. We propose to enhance multimedia information retrieval from two perspectives: (1) extracting relevant concepts from multimedia resources, and (2) improving multimedia descriptions with implicit user-generated data (click-through data). In both cases, we develop frameworks that work independently from the multimedia type, and from the language of the input data.

Regarding the identification of concepts related to multimedia resources, we develop a framework that goes from query-specific search results, to the concepts detected for such a query. Our framework shows that we can leverage a partial view of a large multimedia collection to detect concepts relevant to a given query. Furthermore, we design a user-based evaluation that demonstrates that our concept detection algorithm is more robust than similar approaches based on community detection.

To improve multimedia description, we develop a framework that combines multimedia content with context data in order to enhance multimedia annotations. Specifically, we mine click-through data from query logs, and use queries as surrogates for explicit manual annotations. Upon first inspection, we show that queries provide a concise description of multimedia documents. In addition, we propose a classification for queries and multimedia (e.g., images), based on the semantics they carry, and their interaction with multimedia content in terms of clicks.

Throughout this work, we aim to demonstrate the relevance of multimedia context data when attempting to enhance the process of retrieving multimedia documents on the Web. We show that graph structures provide a natural way to model multimedia problems.

*To women in STEM, keep raising your voice
for a brighter and more equitable environment
to the girls that come next.*

Acknowledgements

It has been a long way since the day I began the doctorate program at the Computer Science Department in the University of Chile. Fortunately, I have always been blessed by the company of good friends and family who lent me support at the times I felt lost, and who celebrated with me those little triumphs of life.

First of all, I thank Barbara Poblete, my advisor, for her guidance during the process of becoming a doctor. Her patience and constant support motivated me to go through this process from beginning to end. I will always treasure all her advice. I would also like to thank the members of my dissertation committee: Benjamin Bustos, Aidan Hogan, and Vanessa Murdock, for generously offering their time, support, and guidance throughout the reviewing process of this document.

I would also like to thank the DCC community; professors, administrative staff and colleagues. In particular, I would like to express my gratitude to Prof. Eric Tanter and Prof. Gonzalo Navarro, for all the priceless support I received; and to Sara Quiñones, Angelica Aguirre and Sandra Gaez, who led me through the sea of bureaucracy.

I would also like to thank my family; my mom, Violeta; my dad, Fernando; and my siblings, Milagros and Juan Diego. Their love, despite the distance, has always been present during all these years. My deepest gratitude goes to my mom, for coming to my side during the times I needed to feel like a daughter. I would also like to thank my sister, Mili, who is part of my family in Santiago, for helping Daniel and me to raise Marco and Andrea. Also, I would like to thank my extended family: Antonio, Carmen, and Diana for all their support and for making me feel part of their family since I met them. We all miss Don Antonio.

I also extend my gratitude to all of the DCC Peruvian Colony, for making me feel part of a bigger family. Specially, I express my gratitude to my friends and former-neighbors Analí Alfaro and Ivan Sipirán, for giving Daniel and me a family-like welcome in Santiago. To Carlos Bedregal and Violeta Chang, my two memorable officemates, for sharing their insights about academy and for being such a nice daily companions. Our paths might be different, but all of you are always present in my memories.

Last but by no means least, I thank Daniel, my partner in crime (I mean life), for his unconditional love and support. Words are not enough to thank you Daniel for being such a understanding husband. I am aware that I have always been like a tornado in your life, shaking things upside down for good or ill, though you have always been at my side in both good and bad times, in sickness and in health, as we promised.

Contents

1	Introduction	1
1.1	Thesis contributions	4
1.2	Publications	5
1.3	Thesis outline	6
2	Preliminaries	7
2.1	Relevance in IR Systems	7
2.2	The Triad of Search: User, Query and (Multimedia) Document	8
2.2.1	The User	8
2.2.2	The Query	10
2.2.3	The Multimedia Document	12
2.3	Expressing an Information Need	14
2.3.1	The Intention Gap	14
2.3.2	Query disambiguation	15
2.4	Retrieving Multimedia Documents on the Web	17
2.4.1	The Semantic Gap	18
2.4.2	Multimedia Context Data	19
2.4.3	Multimedia Retrieval System	22
2.5	Assessing Search Results	22
2.5.1	Presenting the Results	23
2.5.2	User Interaction	25
2.5.3	Evaluation	27
2.6	Summary	35
3	Related Work	36
3.1	Topic Discovery on Multimedia	36
3.1.1	Multimedia search results clustering	36
3.1.2	Community detection for topic discovery	38
3.2	Automatic Multimedia Tagging	39
3.2.1	Analyzing context data to tag multimedia	40
3.2.2	Combining content and context data to tag multimedia	40
3.3	Human Evaluation in Multimedia-related Scenarios	41
3.4	Summary	42
4	Topic Identification in Multimedia Search Results	44
4.1	Framework for Detecting Multimedia-related Concepts	45

4.1.1	Overview	45
4.1.2	Tag Graph Construction	47
4.1.3	Community Detection Algorithm	48
4.2	Adaptive Island Cuts for Topic Detection	49
4.2.1	Islands	49
4.2.2	Island hierarchy based on edges	50
4.2.3	Island hierarchy based on vertices	52
4.2.4	Adaptive island cuts	55
4.3	Experiments	59
4.3.1	Dataset	59
4.3.2	Algorithms	60
4.3.3	User study design	62
4.3.4	Inter-assessors agreement	64
4.3.5	Majority-voting Precision	67
4.3.6	Data-driven Recall	69
4.3.7	Ontology-based Cohesion	71
4.3.8	Correlation between User Opinion and Tag Graph	74
4.3.9	Discussion	75
4.4	Conclusions	76
5	Automatic Tagging of Multimedia Resources	77
5.1	Framework for Automatic Multimedia Tagging	78
5.1.1	Overview	78
5.1.2	Visual-Semantic graph	79
5.2	Characterizing Visual Information Needs	81
5.3	Propagating Tags on the Visual-Semantic graph	82
5.3.1	Stop-images	82
5.3.2	Weighting schema and pruning indicators	83
5.3.3	Bounded propagation of tags	84
5.4	Experiments	86
5.4.1	Dataset	86
5.4.2	Sample selection	88
5.4.3	Automatic Tagging Exploratory Analysis	89
5.5	Conclusions	89
6	Conclusions and Future Work	91
6.1	Main Results	91
6.2	Limitations	94
6.3	Future Work	95
	Bibliography	97
A	Benford's Law and Zipf's Law	109
A.1	Benford's Law	109
A.2	Zipf's Law	110

List of Figures

1.1	Search results for one-term queries	2
1.2	Search results for multi-term queries	2
2.1	Triad of Search	9
2.2	Image search results layout	10
2.3	Types of query for image search	12
2.4	Query disambiguation on search engines	16
2.5	Semantic gap	19
2.6	Image context data	21
2.7	Multimedia retrieval system architecture	23
2.8	Search results for image searches	24
2.9	Search results for video searches	24
2.10	Search results for audio searches	25
4.1	Framework for topic detection	45
4.2	Image search results for query “tiger”	46
4.3	Tag co-occurrence graph for query “tiger”	47
4.4	Edge island hierarchy	51
4.5	Vertex island hierarchy	54
4.6	Functions to determine density coefficients threshold	56
4.7	Functions to determine number of edges threshold	57
4.8	Communities detected using edge island hierarchy	58
4.9	Tag-graph partitioned using edge island hierarchy	58
4.10	Humman Intelligence Task (HIT) to assess topic quality	63
4.11	Rate of terms known with respect to rate of assessors	65
4.12	Inter-assessor agreement distribution using Krippendorf’s α	66
4.13	Majority-voting precision over sampled communities	68
4.14	Majority-voting precision over sampled communities with fair agreement	69
4.15	Relative Recall over communities	71
4.16	WordNet Cohesion vs. WordNet Coverage	73
4.17	Correlation between user study and automatic assessment	74
5.1	Framework for automatic tagging	78
5.2	Visual-Semantic graph	80
5.3	Stop-images on the Visual-Semantic graph	82
5.4	Re-weight of edges in the Visual-Semantic graph	83
5.5	Cross-point weight curves for EHD	85

5.6	Cross-point weight curves for HSV	86
5.7	Cross-point weight curves for OMD	87
5.8	Tag propagation sequence	88

List of Tables

2.1	Efficiency metrics	31
2.2	Datasets for benchmarking - Part 1	33
2.3	Datasets for benchmarking - Part 2	34
4.1	Social20 dataset summary	60
4.2	Statistics of communities detected using experimental setup	61
4.3	Community detection algorithms runtimes	62
4.4	Statistics of communities sampled for user study	64
4.5	List of unknown tags	65
4.6	Rate of communities for which the majority of assessors found at least two related terms	67
4.7	WordNet coverage and cohesion on AIC-* communities	72
4.8	Correlation between users' opinions and Tag Graph	75
5.1	Audio-visual descriptors for the Visual Similarity graph	80
5.2	Number of connected components in the Visual-Semantic graph	88
5.3	Precision comparison	89

Chapter 1

Introduction

Information Retrieval on the Web is not only limited to text documents. The Web provides the perfect platform to share and gather multimedia information from around the world. In this context, Web search engines must represent, index and search within ever-growing collections of electronic text and other media, such as images, audio and video. To fulfill user information needs, commercial search engines apply the same text-based structure to represent text documents and multimedia objects. Users easily adopted this functionality extension as the *default* mechanism for multimedia searches. Hence, query-by-keyword search is still the main paradigm to represent multimedia information requests on the Web.

Though the multimedia documents online search function has been around for the past decade, search engines do not always return accurate results for multimedia searches. Based on our experience as searchers, Web search engines can retrieve multimedia content focused on a single object or idea effectively. For instance, if we introduce “sea” and “lion” in a Web search engine, we get accurate results in most commercial Web search engines as shown in Figure 1.1. However, when we introduce queries with more complex semantics, their performance decreases. For example, if we introduce the query “lion swimming in the sea”, we get results that correspond to the query “sea lion swimming”. In Figure 1.2 we show the results returned for each query. Thus, the more complex the ideas in our queries, the more noise in the results list. Moreover, we notice that multimedia resources with completely different meaning might be associated with similar descriptions, and therefore might be indexed with similar terms. Novel techniques based on deep-learning have been proven to effectively index large collections of multimedia documents by exploiting multimedia content and context. Nevertheless, most of these techniques focus on single-term indexing (which is similar to indexing images with the class they get from a classifier), while multi-term indexing has yet to be further explored.

The loss of accuracy for queries that represent complex ideas can be explained as a result of the *semantic gap* [103]. The semantic gap is defined as the difference between the computational representation of multimedia objects and what users understand from its content. The main issue on having different representations for the same object is that these representations are not equivalent. For instance, we can represent multimedia objects based on their color, though this description is not compatible with another representation based on

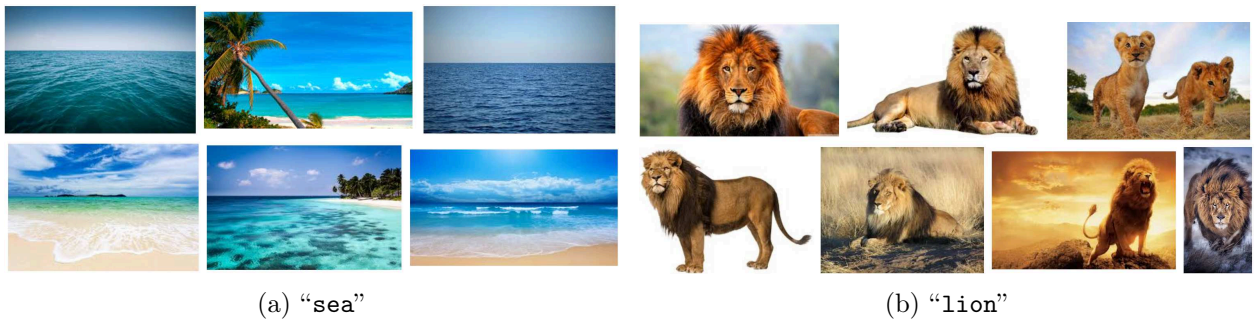


Figure 1.1: Images in the search result listing for queries “sea” and “lion”.

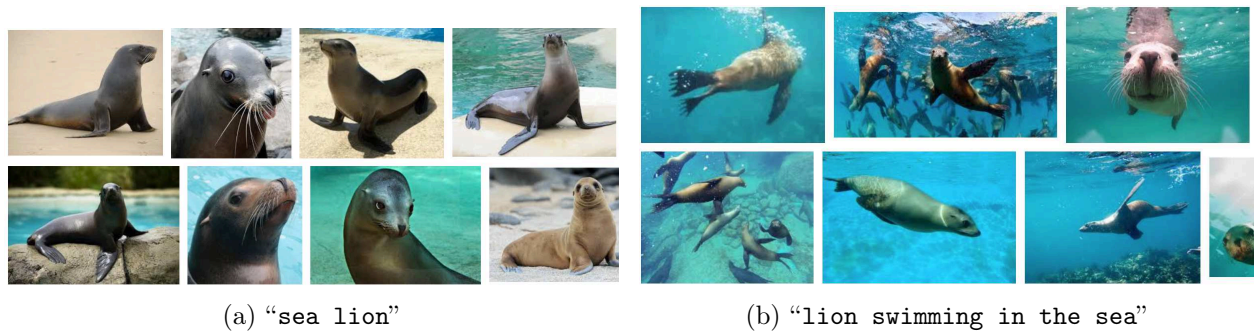


Figure 1.2: Images in the search result listing for queries “sea lion” and “lion swimming in the sea”.

texture. We can also describe multimedia objects using natural language with different terms and levels of detail. Thus, the main challenge in Web Multimedia Information Retrieval is how to establish a meaningful relationship between context and content information of multimedia objects taking into consideration that this relationship must scale to the diversity of context we find on the Web. We can therefore obtain accurate results for user requests regardless of the query complexity.

Regarding the current state of multimedia information retrieval on the Web, this thesis addresses the following research question:

How to extract relevant concepts, and enhance the description of multimedia documents published on the Web?

Our hypothesis is that by aggregating the information users generate on the Web as a result of their interaction, both with multimedia publishing platforms and multimedia search engines, we can enrich the context of multimedia objects. Specifically, we center our efforts on two (2) main goals, which we consider to be the basis for improving multimedia retrieval on the Web:

- **Goal 1: Identify concepts in multimedia search results**
- **Goal 2: Effectively annotate multimedia documents using queries**

Besides our main goals, we also expand our efforts to properly assess the results obtained for Goal 1: *Identify concepts on multimedia search results*:

- **Goal 1.A: Design and deploy a human-centered evaluation to assess algorithms for detecting groups of related annotations.**

It is necessary to clarify that multimedia documents may refer to single mode documents (such as images, video, speech, video and so on), or multimodal documents (i.e., the combination of two or more modes). Despite that in this thesis we aim to generalize our frameworks to any type of multimedia document, our use cases are primarily concerned with images. We address Goal 1 (*identify concepts in multimedia search results*) by focusing on discovering concepts that arise from a set of images returned for a given query using community detection techniques over the set of annotations that describe such images. To accomplish this goal, we analyze and exploit online social community properties of folksonomies [113] to discover semantically meaningful groups of related terms (clusters), based on the way users employ them for tagging images distributed through sharing platforms. We assume that there exists a large collection of manually annotated images, and that such collection is freely available to use (e.g., Flickr¹). In particular we:

1. Design a general framework to discover concepts associated with a set of images.
2. Explore state-of-the-art approaches based on community detection for discovering query-related concepts.
3. Propose a new community detection method based on the paradigm of graph islands.
4. Describe, quantitatively and qualitatively, the behavior and performance of community detection methods when applied to discovering image-related concepts.

Our second goal is to *annotate multimedia documents using queries* by focusing on automatically tagging images by propagating annotations, specifically queries, between images with similar audio-visual content. To accomplish this, we mine user-generated content (UGC) from query-logs to obtain tags. Antonellis et al. [2] state that *logsonomies* [60] (query-log based ontologies) are a rich source of high quality tags. We assume that we have access to a large collection of multimedia documents, and to the set of queries for which they have been clicked. We are aware that although individual clicks do not indicate relevance, when these clicks are aggregated over many users, they do indicate a relationship between a query and a document. Specifically, we:

1. Design a general framework to automatically annotate images based on audio-visual similarity.
2. Explore state-of-the art methods for automatic multimedia annotation.
3. Propose a new automatic annotation method based on a graph structure that combines audio-visual and semantic data (e.g., visual-semantic graph).
4. Describe, quantitatively and qualitatively, the behavior and performance of automatic annotation methods.

¹Flickr is a hosting service for images and videos available at <https://www.flickr.com> (visited 2018/06/01)

Most of the existing work in the area of multimedia retrieval on the Web is performed over hand-crafted datasets. There is a notorious lack of ground-truths to automatically assess the performance of tasks related to multimedia searches other than ranking. Therefore, we believe that designing and deploying well-documented evaluation methodologies is an important step on the path to repeatable and reproducible research. For Goal 1.A (*design & deploy human-centered evaluation to assess algorithms for detecting groups of related annotations*) we focus on designing and deploying user-based evaluations that allow measuring the performance of different approaches to assess the framework proposed for Goal 1. Defining short and focused *Human Intelligence Tasks* (HIT's) is a priority, regardless of whether a user-based evaluation is deployed in a controlled environment such as a user study, or on a large scale, such as a crowd-sourced evaluation.

In particular, we:

1. Propose a methodology to sample equivalent subsets of data to be assessed by humans for *Goal 1*.
2. Design a Human Intelligence Task to measure the effectiveness of using community detection approaches to find semantically relevant sets of terms.

1.1 Thesis contributions

The main contributions of this thesis focus on three main aspects: algorithmic, theoretical, and empirical. We describe them next.

Algorithmic contributions

- A1: A **community detection method** based on island-cuts, called Adaptive Island Cuts (AIC-*), that returns sets of semantically relevant terms that represent concepts associated with multimedia search results. (c.f. Section 4.2)
- A2: An **approach for sampling equivalent subsets of elements** from different communities created by distinct community detection approaches. (c.f. Section 4.3)
- A3: An **automatic tagging method** based on the propagation of information over a graph structure based on multimedia content similarity. (c.f. Section 5.3)

Theoretical contributions

- T1: A **general framework to detect concepts associated to multimedia-related queries**. (c.f. Section 4.1)
- T2: A **general framework for automatically tagging multimedia resources** based on content and context similarity. (c.f. Section 5.1)

Empirical contributions

- E1: An **experimental framework (inspired on IR-metrics) for the assessment of sets of related terms** by collecting multiple user opinions. (c.f. Section 4.3)
- E2: An **empirical evaluation of community detection approaches** for detecting concepts represented as sets of related terms. (c.f. Section 4.3)
- E3: An **empirical evaluation of effectiveness of queries for describing multimedia resources**. (c.f. Section 5.4)

1.2 Publications

1. **Bracamonte, T.**, Poblete, B. “Automatic image tagging through information propagation in a query log based graph structure” (2011). MM’11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops, pp. 1201-1204.
This paper presents our algorithm for automating tagging based on tag propagation over a TagGraph (A3). It also shows the results of our preliminary assessment (E3).
2. **Bracamonte, T.** “Multimedia information retrieval on the social web”(2013). WWW’13 Companion - Proceedings of the 22nd International Conference on World Wide Web, pp. 349-353.
The article contains our initial ideas on how to address this thesis research problem. The paper is mainly oriented to describe the early stages for our theoretical contributions on concept detection (T1), and automatic tagging (T2).
3. **Bracamonte, T.**, Hogan, A., Poblete, B. “Applying community detection methods to cluster tags in multimedia search results” (2016). ISM’16 - Proceedings of the 2016 IEEE International Symposium on Multimedia, pp. 467-474.
This paper presents our study on concept discovery on multimedia annotations. The article introduces our framework for concept detection (T1), and the algorithm we propose for the detection of communities of tags based on island cuts (A1). Also, on the experimental side, the article includes the algorithm to select comparable sets of annotations from different algorithms (A2); and our experimental setup for assessing such sets (E1 – E2).
4. **Bracamonte, T.**, Bustos, B., Poblete, B., Shreck, T., “Extracting semantic knowledge from Web context for multimedia IR: A taxonomy, survey and challenges”. Multimedia Tools and Applications (2017). <https://doi.org/10.1007/s11042-017-4997-y>
This article summarizes and organizes State-of-the-Art work related to this thesis. We also introduce a taxonomy that allows classifying multimedia IR research works based on the type of context data employed to get insights.

1.3 Thesis outline

This thesis is organized as follows:

Chapter 2 describes the preliminary concepts and terminology employed in our research.

We present the Triad of Search: User, Query and Document. We contextualize the triad entities under the scenario of Web Multimedia IR. We describe the main types of users, queries and multimedia documents. We then discuss the main challenges that arise from the interaction between these entities, such as the intention gap (user-query), the semantic gap (query-document), and an assessment mechanism with standardized datasets that include the user in the loop (user-document).

Chapter 3 describes the related work and state-of-the-art approaches for the main topics addressed in this thesis: concept discovery, automatic annotations and user-based evaluations. We orient our literature review only to works with a long term impact in their respective areas. We point out works with extended state-of-the-art literature reviews and contextualize our research under the current state-of-the-art.

Chapter 4 presents our proposal for detecting concepts associated to search results. We present the general framework, as well as our proposed algorithm for concept detection based on island cuts. We describe every stage of the framework, presenting the Tag Graph structure we employed as a building block for our framework. We also include a detailed description of the proposed concept detection algorithm by explaining the concept of “islands”, and the two variations we propose on our base algorithm. We describe in detail the evaluation methodology employed and the results obtained from our assessment.

Chapter 5 presents our proposal for automatically tagging multimedia documents using query logs. We present the general framework for the tagging of multimedia documents with user-generated content, as well as an algorithm for the propagation of user-generated content regarding visual similarity. In addition, we present a taxonomy for Visual Information Needs in the context of Web Multimedia IR. We develop this chapter following the use case of tagging images with queries, and describe the empirical assessment of accuracy performed over a small sample.

Chapter 6 summarizes the conclusions of this thesis. We discuss how our research is positioned in the context of emerging technologies such as deep learning, and current large multimedia datasets such as ImageNet and others released by commercial Web search engines. Additionally, we present the limitations of this work, as well as the direction of future research.

Chapter 2

Preliminaries

In this chapter we provide an overview on the area of Web Multimedia Retrieval. First, we introduce the concept of relevance. Then, we present the entities involved in the problem of retrieving relevant information, which we call the *Triad of Search*. Finally, we discuss concepts related to each of those entities, as well as some challenging problems related to them.

2.1 Relevance in IR Systems

Before we describe the main characteristics of each entity in the so-called *Triad of Search*, it is necessary to understand the concept of *relevance*. Saracevic [94] introduces the concept of relevance as:

“Nobody has to explain to users of IR systems what relevance is, even if they struggle (sometimes in vain) to find relevant stuff. People understand relevance intuitively.”

In a following publication, Saracevic [95] identifies several manifestations (or types) of relevance:

- System or algorithmic relevance: The relation between a query and the objects retrieved by a given algorithm. This manifestation of relevance assumes that the intent is to retrieve a set of objects that the system inferred as relevant for a given query.
- Topical or subject relevance: The relation between the topic expressed in a query, and the topic covered by the retrieved objects. Topicality is inferred based on aboutness.
- Cognitive relevance or pertinence: The relation between the cognitive state of the knowledge of a user, and the retrieved objects. Informativeness, novelty, and information quality are some criteria employed to infer cognitive relevance.
- Situational relevance or utility: The relation between the situation, task, or problem at hand, and the retrieved objects. Usefulness, appropriateness of information, and reduction of uncertainty, are some criteria to infer situational relevance.

- Affective relevance: The relation between the intent and motivations of a user, and the retrieved objects. Satisfaction, and sense of accomplishment, are criteria for inferring motivational relevance.

Each manifestation of relevance focuses on a different side of the timeless concept of relevance. The factors that affect the manifestations of relevance are not the same for all. Even though there is no general agreement on the factors that define the degree of relevance of each manifestation, Saracevic [96] states that users make relevance inferences based on the following basis:

- Content: Is the topic of the retrieved object related to my information need?
- Object: Is the retrieved object stored in an accessible standard format?
- Validity: Is the content of the retrieved object verifiable?
- Use or situational match: Is the retrieved object useful to solve my information need?
- Cognitive match: Is the retrieved object easy to understand, or does it require additional mental effort?
- Affective match: How do I feel about the content on the retrieved object?
- Belief match: Is personal credence given to information? Do I feel confident about the publisher of the retrieved object?

In the following Sections and Chapters, when we mention the term “relevance” we will be referring to “*topical relevance*”, unless explicitly stated.

2.2 The Triad of Search: User, Query and (Multimedia) Document

In Figure 2.1 we illustrate the main entities behind any Information Retrieval (IR) system: the user, the query, and the document (multimedia document, for our specific case). We believe that the interaction between these entities is key to understand the main process performed by any search engine, and to identify the challenges to enhance accuracy in current search engines.

2.2.1 The User

The user is the entity that has an information need related to multimedia content. Datta et al. [28] define three types of users based on search intent:

- **Searcher:** The user has a clear goal and knows how to represent his information need in a concise query. The user has knowledge on how to operate a multimedia retrieval system and employs advanced search features, if necessary. The user is also able to reformulate the initial query to effectively lead to the expected results in a short time session.

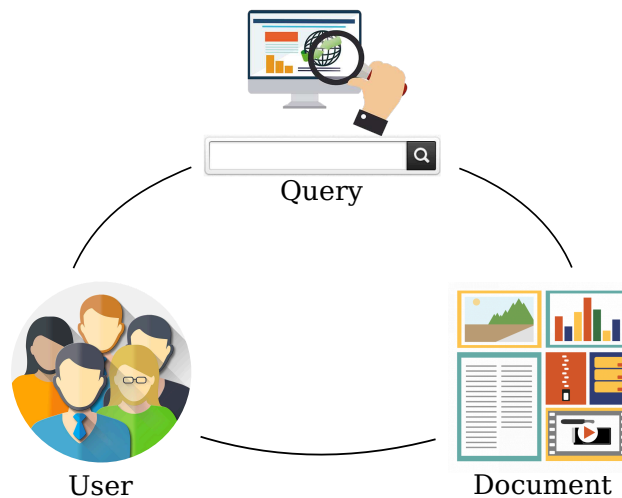


Figure 2.1: Triad of Search: User, Query and Multimedia Document.

- **Surfer:** The user is exploring a multimedia repository, such as the Web, to get an overview of the existing resources; and then come up with more specific queries once an overview has been achieved. Users in this category tend to start with a vague and broad-topic query and then, based on the search results and query suggestions provided by the search engine, narrow the query to get more accurate results. The search sessions of surfers tend to last longer than those of searchers.
- **Browser:** The user goal is less clear. Queries from this kind of user encompass multiple broad (and even unrelated) topics. Browsers do not have a clear idea of what they are looking for, and therefore the relevance value assigned to resources in the search results is highly subjective.

Web search engines were designed to satisfy users in the role of searchers. Most commercial Web search engines employ user relevance feedback to enhance search results for future searches. Relevance feedback is a reliable source of information as it derives from searchers or surfers. Furthermore, we noticed that most Web search engines were not initially designed to support browser behavior. However, lately some search engines have begun to include some additional information in their multimedia-oriented searches layout, which allows expert and non-expert users to explore search results in a friendly and efficient way. For instance, Flickr and Instagram are multimedia sharing platforms that allow users to browse their repositories without submitting an initial query.

Example 2.1. Let us consider three people: Bob, Sue and Sean. Bob is planning a long vacation in Chile, starting at Santiago. Sue is going to visit Santiago, Chile for a weekend, and she is interested on visiting the most iconic places. Sean is preparing a report about Chile for a school assignment. The three of them submit the same query: “santiago chile tourist attractions” to get information, despite the fact that they have different needs. Figure 2.2 shows the search results in a search engine that also provides a Navigation section.

Bob goes over the initial search results, and takes advantage of topics listed on the *Navigation section* to browse content from other related searches. He changes between topics whenever



Figure 2.2: Google image search results for query “santiago chile tourist attractions”. This layout is divided into two main sections: Navigation Section and Search results section.

he ends up looking at tourist attractions that are not located in Santiago. In this case, Bob depicts the behavior of a *browser*.

On the other hand, Sue carefully picks some suggestions at the *Navigation section* and she clicks on some images in the *Search results section*. Sue takes advantage of search engine suggestions to tune the scope of her query. She represents the behavior of a *surfer*.

Finally, Sean is more interested on images from the *Search results section*. He clicks on some images that he considers attractive for his report, without diving on additional suggestions from the search engine. Sean depicts the behavior of a *searcher*.

Given that the context and motivations of Bob, Sue and Sean are different, the way they interact with the search engine also varies.

2.2.2 The Query

We consider the query to be the representation of an information need. The query might have the same format of the multimedia objects included in the archive, or be expressed as text. We generalize the classification presented by Datta et al. [28].

- **Text-based queries:** Most Web search engines index multimedia content using textual descriptions obtained from the Web sites in which the content was uploaded, or from other related resources. Hence, representing information needs using text seems to be the most common way to retrieve multimedia documents. Based on the complexity with which the information need is expressed, we find two ways to formulate queries:
 - *Query by keyword:* Users provide a set of terms that describes the content or context of the multimedia objects they are looking for. The query-by-keyword

paradigm is the *de facto* paradigm of current commercial Web search engines, such as Google¹ and Bing².

- *Free-text query*: Users can express their information needs using more complex phrases (e.g., long sentences, and questions). Search engines that support this type of query require algorithms to process queries on natural language, because they need to correctly interpret such complex queries expressed as text. Kngine³ is a semantic search engine that understands complex queries, and answers questions in response to user information needs.
- **Content-based queries**: Many multimedia resources on the Web are not associated with a textual description. This makes the content difficult to both index and make it available to Web users. Thus, content-based queries provide a mechanism that allows users to search multimedia content when textual descriptions are unavailable. In this scenario, we find that there are two main types of queries based on the amount of information they contain:

Queries for near-duplicate search: This type of query is useful when the intention of the user is to find multimedia documents with similar audio-visual content.

- *Query by example*: Users give an example of a multimedia object similar to the one they are looking for. This type of query is suitable for users that have a clear idea of what they are looking for and already have a sample. For instance, Shazam⁴ is a mobile application that allows users to search for music using only fragments of songs that are being played on their direct vicinity.
- *Query by musical notation*: This type of query applies to audio searches, and is a more professional way of expressing an audio-related information needs. In this type of query, the user describes a request as a set of musical notes. Fournier et al. [35] model music as temporal sequences of musical events represented by their notations.

Queries by approximate example: This type of query allows users to define their searches in a more flexible fashion. In addition, they are useful for users that are interested in specific content, but do not have access to something similar to the expected multimedia resource.

- *Query by sketch*: Users provide a sketch of the multimedia object they are looking for. The sketch is then processed to obtain its signature. The retrieval process uses this signature to get similar multimedia objects. MindFinder⁵ is a technology developed by Microsoft that supports sketch-based searches.
- *Query by humming*: This type of query applies to audio searches, and it requires that the user hums (or whistles or sings) a melodic query. The search engine then looks for matching melodies in its repository [121].

¹<http://www.google.com/>

²<http://www.bing.com/>

³<http://www.kngine.com/>

⁴<https://www.shazam.com/>

⁵<https://www.microsoft.com/en-us/research/project/mindfinder-finding-images-by-sketching/>

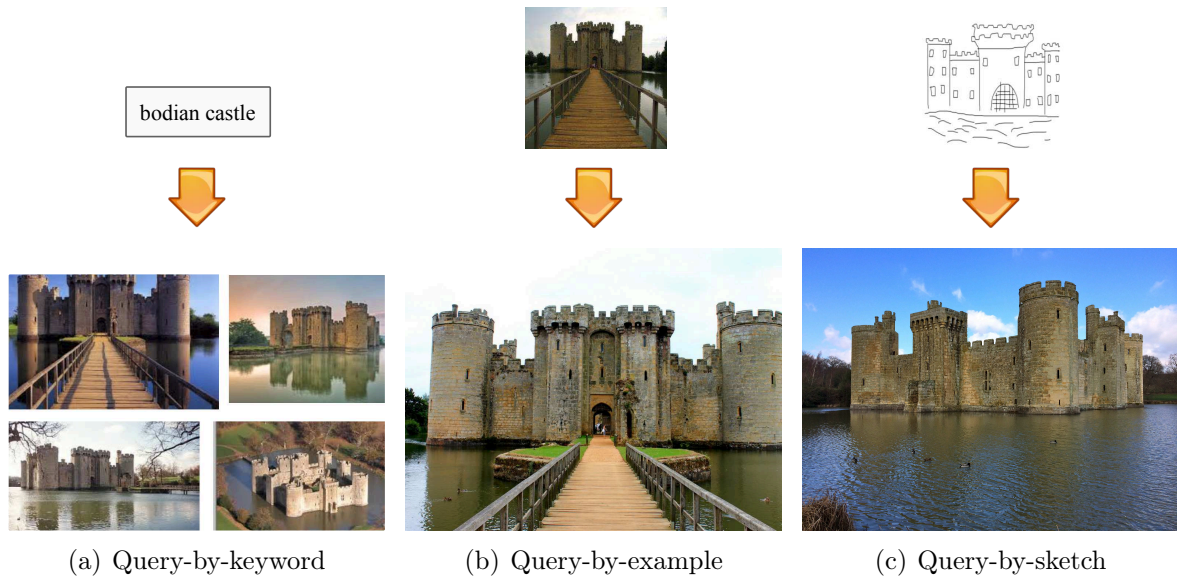


Figure 2.3: Image search results for “bodiam castle” using different types of query. All three types can return more than one result. For simplicity we depict only one output image for query-by-example, and query-by-sketch types.

- **Multimodal queries:** This type of query aims to combine two or more modalities of multimedia content. Currently, none of the Web commercial search engines support multimodal queries. Nevertheless, there are some field-specific search engines that support multimodal queries. For example, the NovaMedSearch [77] system supports queries that combine keywords and medical images.

Example 2.2. Let us imagine a user, who is looking for pictures of the “Bodiam castle”, and who has access to a multimedia search engine that allows searching with different types of query. She could use a *query-by-keyword* and request for images associated to the text “bodiam castle”. She could also use a picture of the place, and search for the same image in higher resolution using a *query-by-example*. If she did not have a picture to use as example, she could use a drawing that represents the main features of the sought object, and retrieve similar pictures using a *query-by-sketch*. We illustrate the different types of query mentioned in Figure 2.3.

2.2.3 The Multimedia Document

The multimedia document may refer to a single mode or multimodal document published on the Web, such as an annotated image published on Instagram, or a tweet linking an image or video. Given that documents under this scope are not like plain text documents, they comprise two types of content: (1) the multimedia object which comprises the non-textual data contained in the document, and (2) the metadata which comprises the additional non-visible data embedded on the document.

- **The multimedia object:** This represents the actual multimedia content, such as an image, video, or audio clip. These modalities might be composed to create more complex multimedia objects.
 - *Image:* Digital images were the first non-text data to be distributed over the Web. Nowadays, online social platforms, as well as mobile devices with cameras, enable users to share pictures from all over the world. Images are usually represented as color value matrices using color spaces (e.g., RGB). Since human vision is tolerant to loss of information, some image formats (e.g., JPEG) allow for lossy compression encoding and thus minimize redundancy in the image representation. Besides the matrix representation of images, there are also vector-oriented representations (e.g., SVG) which encode visual content independently of a given matrix and can scale smoothly to any output resolution.
 - *Video:* Digital videos are composed of a sequence of images, also known as frames. Without compression it typically takes a large amount of data to represent videos based on each frame. Hence, space efficient video representations have been developed, which are widely used in digital video acquisition devices, or for transmission on various media, including the Internet. Similarly to images, video compression standards, such as MPEG4, focus on reducing redundancy and compression without losing too much perceived video quality. The main assumption for compressing videos is that neighboring frames have similar content. Therefore, frames with few differences between one another could be replaced with the pixels of one frame and the motion vector, encoding only the differences between images.
 - *Audio:* Audio data can be distinguished by two main classes of audio objects: music and speech. For example, music is uploaded to different social platforms to share it with other users through the Internet (e.g., podcasts). Most news broadcast channels offer their services on-line. Users therefore have the chance to listen to their favorite news and programs on-line. Despite the widespread use of both types of audio documents, this area is still under development in terms of Web Multimedia IR. Nonetheless, there are notable exceptions such as the technology used in Shazam [116], which consists of capturing small segments of music through the smartphone microphone to search for the full song in a collection with millions of tracks. One of the main difficulties in audio retrieval is that digitized audio signals from speech and music have different properties.
 - *Other types of media:* Text, images, video and audio make up the majority of multimedia data types most often encountered in the Web. However, other data aspects exist that can be identified as multimedia data types. For instance, 3D objects are important data types often used for simulation and visualization, particularly for CAD processing or scientific applications. Retrieval in 3D object data has been studied for some years now [19, 107], due to its relevance for many scientific and industrial applications.
- **The Metadata:** This corresponds to text that is embedded in a document and describes the multimedia content. This is a rich source of information, although it is not always available. We can distinguish between automatically and manually generated metadata. The first type includes technical metadata, which can be derived from the acquisition process, such as the file size and resolution of a digital image. Most

digital devices represent metadata on still images and audio using Exif, which is a metadata standard for images initially introduced by the Japan Electronic Industries Development Association (JEITA⁶) in 1998. Metadata may also be generated as part of an automatic media analysis process. For example, automatic video segmentation methods can generate scene or shot indexes, which are a form of technical metadata.

In the second type of annotation (manual), metadata can be created during an interactive process. Annotations may be added by information experts, e.g., in a library context. Manual annotation may also be performed by individual authors, e.g., in the form of HTML markup of a Web page that is edited. Also, collaborative annotation efforts exist in which user communities share annotations or metadata, such as with the bibsonomy⁷ effort for scientific bookmark sharing.

2.3 Expressing an Information Need

Looking for information to satisfy an information need might be harder than expected. Users with little prior experience, or that are trying to accomplish complex tasks, might struggle with long-lasting search sessions with no acceptable results. Kofler et al. [59] consider that a user information need has two dimensions: (1) the “**why**”, which refers to the search task the user is intended to solve, and (2) the “**what**”, which refers to the object being searched. The first dimension, “why”, might be described as the main goal behind the user information need. This dimension is tied to the type of task the user is trying to solve by interacting with the search engine. The second dimension, “what”, might be described as the topic under research. This dimension requires users to provide a clear request (i.e., a query) so that the search engine can “*understand*” the specific topic the user is looking for.

2.3.1 The Intention Gap

Expressing search intent through a query is not straightforward, since keywords might not fully represent the complexity behind an information need. This gap between the users’ search goals and the queries employed to express them is called *The Intention Gap* [28]. The intention gap is related to the “why” dimension of an information need, and is not exclusive to multimedia searches. Broder [18] provides the first query taxonomy based on Web search intention. He defines three categories: (1) Informational, (2) Navigational, and (3) Transactional. Nevertheless, multimedia searches should be treated somewhat differently, since they require a more fine-grained understanding of the intention behind them. Lux et al. [71] refine Broder’s proposal with a taxonomy for user intention in image search. After analyzing query logs and conducting interviews, Lux et al. [71] determine user intention can be classified into four (4) categories:

⁶<http://www.jeita.or.jp/english/>

⁷ <https://www.bibsonomy.org>

- Knowledge Orientation: The user is looking for and wants an image from which to learn something.
- Mental Image: The user has in mind an image with some specific content.
- Navigation: The user has no idea about the image appearance, but knows about its existence.
- Transaction: The user is searching an image for further use.

Similarly, Hanjalic et al. [44] define a taxonomy for video search based on social Web mining and crowdsourcing. They divide user intention in five (5) categories:

- Information: Users are looking for new information or explanations.
- Experience-Learning: Users want to learn a new ability practically by experience.
- Experience-Exposure: Users aim to have a particular experience from a real-life event or entity.
- Affect: Users aim to change their mood or affective state.
- Object: Users want a video for further use.

Regardless of the type of multimedia content or taxonomy employed to organize user intent, relevance feedback is the most common mechanism employed to determine the intent behind a user information need. Search engines aim to refine the retrieval based on iterative feedback; for example, users might be required to explicitly label resources as *relevant* or *irrelevant*. There are plenty of methods that propose elaborate learning functions to automatically enhance search results based on user relevance feedback. Nevertheless, the original idea of asking users for relevance labels has been replaced by more elegant approaches that analyze historical data from search query logs. Several works [15, 46, 52, 93] show that modeling search behavior using query logs allows search engines to predict the user intention, and therefore improve the search results with fewer user interactions. Hence, user modeling is very important when attempting to bridge the intention gap.

2.3.2 Query disambiguation

An accurate formulation of queries is required to accurately describe the “what” dimension associated with a user information need. Most of the time, formulating a query that accurately represents a user information need requires vast background knowledge about the topic under search. The ambiguity associated to queries is not exclusive to polysemic terms. The ambiguity of a query might be tied to the scope of the query and the popularity of the topic among Web users. Queries with a wide scope might need to be reformulated in order to improve the search results. Moreover, niche queries usually do not have enough user feedback, which makes it difficult to determine whether the search results contain relevant or irrelevant documents.

In many approaches, the disambiguation problem is treated as a query refinement problem. This means that the solution enhances the request sent to the search engine, not the results returned by it. We find that there are two main approaches that deal with query disambiguation:

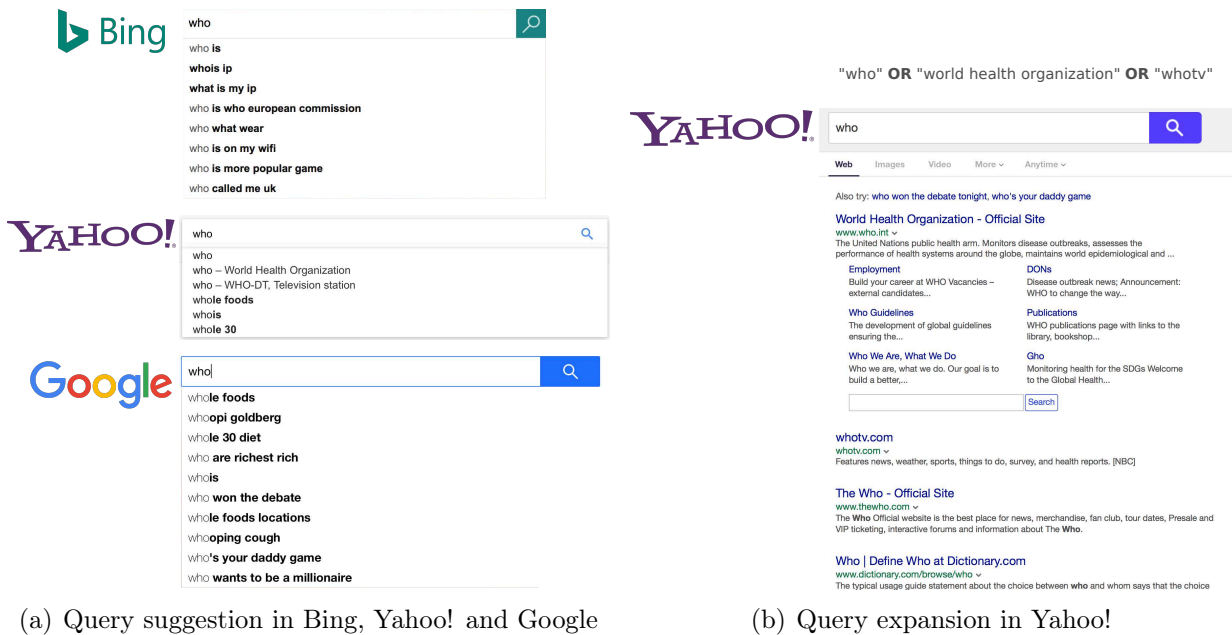


Figure 2.4: Search engines query disambiguation functionality: (a) Query suggestion gives users the possibility to expand their queries, and (b) Query expansion works internally after users submit their queries.

- **Automatic query expansion (AQE):** The mechanisms for expanding queries in multimedia IR systems go beyond determining relevant keywords to expand a query-by-keywords or applying spell checking to increase accuracy in search results. Carpineto and Romano [20] survey current AQE approaches and propose a taxonomy to organize them according to the data source employed to expand the query features.

Regarding multimedia IR systems, it is important to consider that queries might carry information from different modalities (e.g., text, image, audio), and even include context data from the user (such as historical session data, location). Some works trim the query expansion problem by mapping textual and audiovisual features to predefined concepts obtained from an ontology [29] or mined from the multimedia repository [78]. For example, Natsev et al. [78] propose a concept-based model to perform query expansion on multimedia repositories. Similar works [128] expand queries, but analyze only local information relevant for the current query. Other studies on multimedia query expansion focus on the audio-visual expansion of the query. For instance, Chum et al. [26] assume that the input query is part of a bigger image, and hence they expand the area of interest initially selected as the query.

- **Query suggestion (QS):** When translating an information need to a query, some information is usually lost. This loss of information might affect the quality of the search results, and therefore it is necessary for search engines to provide suitable mechanisms that allow users to formulate queries as accurately as possible, in order to obtain relevant results. One of the most common mechanisms to help users is by suggesting additional elements to expand their queries before submitting them to the search engine.

Zha et al. [134] propose a framework for Visual Query Suggestions (VSQ) which complements text-based query suggestions with multimedia resources (such as images) that might help users to quickly identify which suggested query is closer to their search goal. In their framework, they propose presenting suggestions in two ways: (1) in-line query suggestions in the query box, and (2) multimedia snippets next to search results. As a result of their work, they demonstrate that a search engine that allows for VSQ outperforms other search engines in terms of both query suggestion quality and search performance.

To the best of our knowledge, there are no works on multimedia content-based query suggestion in commercial Web search engines.

It is important to remark that while automatic query expansion is transparent for users, the query suggestion process requires users to interact with the system before submitting the query.

Example 2.3. Let us submit the one-term query “who” to different search engines, with no additional information about the user intention. In order to return relevant results, search engines might *suggest* additional terms that lead to reducing the scope of the query, such as “who - World Health Organization”. If none of the suggestions is accepted by the user, search engines likely expand the original query internally in order to get a more elaborate query that stands for a more diverse results. For example, an extended version of the original query might be [“who” OR “world health organization” OR “whotv”]. Nevertheless, the terms added by the search engine are unknown for the user. In Figure 2.4 we show examples of both functionalities. For query suggestion, we show results from three different commercial search engines in Figure 2.4(a); for query expansion we provide a candidate expanded version of the original query based on the search results shown in Figure 2.4(b).

2.4 Retrieving Multimedia Documents on the Web

Recovering multimedia documents on the Web is more complex than retrieving text documents, because the query representing the information need is usually expressed in a domain different from that of the multimedia documents (e.g., employing textual queries for searching videos). Hence, extracting information from other resources apart from the audio-visual content is essential to understanding the context in which the multimedia document is published. Multimedia retrieval systems on the Web are built upon components that deal with the *semantic gap* and the analysis of context data, as well as the user interaction with the search results. In this section, we describe the semantic gap problem and some approaches to reduce it. We then explain the main sources of context data we find on the Web, and briefly describe a generic Multimedia Retrieval System architecture.

2.4.1 The Semantic Gap

Multimedia document semantics cannot be directly mapped to a universally agreed description because people have different opinions and tend not to agree on a consensus. Moreover, the lack of specific semantics associated with audio-visual features makes it difficult for Web search engines to index multimedia objects in the same way they do with text. Smeulders et al. [103] define the semantic gap as : “*the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data has for a user in a given situation*”. In Figure 2.5 we show two different pictures (an apple and a tomato). Although both pictures correspond to different objects, their computational representations carry enough similar information to determine that both elements are the same.

Many proposals address the semantic gap from different perspectives. Some approaches focus on exploiting ontologies and metadata standards to automatically infer knowledge from the relationship between multimedia content and semantic information. Ponnada and Sharda [87] propose a model for a Web search engine centered on multimedia object semantics. Similarly, Straccia [105] proposes a retrieval system which combines an ontology with logic inference over multimedia features. Both approaches intend to establish a proper ontology that reflects the semantic relationship between multimedia objects.

Supervised and unsupervised machine learning techniques have also been applied to bridge the semantic gap. Gao et al. [36] explore the application of co-clustering techniques over a graph that represents the relationship between image features and surrounding text. Similarly, van Leuken et al. [112] apply clustering over visual features of images from query search results to diversify search result listings. In the same way, Gui et al. [43] employ surrounding text and visual features to describe images, and ranks these images based on a linear combination of different representations. Apart from clustering techniques, Chen et al. [24] apply statistical learning to determine meaningful relationships between textual and visual features in vertical searches. Rasiwasia et al. [91] exploit the correlation between visual and textual multidimensional spaces by applying logistic regression to produce a classifier.

An additional way to bridge the semantic gap is by enhancing the annotations of multimedia documents. Wu et al. [124] propose to automatically annotate objects inside images by building a codebook from the concepts detected inside the images of their dataset. Li et al. [64] propose to determine tag relevance using a voting scheme. Tags from similar images represent a vote over the tag in the image under analysis. Li et al. [65] extend their work to include multiple features, where each feature focuses on a different visual characteristic. Aside from image tagging, Turnbull et al. [110] annotate songs based on different objective and subjective aspects related to such songs. In addition, Yao et al. [130] apply 3D convolutional neural networks to represent temporal dynamics, and then detect entities and actions to generate video descriptions.

Recent proposals exploit state-of-the-art machine learning techniques, such as deep learning, in order to reduce the semantic gap. Wan et al. [115] present a comprehensive study on deep learning techniques for content-based image retrieval. Some proposals center on learning other multimedia representations beyond content to apply multimodal retrieval [119] or cross-domain retrieval [120]. Other proposals improve the description of multimedia content,

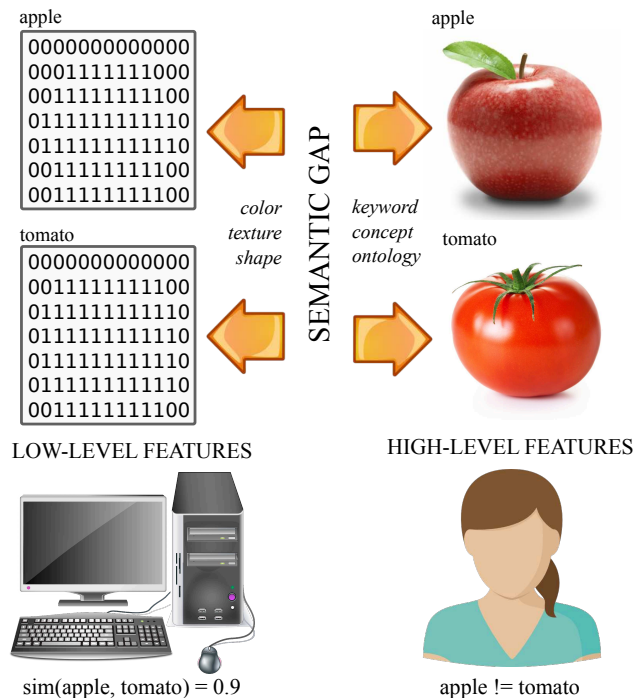


Figure 2.5: Example of semantic gap: *apple* and *tomato* carry different semantics, although they have similar audio-visual representations.

such as generating text descriptions [56], or determining the relevance of manually-assigned annotations [63]. Besides, there are works that exploit Web context data in combination with deep learning [66, 118, 51].

2.4.2 Multimedia Context Data

The data distributed along with multimedia content over the Web offers rich information about the context of said content, and also about how users access and employ it. This context data enhances multimedia understanding [50, 53, 67], and thus reduces the semantic gap. We identify five (5) main resources from which multimedia context data can be mined:

1. **Web structure data:** Documents within the Web are interconnected by hyperlinks, which implicitly indicate a relationship between them. These hyperlinks between documents include *anchor text*, which provides information about the document being pointed to. The directed graph structure built upon hyperlinks (where documents are nodes and hyperlinks are edges) is usually exploited to extract relevance indication for documents [129], and might be useful for multimedia understanding as well. *Linked Data* is an analog structure of hyperlinks, but for the *Semantic Web* [10]. There are some works [45, 48, 74] that describe the benefits of using Linked Data in combination with multimedia content.
2. **Document structure data:** Another type of structural data found on the Web is the structure within a document (e.g., HTML pages). For example, the title of a document

can be considered more relevant than the content of the document, and highlighted words might be more relevant than others in the same sentence or paragraph. Also, the caption or surrounding text of multimedia data embedded in Web documents, is thought of as descriptive of the actual (semantic) content of the multimedia object itself.

3. **Document metadata:** Metadata is the data that is provided manually or automatically in order to enrich the understanding of a certain document (being text, HTML or multimedia). For example, images can often be found stored along with metadata that describe the location where the image was taken (if it is a picture), the camera that was used, and the date it was taken, among others. Additionally, metadata can include human descriptions of the multimedia objects, known as *tags*, which commonly describe objects and concepts represented in the data. These latter types of annotations are usually generated by humans.
4. **Social media data:** In recent years, the concept of the Social Web has substantially broadened, if not changed, the landscape of Web information systems. In the Social Web, user-centric platforms have flourished and users have become editors, publishers, and consumers alike, of the content they generate. On a daily basis, the Social Web is growing with multimedia data generated by users in real time at extremely high rates. Social data includes different types of information, and is very rich in several ways. For example, user preferences for certain content can be extracted from social data. Also, comments generated by users for multimedia content can be potentially exploited to actually understand the semantics of the content. At a higher level, users also form networks in which they become related to each other. This information is also a new factor to incorporate.
5. **Search engine query logs:** Web Multimedia IR systems are constantly exposed to a large number and variety of users that interact with the system. This constant interaction through time, if appropriately processed, may become a rich source of information recorded in the search engine query log. This log keeps track of all the queries issued by users and later on, their clicked results. This information can be mined, for example, to extract implicit relevance feedback information about the best results for certain queries.

The contextual resources listed before describe several aspects of multimedia content. For instance, using Web structure data, we might determine which websites for multimedia content distribution are more popular. By using Social Media data, we could identify how multimedia content spreads inside social circles. Using multimedia sharing platforms, we might get user-generated data from the publisher and further interactions from other users (see Figure 2.6). And finally, by using Search Engine query logs, we might identify which terms applied to index multimedia content are more effective and are employed by a larger number of users.

The screenshot shows a Flickr photo page for a landscape in Austria. The main photo is a wide-angle shot of a valley with a small village and mountains in the background under a cloudy sky. The page layout includes:

- Camera Metadata:** Nikon D90, 18-200mm f/3.5-5.6, f/10.0, 23.8 mm, 1/400, ISO 200, Flash (auto, no disparar).
- Map:** Shows the location in Finkensteintal, Carinthia, Austria.
- Owner Information:** Steve Lamb, Rural Austria, Carinthia, Southern Austria. Includes a 'Seguir' button and aggregated social interaction info.
- Engagement:** 3,408 visitas, 42 favoritos, 1 comentario. Taken on September 27, 2014.
- Groups and Albums:** 'Esta foto está en 37 grupos' and 'Esta foto está en 2 álbumes' with thumbnails for various categories like '200 views at least' and '2500+ views'.
- Tags:** A list of tags including 'Austria', 'Carinthia', 'Finkensteintal', 'stevelamb', 'nikon', 'D90', 'rural', 'landscape', 'vista', 'mountain', 'alps', 'valley', 'farm', 'buildings', 'clouds', 'village', 'cornfield', 'aire libre', 'montaña', 'paisaje', 'campo', 'nube', 'cielo', 'prado', 'outdoor', 'field', 'cloud', 'sky', 'grassland'. Two yellow boxes highlight specific tag groups: one for 'aire libre', 'montaña', 'paisaje', 'campo', 'nube', 'cielo', 'prado' (labeled '(ES)') and another for 'outdoor', 'field', 'cloud', 'sky', 'grassland' (labeled '(EN)').
- Annotations:** A yellow box explains that tags inside a rectangle with a transparent background are automatically assigned and translated to the visitor's language.

Figure 2.6: Context data of images published on Flickr includes information of the publisher, capture device, and annotations, among others. Descriptions for specific context sections is provided inside yellow rectangles.⁹

2.4.3 Multimedia Retrieval System

As the volume of digital repositories on the Web increased, it became more and more difficult to manually classify them into different topics, as well as searching for a specific document. Web information retrieval systems were created to solve this problem in an automatic fashion, allowing users to search by formulating a query that describes the information that they need. In this section we give an overview of the architecture of a Web retrieval system with a focus on the multimedia documents represented in Figure 2.7. We see that there are two main processes: (1) the indexing process, which is performed off-line; and (2) the querying process, which works on-line.

For the indexing process, we notice that the **Multimedia Content Acquisition** component is in charge of crawling the Web, gathering multimedia documents that are stored in a *Multimedia Repository*. Then, the **Feature Extraction & Transformation** component processes those documents. As a result of this process, a set of features that describes the content and/or context of the multimedia documents is sent to the next component. The **Index Creation** component computes some document statistics, and may apply a weighting scheme that determines the relevance of a feature with respect to the document. The main step in the indexing process consists of inverting the stream of document-feature into feature-document information for the inverted index structure. Once this process is completed, the *Index* structure is available for searching. It is important to remark that the Index might be distributed across multiple servers.

For the querying process, we assume that there is an *Index* that supports the search process. The querying process starts with the user submitting a request (i.e., query) through the **User Interaction** component. This component is also in charge of applying some initial query transformations (e.g., spell checking, query expansion), and formatting the display of the search results listing. When the **Ranking** component receives the query, it recovers a set of relevant documents for it, and produces a sorted list of documents from the most relevant to the least relevant. This list is sent back to the user. The interaction with the ranked list is assessed in the **Evaluation** component, which is in charge of monitoring and improving the system performance, as well as logging the users' queries and their interactions with the system in the *Log data* server.

2.5 Assessing Search Results

The quality of the search results returned by a Multimedia IR system is not only measure based on the similarity between the retrieved documents and the query. The usefulness to the users' information needs must also be considered. Indeed, the way in which the results are presented to users plays an important role on the quality assessment. For instance, a flexible, navigable presentation of “*average quality*” results may work better than a rigid, list-based presentation of “*high-quality*” results. In this section, we discuss different ways to present search results employed by most search engines, as well as the different ways users might interact with a search engine. We also describe the main evaluation metrics employed in

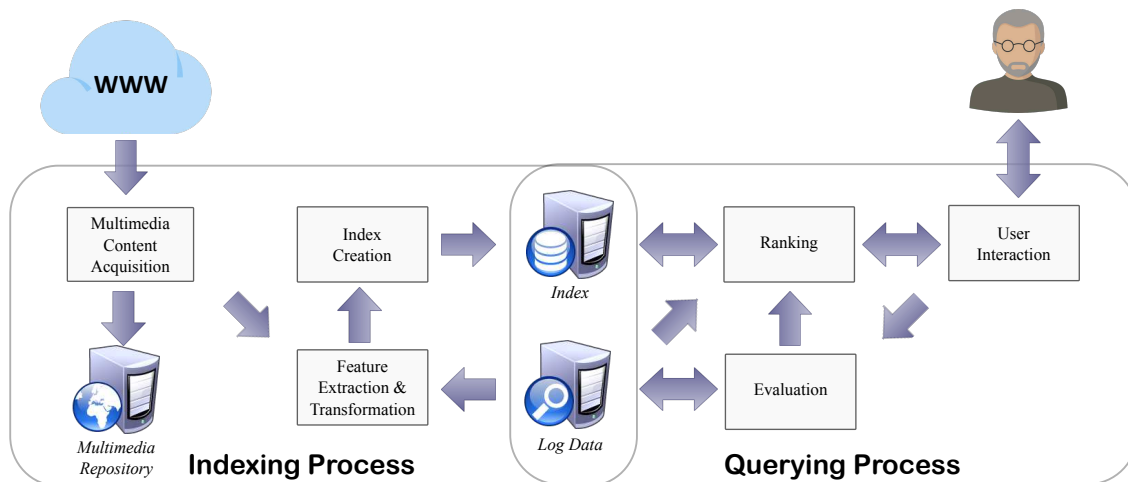


Figure 2.7: Multimedia Retrieval System Architecture¹¹ includes two main processes: indexing and querying. In the Indexing Process, the Multimedia Repository and Index are built. In the Querying Process, the user requests information from the IR system using a query. The IR system uses the Index as the main source of information for retrieving relevant documents. All interactions between the user and the system are registered in the Log Data.

IR systems, focused on effectiveness and efficacy, and point out some Web-based collections employed to assess large-scale multimedia retrieval systems.

2.5.1 Presenting the Results

Once a user request is complete and the search engine has retrieved a set of relevant multimedia documents, they must be shown to the user in a friendly and useful fashion. Current commercial Web search engines that provide multimedia search functionality have a grid-based template for presenting their search results listings.

For image searches, search engines such as Google¹² or Bing¹³ provide a thumbnail preview of the relevant images, and when a user clicks on one, a bigger version of the image appears, as shown in Figure 2.8. This way the user can decide if she wants to open the Website that contains that image.

For video searches, engines such as Yahoo!¹⁴ and Google provide different display templates. While Yahoo! opts for a grid-based presentation, in which you can click a video and get an embedded preview, Google displays the results as a list (similar to YouTube¹⁵). Figure 2.9 shows the search results page from the previously mentioned search engines.

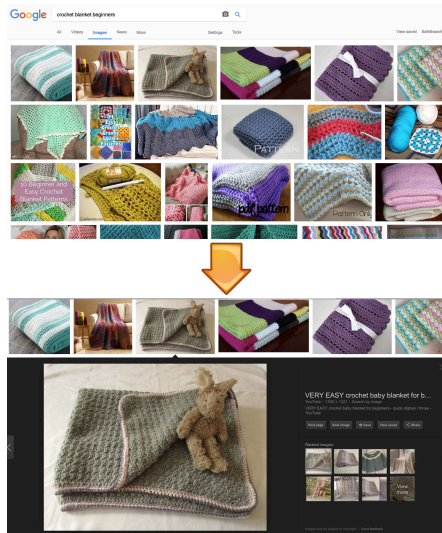
For audio searches, providing an interface that allows for a preview of the multimedia document is more difficult than for images and videos. In the case of audio, we noticed that

¹²<http://www.google.com>

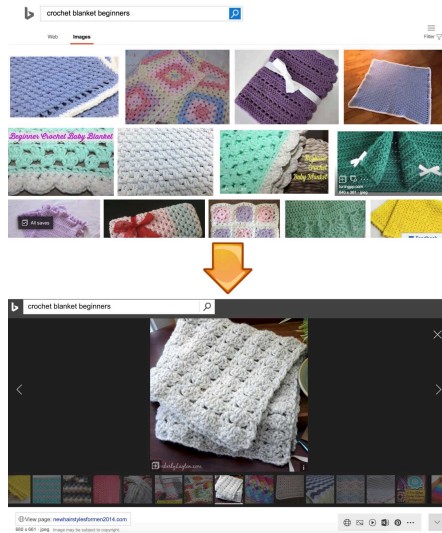
¹³<http://www.bing.com>

¹⁴<http://www.yahoo.com>

¹⁵<http://www.youtube.com>

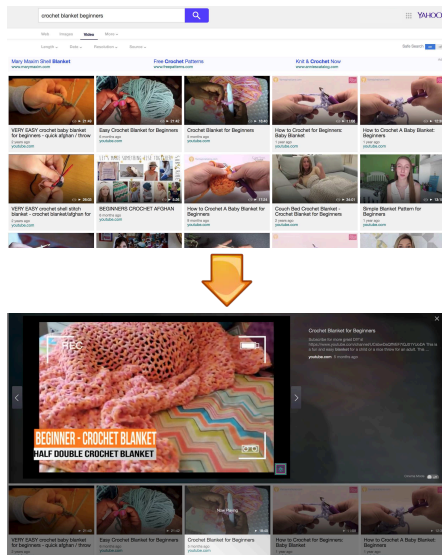


(a) Google image search

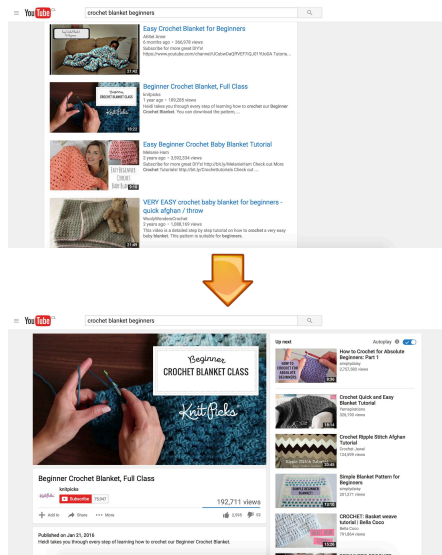


(b) Bing image search

Figure 2.8: Search results display for image searches in two different search engines. In both cases, the layout changes when a user clicks on an image.



(a) Yahoo video search



(b) Youtube video search

Figure 2.9: Search results display for video searches in two different search engines. Despite that the layouts are different, in both cases they show the video next to recommendations of other similar videos.

The screenshot shows the SoundDogs.com website with search results for 'cats'. The page has a navigation menu at the top with links like 'ABOUT SOUNDDOGS', 'FAQ & SUPPORT', 'PRODUCTION MUSIC', 'SOUND EFFECTS', 'SITE DEMO', and 'CONTACT SOUNDDOGS'. Below the navigation is a search bar and a list of search results. Each result includes a small audio preview icon, a description, the duration in seconds, and the price. For example, the first result is 'Voices, Gormy, Music For Cats, Accent, Transition' with a duration of 3 seconds and a price of \$2.73. The page also features a sidebar with various categories and a 'Live Chat' button.

(a) SoundDogs

The screenshot shows the FindSounds.com website with search results for 'cats'. The page has a search bar at the top and a list of search results. Each result includes a small audio preview icon, a description, the duration in seconds, and the price. For example, the first result is 'Voices, Gormy, Music For Cats, Accent, Transition' with a duration of 3 seconds and a price of \$2.73. The page also features a sidebar with various categories and a 'Live Chat' button.

(b) FindSounds

Figure 2.10: Search results display for audio searches in two different search engines. Search engines oriented to audio files do not have a standardized layout to present their results.

existing search engines provide a display fashion similar to that of text, and show annotations associated to multimedia documents as snippets. For example, SoundDogs¹⁶ shows the search results as a list (the ranking criteria is not clear). For each element in the list, SoundDogs provides information such as a brief description, the length of the audio (in seconds); and allows users to hear a *preview* of the audio. Similarly, FindSounds¹⁷ shows search results as a list with additional metadata. However, FindSounds includes a visual representation of the audio file using a visual representation of the wave sound. Figure 2.10 shows the search results of both audio search engines.

Contrarily to text-oriented Web search engines, for image search results it is not necessary to add snippets as for other type of resources. For video search results, it is helpful to have a brief description of the video, which might help users to select relevant videos when the preview static image does not provide enough information. Finally, for audio, it is completely necessary to include in the search result a textual description that summarizes the content in the audio file. Search engines face the challenge of displaying multimedia content and reducing the loading time of the result page, buffering audio-visual content efficiently to deliver it glitch-free. Accomplishing these challenges would provide a pleasurable user experience. [12]

2.5.2 User Interaction

Multimedia search engines not only have to cope with the data deluge¹⁸, but also with the task of including features that allow them to manage the huge amount of multimedia data distributed across the Web. Search engines have to provide enough guidance and support

¹⁶<http://www.sounddogs.com>

¹⁷<http://www.findsounds.com>

¹⁸Data deluge refers to the ever-growing amount of data or information published on the Web, as well as the challenges it represents.

to users to make their search interaction an enjoyable experience, through a smooth and friendly sequence of seamless interactions.

In this section we provide the four ways of user interaction with multimedia systems described in Blanken et. al [12], into the context of Web multimedia retrieval. These interactions span the scenario of users retrieving responses to their questions, to search engines providing personalized recommendations without explicit questions.

- **Retrieval:** This is the most common way we found to interact with multimedia content on the Web. Nevertheless, it requires the user to know what she is looking for before having an idea of the content available. This the scheme employed by commercial search engines. Regarding the user interaction side, there are two main aspects:
 - The user should know the syntax required by the search engine, such as special tags for advanced search functionality.
 - The user should formulate the query in such a way that it represents her information need.

Since search engines limit the amount of multimedia content a user can preview for a given query, the retrieval interaction is usually an iterative process of query formulation, results exploration, query reformulation, and so on. This process continues until the user finds some elements that satisfy her information need, or until she decides that the information searched is not available.

- **Dynamic query interaction:** Blanken et al. [12] describe this interaction as a very visual way to interact with the multimedia content through sliders, buttons, and other components that allow users to formulate and refine queries at a fast pace. Dynamic query interaction requires search engines to provide and update search results very quickly, based on changes in the user input. Hence, search sessions tend to be shorter than other types of interactions. Furthermore, queries in this type of interaction are not as expressive as those for keyword-based-queries.

Given the large amount of documents indexed by major commercial search engines (e.g., Google, Bing), or internally managed by multimedia hosting websites (e.g., Flickr, YouTube); a content-based dynamic query mechanism is not available in commercial search engines. Nevertheless, query suggestions are a good surrogate for this type of interaction.

- **Browsing:** This interaction between the user and the multimedia system does not require the user to input a query. Instead, the multimedia system provides a functionality that allows users to freely navigate the search space. Browsing is similar to entering a museum and walking around looking for a “nice” picture. Blanken et al. [12] state that “the characteristic for the browsing model is that there is no explicit specification of information need, like there is in query specification.”

Although Web search engines do not provide this interaction model, for some queries, such as popular queries, they provide tools for browsing the search results. For instance, Google provides a set of query related topics on top of their search results, which

might allow users to refine their query through browsing. Users may alternate between querying and browsing. In this fashion, users can access multimedia documents related to their original request without explicitly querying them.

- **Recommendation:** Unlike the interaction types previously described, the recommendation model does not need any input from the user. Recommendation interactions are initiated by the multimedia system, and do not require users to ask for any type of multimedia content. Since Web search engines do require an input from the user to start the search mechanism, recommendations are usually presented under labels such as “Related images” or “People also view”. On the other hand, multimedia related services such as Netflix¹⁹ and Spotify²⁰ do include these recommendations as they push new content to their users. For both platforms, personalizing recommendations is an important characteristic when attempting to provide better recommendations.

2.5.3 Evaluation

Assessing search engines requires measuring the relevance of the documents it returns (effectiveness), as well as the time and space it requires to accomplish its goal (efficiency). Measuring the effectiveness of search engines requires to count on a ground-truth built from user feedback on the relevance of documents returned for a given query. On the other side, measuring efficiency can be performed automatically. In this section we describe the main metrics for measuring effectiveness and efficiency in search engines.

- **Effectiveness metrics:** Effectiveness metrics can be organized in two main categories, based on whether they include the order of the documents on their computation or not. Ranking-independent metrics do not take into account the order in which documents are presented to the user. On the other side, ranking-dependent metrics consider the order in which documents are listed in the search results.

Ranking-independent metrics

We define the evaluation metrics following the notation described in the following matrix:

	Relevant	Non-relevant
Retrieved	TP	FP
Not retrieved	FN	TN

and $\#S$ represents the cardinality of a set S .

- *Precision(P)*: defined as the rate of relevant documents retrieved with respect to the full set of retrieved documents.

$$P = \frac{\#TP}{\#TP + \#FP}$$

¹⁹www.netflix.com

²⁰www.spotify.com

- *Recall(R)*: defined as the rate of relevant documents retrieved with respect to the full set of relevant documents.

$$R = \frac{\#TP}{\#TP + \#FN}$$

- *F-measure*: an evaluation metric that combines precision and recall to provide a better understanding of the overall relevance of the search results and the distribution of relevant documents. The F-measure is defined as the harmonic mean of Precision and Recall.

$$F = 2 \times \frac{R \times P}{(R + P)}$$

Ranking-dependent metrics

- *Precision at rank (P@k)*: Similarly to P , $P@k$ measures the rate of relevant documents over the set of retrieved documents in the top- k positions.
- *Precision at fixed Recall levels (P(R))*: This metric describes “the effect of precision and recall on the performance of an IR system” (Ceri et al. [21]). Standard recall levels are defined step-wise from 0.0 to 1.0, in increments of 0.1. $P(R)$ is defined as the maximum Precision value at a fixed standard Recall level. The values obtained using this metric are interpolated precision values.

$$P(R) = \max\{P' : R' \geq R \wedge (R', P') \in S\}$$

where S is the set of (R, P) pairs for a given results set.

- *Mean Average Precision (MAP)*: It represents a summary of the effectiveness of a retrieval system over a set of queries (specifically, its ranking algorithm). Before defining MAP, it is necessary to introduce the Average Precision (AP):

$$AP = \frac{\sum_{k=1}^n P@k \times \Delta r(k)}{\#TP}$$

where n is the number of retrieved documents for the query, and $\Delta r(k) = \lceil R@k - R@k' \rceil$ ($k' = k - 1$).

Once we know the AP for every query q in our dataset we proceed to compute the MAP as follows:

$$MAP = \frac{\sum_{q=1}^Q AP(q)}{Q}$$

where Q is the number of queries.

- *Discounted Cumulative Gain (DCG)*: This measure assumes that the relevance of a document is proportional to its utility, and also that highly relevant documents ranked in low positions are not useful for users, since those documents are not likely to be reviewed. The DCG is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i}$$

where rel_k is the relevance value of the document at the k -th position, and p is the position of the last document in the ranking.

The DCG is likely to have a value between 0 and an arbitrary number; this value might increment indefinitely if the ranking list contains many relevant documents. Hence, this value should be normalized in order to keep it within the interval $[0,1]$ at any point during the ranking assessment. The Normalized DCG (NDCG) can be computed as follows:

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

where $IDCG$ is the ideal DCG value for that query. The $IDCG$ corresponds to the DCG in the scenario in which documents on the results list are sorted by relevance in descending order.

- *Ranking correlations:* Relevance judgments are not always available; therefore, assessing retrieval systems requires other metrics that compute the effectiveness of retrieval systems, by means of surrogate preference information available in query logs. In that case, it is possible to measure the effectiveness of a retrieval system, by comparing its ranking with respect to user’s preferences stored in query logs. The *Kendall tau coefficient* (τ) is one metric to compare two rankings:

$$\tau = \frac{P - Q}{P + Q}$$

where P is the number of pairs for which both rankings agree with one another, and Q is the number that disagree. The τ value varies in the range 1 (rankings fully agree) and -1 (rankings fully disagree).

Example 2.4. Let us assume we are assessing a retrieval system that returns the following documents for two different queries. Using the information from the result list, we compute Precision, Recall, MAP, and NDCG.

Here we consider a ranking for query q_1 which contains 5 relevant documents in the full collection.

	1	2	3	4	5	6	7	8	9	10
Ranking A										
Relevance	✓	✗	✓	✗	✓	✓	✗	✗	✗	✓
Precision	1.00	0.50	0.67	0.50	0.60	0.67	0.57	0.50	0.44	0.50
Recall	0.20	0.20	0.40	0.40	0.60	0.80	0.80	0.80	0.80	1.00

Here we consider a ranking for query q_2 which contains 4 relevant documents in the full collection.

	1	2	3	4	5	6	7	8	9	10
Ranking B										
Relevance	✘	✘	✔	✔	✘	✔	✘	✘	✘	✔
Precision	0.00	0.00	0.33	0.50	0.40	0.50	0.43	0.38	0.33	0.40
Recall	0.00	0.00	0.25	0.50	0.50	0.75	0.75	0.75	0.75	1.00

The MAP of the retrieval system, based on these two queries, is computed as follows:

$$AP(q_1) = (1.00 + 0.67 + 0.60 + 0.67 + 0.5)/5 = 0.67$$

$$AP(q_2) = (0.33 + 0.50 + 0.50 + 0.40)/4 = 0.43$$

$$MAP = (0.43 + 0.67)/2 = 0.55$$

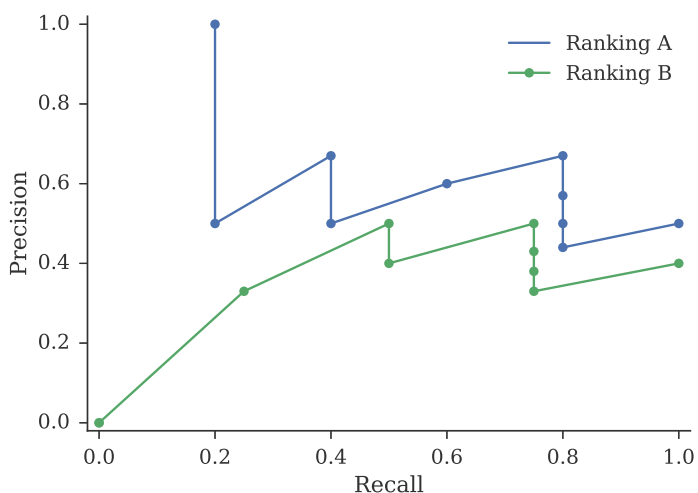
To compute the *NDCG* for query q_1 we assume that documents have a binary relevance value where relevant documents have *relevance* = 1, and irrelevant documents have *relevance* = 0.

$$DCG(q_1) = 1 + 0 + 0.63 + 0 + 0.43 + 0.39 + 0 + 0 + 0 + 0.30 = 2.75$$

$$iDCG(q_1) = 1 + 1 + 0.63 + 0.5 + 0.43 + 0 + 0 + 0 + 0 + 0 = 3.56$$

$$NDCG(q_1) = 2.75/3.56 = 0.77$$

Finally, the Recall-Precision graph depicting the P(R) values of query q_1 is:



- **Efficiency metrics:**

Measuring the efficiency of search engines does not require knowing the relevance of the retrieved documents, so the metrics can be automatically computed. We find three aspects related to search engines for which efficiency plays an important role: (1) index construction time, (2) index space, and (3) query response time. Table 2.1 exhibits a summarized list of the most common efficiency metrics.

Metric name	Description
Elapsed indexing time	Measures the amount of time necessary to build a document index on a particular system.
Indexing process time	Measures the CPU seconds used in building a document index. This is similar to elapsed time, but does not count time waiting for I/O of speed gains from parallelism.
Query throughput	Number of queries processed per second.
Query latency	The amount of time a user must wait after issuing a query before receiving a response, measured in milliseconds. This can be measured using the mean, but is often more instructive when used with the median or a percentile bound.
Indexing temporary space	Amount of temporary disk space used while creating an index.
Index size	Amount of storage necessary to store the index file.

Table 2.1: Definition of some efficiency metrics (Croft et al. [27])

Since index-related efficiency metrics do not directly affect the user’s perception of a search engine, the most commonly used metrics are those related to query response time, such as query throughput and query latency. Query throughput is an intuitive metric that measures the number of queries processed per second. To compare different systems, it is necessary to make their respective runs on hardware with the same characteristics. Query throughput values helps determine whether a retrieval system can handle its workload, or whether it needs to tune its capacity planning policies. On the other side, query latency measures the amount of time that passes since the user submits a query and until the system shows the results. Latency might be affected by the level of parallelism set up in the retrieval system. Low latency and high throughput are advisable properties, but they cannot be optimized at the same time.

Although users are not aware of the time and space consumed to build the index structure, this structure has a direct effect on the retrieval system performance. For example, building a large index that stores several pre-computed values might increase the query throughput and reduce the latency, but will likely take more time to build, and use additional storage space.

- **Collections:**

To advance the state-of-the-art in the field of Web Multimedia IR, it is essential to be able to reproduce and compare the performance of new proposals with respect to state-of-the-art approaches. Several communities dedicated to Multimedia IR foster the comparison of effectiveness using standard benchmarks based on real data. The Text REtrieval Conference (TREC²¹) and The Conference and Labs of the Evaluation Forum (CLEF Initiative²²) provide the scientific community with huge amounts of non-synthetic multimedia-related data. Similarly, the MediaEval initiative²³ has brought together various research groups interested in addressing multimedia retrieval using multimodal approaches that involve user-generated data. MediaEval emphasizes the use of multimodal data, and encourages the use of context data. Furthermore, MediaEval centered on the social and human aspects of multimedia retrieval. Table 2.2 contains descriptions to the main TREC, CLEF and MediaEval initiative datasets, and also to other independent efforts.

In addition to the initiatives of the multimedia community, major commercial Web search engines (i.e., Google Research²⁴, Microsoft Research²⁵ and Yahoo! Lab²⁶) have also put forth efforts to provide public datasets. For instance, Microsoft Research has made various datasets available that combine information from the Web with crowd-sourced context data. For Google Research, we notice a larger effort in building video-oriented datasets. This makes sense in that YouTube is a service owned by Google, and has been widely adopted in the Web community. Finally, regarding the variety of services provided by Yahoo!, the most diverse datasets are published by Yahoo! Labs. For example, Yahoo! Labs provides datasets based on user opinions, e.g. ratings. Also, given that Yahoo! owned Flickr for several years²⁷, they built one of the largest image datasets which can be used for many different tasks such as classification, geo-location, automatic tagging, among others. Table 2.3 summarizes the main datasets provided by commercial Web search engines.

Research in the multimedia community would greatly benefit from the use of benchmarks with a well-defined ground-truth, as well as from standard evaluation metrics. In this manner, the amount of reproducible research would increase, and the comparison between methods would be easier. Besides the ground-truth and metrics used, it is important to build benchmarks based on public datasets, or publishing all data employed in current Web Multimedia IR research in standard formats. Unfortunately, most research in Multimedia IR is conducted on hand-crafted datasets, which are not always available to other research groups. This is sometimes due to the lack of specific information required to boost the methods proposed, or because research is conducted within private organizations.

²¹<http://trec.nist.gov>

²²<http://www.clef-initiative.eu>

²³<http://www.multimediaeval.org>
















²⁴<https://research.google.com/research-outreach.html#/research-outreach/>

research-datasets

²⁵<http://research.microsoft.com/en-US/projects/data-science-initiative/datasets.aspx>

²⁶<https://webscope.sandbox.yahoo.com/#datasets>

²⁷Since April, 2018 Flickr is owned by SmugMug, an image hosting service. (<https://www.smugmug.com>)

Dataset		Size	Content	Data available	Explicit Context
				Implicit Context	
TRECVID					
IACC ^a		7,300 videos	Video files, shot reference		Title, keywords, description
BBC EastEnders		244 videos	Video files, shot segmentation, face recognition		Metadata embedded in video
BBC Archive		6,000 hours	Video files, shot segmentation, face recognition		Subtitles, descriptions, UK celebrities
HAVIC		9,300 hours	Video files		Events related to videos
MediaEval					
Flickr Places [25]	 	5M images 25K videos	Image and video files, visual features	Geographic data (lat, lon), text with geo location	
Flickr Images [49]		90K images	Image files	300 location queries	Related Wikipedia pages
Flickr Events [85]		500K images	Image files		Title, description, tags, geo coordinates
MusicBrainz music [98]		13K songs	Audio features	Artist, title, last.fm tags, genre, mood	
Blip 10000 [99]		14K videos	Video files, shot boundaries and key frames	25,000 tweets and 8,800 Twitter users	Title, description, duration, tags, transcripts
Other resources					
TREC Twitter Dataset ^b		240M tweets			Tweets posted from Feb. 2013 to Mar. 2013
CLEF Images from Search Engines [114], [39]		500K images	Image files and visual features	Web pages: (word, source, rank), (word, score)	
MSD ^c		1M songs	Audio features	Hotness	Title, year, artist, album
NUS-WIDE ^d		269K images	Visual features	Image URL, tags	
MIRFlickr ^e		1M images	Image files and thumbnails, and visual features	Creator, license, image URL, title, tags, exif	

^a Internet Archive videos under Creative Commons license.














^b <https://github.com/lintool/twitter-tools/wiki>

^c <https://aws.amazon.com/es/datasets/million-song-dataset>

^d <http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm>

^e <http://press.liacs.nl/mirflickr>

Table 2.2: Datasets for benchmarking - Part 1

Dataset		Size	Content	Data available	Explicit Context
				Implicit Context	
Google Research					
YouTube Video Games		120K videos	Audio-visual features		Title, tags, comments
YouTube Speakers		1,111 videos			Speaker ID, video URL
Youtube What's Cookin'		365K videos			Video URL, start/end timestamps, event name, tags
Microsoft Research					
Flickr Visual Annotations [131]		500 images			Image URL; 100,000 labels for objects in images
YouTube Video Description [22]		2,000 videos			85,000 descriptions about actions in videos
Yahoo! Lab					
YFCC100M ^a	 	100M media objects	Audio-visual features	Comments and favorites can be obtained using Flickr API	Id, user, URL, camera, timestamp, location, title, description, tags
Flickr European Cities (EC1M)		910K images	Visual features		Image URL, image relevance for 25 queries
Y! Musical Artist Ratings		10M ratings		10M artist ratings	
Y! Song Ratings		717M ratings		717M ratings of 136,000 songs	Song, artist, album, genre
Y! Internet Radio Playlists		4,000 stations			Radio station, track play, local/system time of play
Y! Movie Ratings		220K ratings		220,000 ratings of 14,000 movies	Cast, crew, awards, synopsis, genre, avg. rating
TVSum50 [104]		50 videos	Video files	Shot-level importance scores	Video URL

^a A subset of 200,000 images from YFCC100M with labels for 10 classes is also available.

Table 2.3: Datasets for benchmarking - Part 2

2.6 Summary

First, we introduce the *Triad of Search*: user, query, document, and describe the main characteristics of each entity from the triad. For example, based on the intention of the users (as well as their interaction with the search engine), they can be classified into three (3) main categories: searchers, surfers, and browsers. Each type of user has a different way of interacting with the search engine, as well as representing their information needs. We find that users can express their information need in different modes: text-based, with limited vocabulary (keywords) or free-text; and content-based, with one or more modalities. Currently, the Web has a huge amount of multimedia resources available for users that combines different modalities, such as posts in social platforms.

We found that there are big issues to be addressed for each of the entities in the triad of search. Regarding the query, we find that the *intention gap* is the most challenging aspect. Besides that, disambiguation of queries is extremely useful and most search engines have implemented auxiliary functions such as automatic query expansion and query suggestion, in order to improve the search experience. The main difference between these two procedures is that while the former is invisible to the user and is usually done back-end, the later performs front-end and requires user response.

With respect to multimedia documents, the *semantic gap* is the most challenging issue faced by the multimedia research community, and there are plenty of approaches addressing it. Recent proposals exploit machine learning techniques such as deep learning. We notice that current state-of-the-art approaches also employ different modalities combined. Mainly, we find that context data is a valuable source of information when attempting to understand multimedia content itself. Context data can be found on Web document structure, social platform posting metadata, and search engine query logs, to mention a few examples.

Regarding users, there are several ways in which they can interact with search engines. For example, a user could search for specific documents, browse within a collection, or receive personalized recommendations. In the past decade, search engine design has evolved to offer a friendlier user experience, such as providing easy-to-browse search results. Besides the interaction between search engine and user, the multimedia research community should be able to reproduce experiments of other researchers, as well as compare their new proposals with the state-of-the-art by using standard datasets and metrics. We find that initiatives such as ImageCLEF and MediaEval do a great effort by building standard datasets, and then designing challenges that use those datasets for solving problems of interest to the community. Besides those initiatives, we find that the main commercial Web search engines also provide some datasets with which the scientific community can benchmark state-of-the-art approaches related to Web Multimedia IR.

Chapter 3

Related Work

The main problems addressed in this thesis correspond to (1) discovering topics related to multimedia search results; and (2) automatically tagging multimedia documents. Thus, we describe state-of-the-art approaches for both topics. In addition, we also describe (3) user-based evaluations which are relevant to assess performance in the absence of suitable ground-truths.

3.1 Topic Discovery on Multimedia

Detecting topics associated with multimedia resources has been studied from different perspectives. In this section we focus on clustering search results, considering that each cluster of multimedia documents potentially stands for a topic. The *curse of dimensionality*¹ makes it difficult to accurately find “good” clusters. Therefore, current approaches for topic discovery model the data space as a graph and address the issue as a community detection problem.

3.1.1 Multimedia search results clustering

In recent years, *Multimedia retrieval* has become an important topic from the perspective of the quality of results, as well as the quality of experience provided to users. This means that a multimedia system must be easy to use and allow users to explore results seamlessly. Here we focus on methods related to clustering multimedia content, which has applications, e.g., to help users make sense of search results, or to automatically annotate multimedia resources, etc. Given the wealth of literature in this area, our goal is to provide a high-level summary of the main trends to cluster multimedia content.

¹The *curse of dimensionality* refers to the issues that arise when the dimensionality increases, and the volume of the space increases so fast that the available data become sparse. The term was coined by Richard Bellman in his book *“Dynamic programming”* [8]

- **Content-based multimedia clustering** relies on the extraction and analysis of audio-visual features. Most approaches for visual clustering are based on machine learning techniques, such as spectral clustering [73], manifold learning [132] and support vector machines [72]. In addition, sophisticated techniques [6, 30, 57, 70] for extracting accurate audio-visual features are computationally expensive. For example, spectral clustering can deal with data modeled in high dimensional spaces while a near optimum result is guaranteed [81]. However, it is computationally costly to apply in larger environments such as the Web. Hence, it is necessary to apply various heuristics and optimization decisions to enable scalability. Zheng et al. [135] propose a clustering algorithm named *Locality Preserving Clustering* (LPC) based on projections that map multidimensional elements to lower spaces and therefore reduce the computational cost.
- **Context-based multimedia clustering** exploits information other than content to discover partitions of similar multimedia documents. For example, the content of the webpage in which a multimedia object is embedded is a rich source of context information. The surrounding text, tags, and the page title are a suitable textual surrogate for multimedia descriptions [100] over which text clustering techniques can be applied. User-generated content, such as annotations, comments [31], and click-through data [109], also plays an important role in understanding and clustering multimedia content. Most text-based clustering methods are based on extensions of LDA [13]. For instance, Blei & Jordan [13] work on *correspondence LDA* for modeling text fields, such as captions, to employ them as annotations for images. Furthermore, there are variants of LDA focusing on annotating [88] or understanding [108] multimedia content. Within context-based clustering approaches, tag-based multimedia clustering leverages collaborative tagging systems to cluster multimedia resources. Although tag-based clustering could be seen as text-based clustering, using only tags we do not have the notion of word position or proximity (resources are typically associated with an unordered set of tags), nor we do have the notion of word frequency *within* a document (each resource is associated with zero or one occurrences of a tag). Thus, many works that proposed dedicated methods for clustering tags consider graph clustering methods applied over the graph of co-occurrences of tags. Begelman et al. [7] cluster tags using a graph clustering algorithm based on spectral bisection over a weighted version of the tag co-occurrence graph. Moellic et al. [76] propose a shared-nearest-neighbor approach to extract clusters based on both tags and content-based features. Gemmel et al. [38] apply hierarchical agglomerative clustering over tags to enable personalization, matching users with tags, clusters and resources they may be interested in. Papadopoulos et al. [83] employ community-detection methods to perform clustering of tags.
- **Hybrid multimedia clustering** combine some of the above techniques; in fact, many of the papers we discussed involve a combination of content, text, tags, etc. Some hybrid techniques model content and context data in the same space – usually as a graph structure [82, 117] or an embedded representation [42, 91, 108] – allowing for a more holistic clustering process where information from multiple spaces is considered together. Other methods process both types of data independently and subsequently complement the results for each type of data [111]; in this case, it is possible to combine the results of the specialized techniques previously discussed.

3.1.2 Community detection for topic discovery

Real world network representations frequently have one or more underlying community structures. Finding these structures is important for several reasons. For instance, communities may have different properties than the network that comprises them, or communities may stand for functional units in the network system. Also, community structures influence the speed at which information spreads on the network.

More formally, given a graph, a *community* is a sub-graph that is more densely connected than the graph in which it is contained. *Community detection* might be considered a graph partitioning problem where the goal is to identify the “densest” possible set of communities that form a partition. As described in the surveys of Fortunato [34] and Papadopoulos et al. [84], a variety of metrics and algorithms have been proposed to solve this problem. Initial community detection approaches apply a standard *minimum-cut* method that partitions a graph while reducing the cost of the cut (e.g., the number of edges broken). However, minimum-cut methods do usually force a fixed number of cuts or partitions, rather than identifying the “natural” communities in the graph.

More current community detection approaches are designed to achieve the goal of automatically identifying the best communities regardless of their number. One such way to do this is to specify a general metric that the community detection method should try to optimize (independent of how many communities are required). The most common metric is *modularity*, which is used by a variety of methods to measure the quality of communities. More specifically, modularity is the ratio of all edges in the graph that fall within the (candidate) communities, minus what would be expected in a graph with the same number of vertices and edges but where edges are assigned randomly (typically preserving the degree of vertices or the distribution from the original graph).

Community detection can then be posed as a modularity optimization problem. Although modularity offers quite a general foundation for community detection, in practice, it is not possible to compute it for most real world scenarios. First, the set of possible combinations of partitions grows exponentially with respect to the amount of vertices in the graph and thus finding the optimal community configuration is intractable, and indeed even computing the global modularity of a particular partition can be expensive. Practical algorithms often employ approximations such as local modularity measures [14], spectral analysis [80], and so on. Another issue is the *resolution limit* of modularity. This means that in large graphs with a relatively low mean vertex-degree even a single edge between two communities is seen as an “unlikely event” (assuming edges are randomly assigned). Under this condition, small communities end up merged (often unintuitively) into a few very large communities.

Given these practical issues with modularity, some authors have explored other options for community detection. For example, Girvan & Newman [41] propose a divisive algorithm that iteratively removes the edge with the highest betweenness (seen as a bridge between two parts of the graph) and then recomputes betweenness to select the next edge; however, this process of recomputing edge-betweenness is costly for large graphs. Other authors propose efficient algorithms that are more process- or structure-based. For example, Rosvall and Bergstrom [92] propose an information theoretic approach to the problem, using efficient

codings to identify communities. Raghavan et al. [90] model the community detection problem as a label propagation process, where closely-knit neighbors that eventually “vote” on the same label are seen as communities.

In previously described works, any node in the graph is eventually assigned to a community. As a result, it is possible to get groups of nodes that do not represent a real community. Hence, there are techniques that do not have this problem since they dismiss some nodes in the graph and consider only those that could belong to well-formed communities. Xu et al. [127] propose a variation of the clustering algorithm DBSCAN [32] for graphs. They define a similarity metric based on a graph structure. They define cores (i.e., nodes in a strongly connected neighborhood) as seeds for their clustering process. The novelty of this method is that it labels nodes as hubs and outliers. Hubs are nodes with neighbors that belong to at least two different clusters. Outliers are nodes that are not strongly connected to any node. Both methods are able to detect cohesive substructures in a graph.

In recent years, Papadopoulos et al. [83] propose a clustering technique, called HGC, for identifying related tags. Their proposal is inspired by the SCAN algorithm [127]. Similarly to SCAN, the first step of HGC focuses on the discovery of core nodes, which are used as seeds for detecting communities. The main difference with respect to the SCAN algorithm is that HGC does not require any parameter set up. In addition, it automatically discards all the nodes that are not relevant for any community.

In addition to ad-hoc algorithms for community detection, we consider general graph clustering algorithms as a key to finding related tags at a query level. The graph partitioning method proposed by Zaverovnik et al. [133] employs hierarchical clustering of connected nodes to determine groups of locally relevant nodes (a.k.a. islands) in a graph. They propose two different approaches to determine relevant elements in a graph. The first approach is based on employing node properties and picking nodes from the graph in descending order according to the value of this property. The second approach proceeds in a similar manner using properties of the edges.

3.2 Automatic Multimedia Tagging

Usually commercial search engines rely on text descriptions associated with multimedia documents as a first source of data for indexing such resources. Nevertheless, text descriptions are not always available and many Web resources might be left out. In this section we describe approaches that address the issue of automatically annotating multimedia resources. Since this thesis is oriented to analyze context data to enhance multimedia retrieval, our main focus is automatic tagging approaches that exploit multimedia context data or the combination of multimedia content and context data. For a more extensive reading on the topic of tag-related task, such as automatic tagging, tag refinement and tag retrieval, we refer to Li et al. [66].

3.2.1 Analyzing context data to tag multimedia

Understanding the context in which multimedia content is generated and consumed leads to provide better textual descriptions that might turn into effective tags and therefore improve the user search experience. There are many works that analyze explicit context data to propose new annotations for unannotated multimedia content. Most of the works rely on the soundness of the tagged dataset. However, building accurate tagged datasets with diverse topics requires vast human editorial effort. To bridge this issue, some research works (e.g., [23, 37, 68, 69, 101]) address the lack of huge training datasets using the *wisdom of crowds* [106]. In this fashion, datasets can be generalized and the initial tagged dataset is build upon collaborative work. Nevertheless, it requires a pre-processing stage at which the most representative tags are selected, and irrelevant tags are dismissed. For instance, Sigurbjornsson and van Zwol [102] exploit co-occurrence of tags to recommend relevant tags for images posted in online media sharing platforms. The proposed technique recommends a set of tags that can be associated to a picture.

Although manually assigned annotations add context information to images, this context information is constrained to the vocabulary of taggers. To bridge this problem, Antonellis et al. [2] analyze the quality of queries as tags. They find that queries are a rich source of information for tagging Web content. Also, Tsikrika et al. [109] propose to use clicks from experts as a source of information about user agreement between the queries and the images in the results list. Thus, clicks can simulate human annotations. Similarly, Leung et al. [62] propose an architecture for an adaptive search engine that could improve multimedia search results based on user feedback over the queries they submit.

Aside from annotations and queries, it is also possible to extract valuable context data from comments in online multimedia sharing platforms. Eickhoff et al. [31] propose a technique for identifying tags through the analysis of video comments from YouTube users. First, the method detects “bursts” in comments, which are short peaks of activity. The selected bursty comments are then used to infer the tags. They prove that comment streams are a suitable source of information to get meaningful tags without exploiting other metadata, or performing further multimedia content processing.

3.2.2 Combining content and context data to tag multimedia

The exploitation of context data along with content data boosts multimedia understanding. Thus, combining both types of data enables a considerable improvement in the effectiveness of tagging multimedia content. Content information employed in current automatic tagging approaches may range from user identity information [64], to tagging preferences [97], to user reliability [40], to image group memberships [54].

In recent years, research in the area has expanded rapidly. For instance, Li et al. [64] propose to determine tag relevance using a voting scheme that can be generalized for multiple features [65]. Every image gets a voting neighborhood (i.e., set of visually-similar image), and every image in the voting neighborhood contributes with a vote to each tag that agrees with

the target image. Also regarding image tagging, Wu et al. [123] propose to automatically annotate objects inside images by building a codebook from the concepts detected inside the images of their dataset. For most approaches that combine content and context data, it is common to enforce that similarity derived from multiple resources must be somewhat consistent with the image-tag association matrix [122, 126, 136].

Besides tag relevance, it is also important to focus on tag diversity. Qian et al. [89] propose a method that leverages tag diversity, so proposed tags “are not only highly relevant to the image but also have significant semantic compensations with each other” [89]. Furthermore, Kannan et al. [55] propose a method that produces text snippets for images by looking for near-duplicate images in Web pages, and then selecting the top-k snippets. As a result, images get related to snippets that are both relevant to the images and show diversity.

Aside from image tagging, Turnbull et al. [110] propose to annotate songs. The annotations they formulate take into consideration different objective and subjective aspects related to the songs. More recent, there are approaches that focus on automatically refining tags for specific frames inside a video [4] by exploiting user collective knowledge extracted from manual annotations, and content-based similarity between frames and images available on the Web. For a detailed literature review about recent approaches for automatic tagging, please see Li et al. [66].

3.3 Human Evaluation in Multimedia-related Scenarios

The evaluation of IR systems has been a relevant topic of research since these systems were first developed [3]. The Text Retrieval Conference² (TREC) was established to join efforts in order to benchmark IR systems, therefore improving search algorithms. The basic model for IR evaluation makes use of a collection that contains: (1) a corpus, (2) a set of queries, and (3) relevance assessments that indicate which documents are relevant to which queries. Using this information, evaluation measures show how well an IR system performs. This performance depends on the number of relevant documents retrieved and the position of these documents within the search results. Common measures include Precision (P and P@k), Recall (R), Mean Average Precision (MAP), and Discounted Cumulative Gain (DCG) [3].

In the early years of IR systems, evaluation frameworks were limited to user models based on experienced searchers with clearly defined search tasks. Hence, evaluations mainly centered on the topical relevance of the documents for a given query [58]. Web IR systems should not omit the strong influence of users in the quality of the results obtained. Nonetheless, when performing evaluations, researchers prefer to narrow down user influence by contextualizing search tasks. For instance, a researcher might define the level of experience a person requires to complete a search task, or how much time is available to complete it. The widespread use of commercial search engines provides valuable datasets that contain Web log data, which can be used to automatically assess IR systems. This logged data contains billions of search records, contributed by millions of users. This data is usually analyzed in an aggregated

²<http://trec.nist.gov>

fashion in order to obtain valuable information about users' opinions. However, these logs are not always publicly available.

Using search engines logs is not always possible. Thus, obtaining large datasets that combine information from corpus, queries, and relevance assessments is a challenging task. Over recent years, crowdsourcing has emerged as a suitable platform to perform large-scale relevance assessments [1]. Crowdsourcing was introduced by Jeff Howe [47] as the application of principles from the open source movement (i.e., community collaboration) to traditional jobs, in such a way that these jobs can be outsourced to a large group of people. The main reason behind the widespread use of crowdsourcing is that it makes it possible to conduct large experiments extremely fast, with good results, and at low cost. In spite of the advantages brought by crowdsourcing-based evaluations, there are several details that could make an experiment fail. For example, paying too little for time consuming tasks might convey users to give low quality responses; similarly, providing vague instructions (or not giving examples) could make it difficult for users to understand the requirements of the task. Alonso [1] proposes that it is important to consider user interface guidelines and inter-agreement metrics in order to gather useful results.

Amazon Mechanical Turk³ (AMT) has gained a lot of attention as a crowdsourcing platform to perform large scale evaluations. The service provided by AMT allows unexperienced users to easily design Human Intelligence Tasks (HITs) employing built-in templates. In addition, experienced users can design more complex HITs using AMT's API. This API provides a richer and more flexible service that can be accessed through the Command Line Tool (CLT). Regardless of the method employed to design a HIT in AMT, it is important to pay attention to the overall design of the experiment and its execution to gather useful results. In addition, AMT allows a requester to accept or reject individual user responses in order to reduce noise. When rejecting an assignment, it is important to explain to the user the reason for this decision. Alonso [1] recommends to paying users (workers) even when the response is not as good as one expected. We can always use a new qualification test to filter out lazy workers or pay bonuses to good ones.

3.4 Summary

In this chapter we review the main topics related to this thesis. Regarding *Topic Discovery*, we review approaches that address the problem of topic discovery as a clustering problem. We find that initial approaches focus on clustering multimedia content using content or context data, where recent approaches focus on leveraging the relationship between content and context data to improving clustering results. Since determining the similarity between two or more multimedia objects gets more difficult as the amount of dimensions included to model multimedia increases⁴, we opt to study techniques that use a graph representation to describe the relationship between multimedia documents. Specifically, we survey non-overlapping community detection techniques given that for these types of communities each

³<https://www.mturk.com/mturk/>

⁴This problem is known as the curse of dimensionality [8]

element in a graph belongs to a single community and it is not necessary to explicitly define a degree of belonging to a element with respect to each of the communities.

We find that most community detection techniques are oriented to optimize the graph partitioning based on modularity. Nevertheless, computing modularity cannot be done in polynomial time, and many approaches are based on approximations. Thus, in this thesis we extend the graph clustering approach based on islands [133], and propose a community detection technique able to automatically detect communities that is focused on optimizing the cohesion of each partition instead of its modularity.

Regarding *Automatic Multimedia Tagging*, we find that there are plenty of approaches that address this problem, so we focus on reviewing approaches that exploit multimedia context data, most of the time extracted from user-generated content. In the context of Web Multimedia IR, we notice that initial approaches focus on tagging images by analyzing text from Web pages and manual annotations as surrogates for image descriptions. Although explicit context data is the major source of multimedia context data, it is usually biased and loose, in the sense that user expertise directly impacts the quality of context data. Therefore, some research works explore the potential of implicit context data (for example, queries, and click-through data) and demonstrate the advantages of using it as a source of multimedia descriptions.

We find that regardless the context data type, the evaluation of automatic tagging approaches usually generalizes tags and groups them into classes with wide semantics. Given this scenario, our proposal in this topic (automatic tagging) focuses on leveraging implicit context data, specifically queries. But, unlike current approaches, we do not aim to map queries to classes which later may lose the semantics of what users wanted to express. Actually, we aim to generate long descriptions that support better indexing multimedia resources later on.

Finally, about *Human-based accuracy assessment* we summarize the challenges of gathering user opinions to assess multimedia-related systems. We review some good practices about crowdsourcing, which has emerged as a suitable mechanism to gather a large amount of responses for a wide range of tasks. As we notice from the list of datasets described in Chapter 2, many of them are built upon crowdsourcing tasks. We realize that in the majority of cases crowdsourcing was used as a mechanism for collecting initial context data, for example long descriptions and bounding boxes for specific objects inside images show and videos.

The use of human assessors' opinion in this thesis is oriented in a slightly different direction: we aim to collect opinions about the output of our framework, instead of using them as input for them. This involves challenges such as selecting a relevant and unbiased subset of output data to assess, as well as reducing spam in the responses and measuring their reliability. Furthermore, analyzing the data collected for explaining the behavior of assessors, even when responses are dissimilar, is also a challenging aspect of applying crowdsourcing to assess multimedia related systems.

Chapter 4

Topic Identification in Multimedia Search Results

The deluge of multimedia data on the Web has raised the need for multimedia search engines to provide mechanisms to improve user interaction with search results. Additionally, query-by-keyword is inherently ambiguous, and in the specific case of multimedia searches it might not depict the true user intention when they search for non-textual documents, such as images. Thus, modern search engines take into account a variety of signals (e.g., similarity measures over text) to make an accurate guess on what the user is most likely interested in. To attempt to satisfy as many users as possible, a multimedia search engine needs to provide a degree of diversity in the returned results, be it by visual diversity, topical diversity, or both. Several studies have shown that grouping similar multimedia documents helps users to quickly make sense of the search results. Indeed, users may browse different categories while providing implicit feedback about the intention behind their initial request.

In this thesis, our focus is on topic-based clustering, which can help users make sense of search results by making explicit the different interpretations of ambiguous textual queries, or different aspects of the same interpretation. Topic-based clustering traditionally relies on textual features, which are usually obtained from contextual information; for instance, the text that surrounds an image in a Web page, or human annotations, also known as tags. Regarding the wide adoption of multimedia social sharing platforms, specifically of the use of tags as keys to describe multimedia content, we aim to use tags as surrogates for short descriptions of the multimedia content returned by a search engine for a given query. We propose an online query-centered framework for topic discovery which takes as input tagged search results, and models tag relationships as a graph over which a community detection technique is applied. As a result, the framework returns a set of topics, related to a given query, represented by a set of tags.

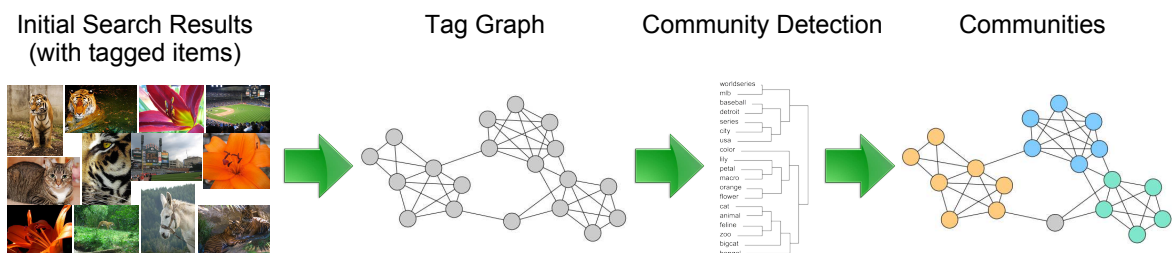


Figure 4.1: Framework for topic detection: Given a set of images from a query search results, it builds a tag co-occurrence graph over which a community detection algorithm is applied. Each community returned by the community detection algorithm represents a concept associated with the initial query.

The main characteristics of our framework are:

- **Multimedia-type independence:** We do not employ content-based features in the tag graph construction process. Our model could potentially discover topics across different types of multimedia resources in a transparent fashion.
- **Tag and topic independence:** Our framework does not require any training data, it is not fixed to domain or language, nor to a limited or fixed number of topics.
- **Query specific:** Tag graphs are built with respect to a specific query, which helps disambiguate (non-query) polysemous tags; e.g., the tag `jaguar` appearing on results for a query “zoo” will (likely) only refer to the cat, not the car.
- **Online detection of concepts:** Given adequate physical infrastructure and optimized community detection algorithms, it is possible to perform multimedia topic detection in an online fashion (e.g., on the client side).

4.1 Framework for Detecting Multimedia-related Concepts

In this section we describe our framework for online topic detection. We first introduce the general framework, specifically, we point out the input and output of each step. And, in subsequent sections, we describe each step of the framework in detail.

4.1.1 Overview

We introduce a framework to detect semantically relevant groups of tags (“topics”) associated with queries encoding user information needs. We are motivated by the use-case of performing an online clustering of heterogeneous multimedia search results using only the tags in the results. In Figure 4.1 we show the proposed framework, which consists of the following four main stages:

1. **Retrieval of multimedia resources based on a specific query:** Search results are the starting point for our framework. In order to keep our model as general as possible,

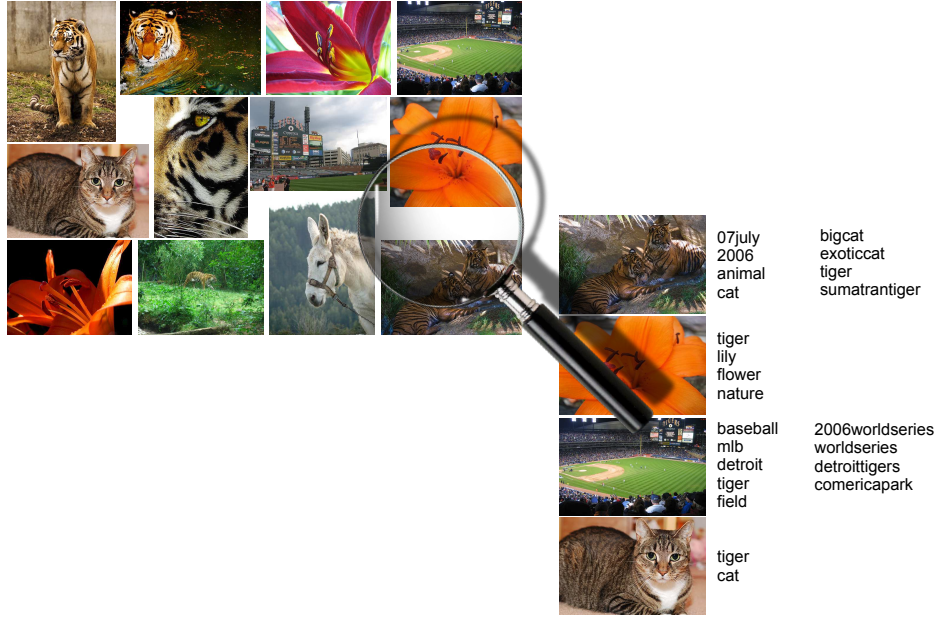


Figure 4.2: Image search results for query “tiger” and a subset of images in the results with their respective annotations.

we assume as input a collection of items (instead of a specific type of multimedia data) associated with a set of tags, over which we perform clustering using tag co-occurrence information. Since we only use tags to represent multimedia content, our framework is able, in theory, to combine different types of multimedia resources (such as, images, videos, audio) in a transparent fashion.

2. **Construction of the Tag Co-Occurrence Graph (*Tag Graph*):** The tag graph is the key structure over which our framework works. Given a finite set of images R retrieved for a given query, where each resource is associated with a set of tags, we define the tag co-occurrence graph as

$$G_\lambda = (V, E, \lambda)$$

where:

$V = \bigcup \{r \in R\}$ r are tags associated to resources, such as images, in R ,

$E = \{(v, v') \mid \exists r \in R \text{ such that } v \in r, v' \in r \text{ and } v \neq v'\}$ are edges that represent an explicit relationship between tags in V , and

$\lambda : E \rightarrow R$ is a weighting function that labels each edge in E with a real value, such as the number of co-occurrences, or the structural similarity between two tags in V .

In Figure 4.3 we show the tag graph for a partial set of annotations related to the results for query “tiger”.

3. **Tag Clustering based on Community Detection Algorithms:** We employ community detection algorithms because of their flexibility to compute cohesive groups of nodes in a graph without the explicit indication of the number of groups or the size of them. Under this approach, no additional knowledge about the multimedia resources is necessary to detect relevant groups of tags (other than the tag graph structure). Initial

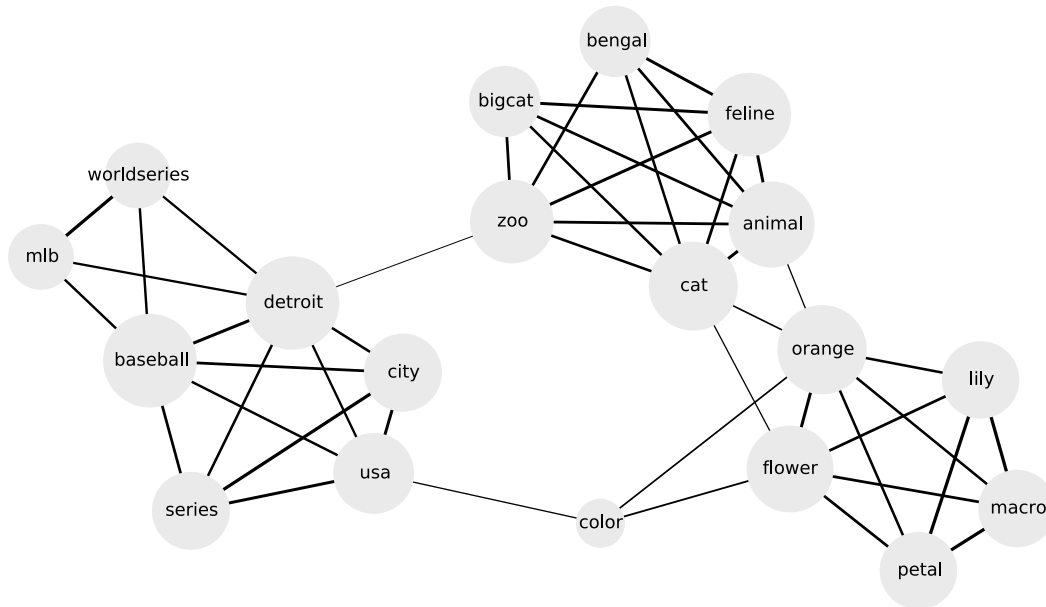


Figure 4.3: Tag Graph example for query “tiger”: nodes are tags associated to resources in the search results and edges indicate co-occurrence of tags.

empirical experiences revealed that using existing community detection algorithms to cluster search results from the Flickr image search engine often led to senseless topics (e.g., we found clusters that were too large, and grouped unrelated terms). Hence, we also propose two new community detection algorithms based on island cuts.

4. **Topic representation using tags:** Ideally, it would not be necessary to apply any additional process to the output of the community detection algorithms. However, for algorithms that return large communities, we would need to apply a ranking technique in order to reduce the subset of tags to a manageable size. The simplest approach to ranking tags inside a community would be sorting them by frequency or degree.

4.1.2 Tag Graph Construction

We now provide preliminary definitions relating to tag co-occurrence that we use throughout the chapter.

(Weighted) Tag co-occurrence graph: connects two tags if and only if they are associated with a common resource (e.g., a common image, video, or music file). We define the *weighted tag co-occurrence graph* $G = (V, E, \lambda)$ of a *bag*¹ of resources R where V and E are defined as before, and where $\lambda : E \rightarrow \mathbb{R}$ is a (total) weighting function that assigns each edge in E with a real value.

¹A bag is a set that allows duplicates. We need to consider a bag for the weighted version since, e.g., two images may have the exact same set of tags.

We can consider different weighting functions for λ , for our Tag Graph the simplest of which counts the number of co-occurrences for a pair of tags:

$$\lambda(v, v') = \#\{r \in R \mid v \in r \text{ and } v' \in r\}$$

Here, $\#S$ denotes the cardinality of the set S . In this case, λ maps edges to positive integers. We call this scheme *cardinality-weighting*, or *c-weighting* for short.

In initial experiments, we found that cardinality-based weights were sensitive to the number of resources considered, and that isolated tag pairs can sometimes occur quite frequently, perhaps due to the tagging preferences of a single user. Thus, we investigate a more robust weighting scheme based on *structural similarity* [127], which is defined for two nodes v and v' as follows:

$$\text{sim}(v, v') = \frac{\#\text{nv}(v) \cap \text{nv}(v')}{\sqrt{\#\text{nv}(v) \times \#\text{nv}(v')}}}$$

where $\text{n}(v) = \{v'' \mid (v, v'') \in E\}$ are the neighbors of v in the undirected graph and $\text{nv}(v) = \text{n}(v) \cup \{v\}$ includes v ; here v is included to count the case that v and v' are connected. Structural similarity is thus the number of neighbors the vertices share in common divided by the geometric mean of the number of total neighbors they have. The result is a value in the interval $[0, 1]$ (inclusive): if the vertices share no neighbors, the value is 0, whereas if they share all neighbors (and are connected), the value is 1. We call the scheme where $\lambda(v, v') = \text{sim}(v, v')$ *similarity-weighting* or simply *s-weighting* for short.

For brevity, we refer to the (*[c/s]-weighted*) *tag co-occurrence graph* as simply the “([c/s]-weighted) tag graph”. Both weighted graphs are undirected, edge-labeled graphs.

Example 4.1. A partial view of the tag graph for the query “tiger” is shown in Figure 4.3. Vertices are individual tags. Edges between tags indicate that they are used as tags for at least one common resource. Note that in the online clustering scenario, we remove tags that match the query term itself since they conceptually belong to all clusters and are not useful for detecting topics. Likewise, note from the graph that some vertices may connect with multiple clusters; in some cases this is due to polysemy, or as in the case of **orange**, simply because they relate to both topics.

In the same figure, we also encode c-weights using the boldness of the edge line: the more frequently two tags co-occur, the heavier their line. Based on this graph, we could also compute the s-weights of two nodes: for example, $\text{sim}(\text{detroit}, \text{zoo}) = \frac{2}{\sqrt{8 \times 7}} \approx 0.27$, while $\text{sim}(\text{detroit}, \text{city}) = \frac{5}{\sqrt{8 \times 5}} \approx 0.79$. \square

4.1.3 Community Detection Algorithm

Community detection algorithms provide a suitable way to detect structurally cohesive graph structures. In the literature we found two main types of community detection algorithms: (1) non-overlapping community detection algorithms, and (2) overlapping community detection

algorithms. The main difference between these approaches is that the former ones return communities for which each node n belongs to a single community, while the latter consider that a node might belong to multiple communities. Specifically, non-overlapping community detection algorithms are a simple and effective tool to determine if the tag co-occurrence structure represents a community. Nevertheless, in our preliminary analysis of community detection techniques for topic discovery we found that techniques can behave in two different ways: (1) some techniques return a few communities with a large amount of elements in each one, and (2) other techniques return many communities with a smaller size than former approaches. For the first group, methods do not return sets of terms from which topics can be detected straightforward. This is mainly because the returned groups contain many noisy terms. On the other hand, the second group of techniques offer the possibility to obtain smaller groups, which can be easier to understand for humans, and thus could lead to topics being quickly identified. However, it is also possible that some topics are split into several clusters of terms. It is important to remark that the amount of terms related to a topic cannot be generalized, and therefore state-of-the-art methods must be able to identify the suitable size for each topic.

4.2 Adaptive Island Cuts for Topic Detection

In this section we describe the algorithm for topic detection based on island cut with an adaptive size of clusters. First, we present the definition of island as introduced by Zaveršnik and Batagelj [133]. Then, we describe two fashions to build a hierarchy of islands. Finally, we describe the algorithm for adaptive selection of islands.

4.2.1 Islands

An island is a sub-graph that is maximal in its neighborhood for a given property of the graph [5, 133]. Zaveršnik & Batagelj consider two types of islands: *vertex islands* and *edge islands*. An island is defined relative to some vertex (or edge) property p , where no external neighbor of the island has a higher value for p than any vertex (or edge) in the island. Also, if no such neighbor has an equal p value to a vertex (or edge) within the island, that island is *regular*.

We call a sub-graph of an (x -)island that is itself an (x -)island an (x -)*sub-island*, where x could be vertex, regular vertex, edge, or regular edge. We call a vertex $v' \in V'$ a *port* of a vertex island if it has the lowest value for p in that vertex island, and removing v' and its associated edges yields a vertex sub-island. A vertex island may have multiple ports. We also call an edge e that can be removed from an edge island to yield an edge sub-island a *port*; it does not necessarily need to have the lowest value for the edge-property w .

Islands are built following a “greedy” algorithm that initializes islands with the vertices (or edges) that have the highest values for the given property, and then enlarge and/or combine those islands by traversing the graph from these starting points to include their neighbors.

This process is similar to building a spanning tree, except that it records the order in which the vertexes (or edges) are added to the solution. We now describe two greedy algorithms for extracting islands and later discuss how we choose communities from those islands.

4.2.2 Island hierarchy based on edges

To build the hierarchy $H = (V_H, E_H)$ based on an edge property for a given graph $G = (V, E)$, edges in G must be sorted in descending order with respect to their edge property p . First, the hierarchy of islands is filled with single-vertex islands for each successive $v \in V$, and adding it to V_H . Next, for every edge $e = (v_i, v_j) \in E$, if v_i and v_j belong to different islands I_i and I_j respectively, a new super island is created with all nodes I_i and I_j . This super island replaces its sub-islands I_i and I_j in the hierarchy. The detailed process is described in Algorithm 1. Furthermore, Example 4.2 illustrates the process based on the structure of graph G shown in Figure 4.3. The final result for the example is depicted in Figure 4.4.

Algorithm 1 Algorithm for computing hierarchy of nodes based on edges property

```

1: procedure BUILDEGEHIERARCHY( $G$ )                                ▷  $G$  is a tag graph
2:    $E \leftarrow \text{edge\_attributes}(G, \text{weight})$                     ▷  $Tuples : (v_1, v_2, \text{weight})$ 
3:    $\text{sort\_reverse}(E, \text{key} = \text{weight})$ 
4:    $\text{hierarchy} \leftarrow \text{get\_nodes}(G)$ 
5:   for all  $\text{island} \in \text{hierarchy}$  do
6:      $\text{island.port} \leftarrow \text{NULL}$ 
7:   end for
8:   for all  $(v_1, v_2, \text{weight}) \in E$  do
9:      $i_1 \leftarrow \text{get\_island}(v_1, \text{hierarchy})$ 
10:     $i_2 \leftarrow \text{get\_island}(v_2, \text{hierarchy})$ 
11:     $\text{island} = [ ]$ 
12:    if  $i_1 \neq i_2$  then
13:       $i_1.\text{regular} \leftarrow (i_1.\text{port} = \text{NULL} \text{ or } i_1.\text{port.weight} > \text{weight})$ 
14:       $i_2.\text{regular} \leftarrow (i_2.\text{port} = \text{NULL} \text{ or } i_2.\text{port.weight} > \text{weight})$ 
15:       $\text{island.port} \leftarrow (v_1, v_2)$ 
16:       $\text{island.subisland}_1 \leftarrow i_1$ 
17:       $\text{island.subisland}_2 \leftarrow i_2$ 
18:       $\text{hierarchy} \leftarrow \text{hierarchy} \cup [\text{island}]$ 
19:       $\text{hierarchy} \leftarrow \text{hierarchy} - [i_1, i_2]$ 
20:    end if
21:  end for
22:  for all  $\text{island} \in \text{hierarchy}$  do
23:     $\text{island.regular} \leftarrow \text{True}$ 
24:  end for
25:  return  $\text{hierarchy}$ 
26: end procedure

```

Example 4.2. Let us assume we sort the edges in the tag graph of Figure 4.3 in descending order with respect to the edge property w , in this case s-weighting:

Edges sorted by structural similarity value

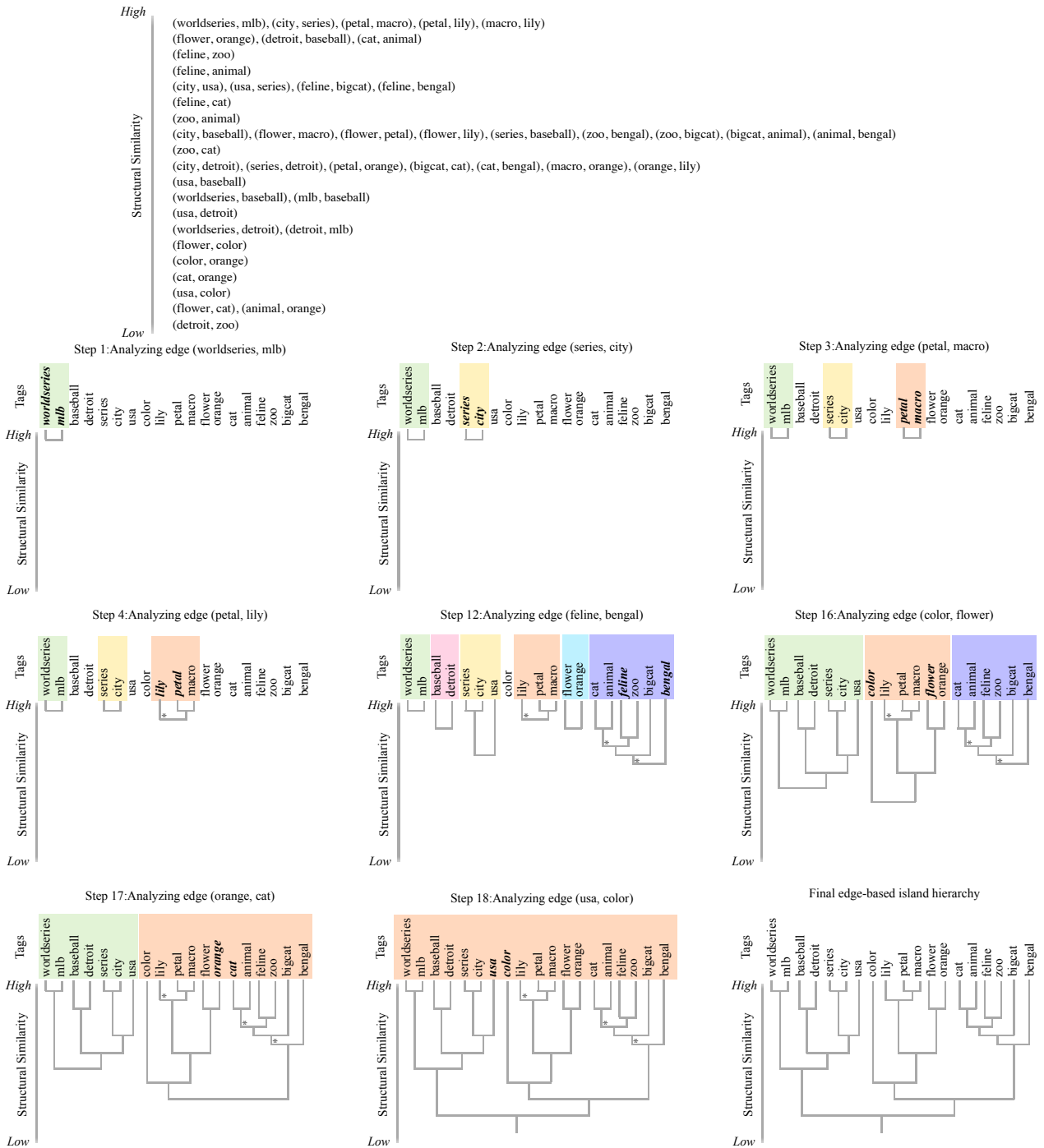


Figure 4.4: Edge island hierarchy for annotations related to query “tiger” (see Tag Graph in Figure 4.3). Each step processes an edge of the set sorted in descending order by structural similarity value. If the edge being analyzed is connected to the port of any of the hierarchies, it is added to such set; otherwise, a new hierarchy is built with such edge. (*’ means that we include an edge with equal value than the port.)

$E = \{(\text{worldseries}, \text{mlb}), (\text{city}, \text{series}), (\text{petal}, \text{macro}), (\text{petal}, \text{lily}), \dots, (\text{color}, \text{usa})\}$
 This time, $H = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \subset 2^V$ – sets of vertices representing islands – and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We initialize \mathcal{V} with all singleton vertices and \mathcal{E} as empty. We take the first edge $(\text{worldseries}, \text{mlb})$. We retrieve the largest islands in \mathcal{V} containing both nodes; in this case $\{\text{worldseries}\}$ and $\{\text{mlb}\}$. If they are the same islands, we continue to the next edge. If they are not the same, we create a new island which is a union of the two – $\{\text{worldseries}, \text{mlb}\}$ – and add a directed edge from the new island to the two old sub-islands. Once all of the edges are exhausted, H is a tree where all nodes in \mathcal{V} are edge-islands and all edges in \mathcal{E} represent a sub-island relationship. We show the process to build the hierarchy in Figure 4.4. In practice, we do not store the vertices representing larger islands, but instead store the hierarchy from which the islands can be generated. Once again, for an edge-island to be regular, it cannot have an incoming edge in H that was derived from an edge with the same value for w as the outgoing edges of the island in H . To illustrate this, we provide an example: while $\{\text{petal}, \text{macro}\}$ is an edge-island, it is not regular, since the original edge $(\text{petal}, \text{macro})$ in G had the same weight as $(\text{lily}, \text{petal})$ (and indeed $(\text{lily}, \text{macro})$). On the other hand, $\{\text{petal}, \text{macro}, \text{lily}\}$ is regular. \square

4.2.3 Island hierarchy based on vertices

To build an island hierarchy $H = (V_H, E_H)$ based on a vertex property for a given graph $G = (V, E)$, vertices in G must be sorted in descending order with respect to their vertex property p . The process starts creating single-vertex islands for each successive $v \in V$, and adding it to V_H . Next, v is connected to all ports v' (the last vertex added to an island) of existing islands in H that are neighbors of v in G : we add the edge (v, v') to E_H , where v now replaces each such v' as a port for a new larger island. The detailed process is described in Algorithm 2. Furthermore, Example 4.3 illustrate the process based on the structure of graph G shown in Figure 4.3. The process described on the example is depicted in Figure 4.5.

Example 4.3. Let us assume we take PageRank [17] as the vertex property p . If we compute the PageRank of every vertex in the tag graph in Figure 4.3 and sort them in descending order we obtain:

$V = (\text{baseball}, \text{detroit}, \text{orange}, \text{cat}, \text{flower}, \dots, \text{color})$

We start with $H = (V_H, E_H)$ blank. Iterating over V , we first add **baseball** to V_H . Next, we add **detroit**, which is a neighbor of **baseball** w.r.t. G , where **baseball** is a port in H , so we add the directed edge $(\text{baseball}, \text{detroit})$ to E_H . Then, we add **orange**, but do not connect it to anything since it has no neighbor w.r.t. G already in H . We continue in this manner until we reach the hierarchy shown in Figure 4.5. In the case of **zoo**, for example, we can see that a vertex can be connected to two ports. In the case of **feline**, vertices are only connected to ports. Likewise, nodes with the same value for p have bidirectional links. We can get the vertices of an island by taking any vertex and retrieving all of its ancestors. For an island to be regular, the ancestors must include vertices reachable through bidirectional links. \square

Algorithm 2 Algorithm for computing hierarchy of nodes based on vertex property

```
1: procedure BUILDVERTEXHIERARCHY( $G$ ) ▷  $G$  is a Tag Graph
2:    $V \leftarrow \text{node\_attributes}(G, \text{weight})$  ▷  $Tuples : (v, \text{weight})$ 
3:    $\text{sort\_reverse}(V, \text{key} = \text{weight})$ 
4:    $\text{hierarchy} \leftarrow []$ 
5:   for all  $v \in V$  do
6:      $\text{island} \leftarrow \text{NULL?}$ 
7:      $\text{island.port} \leftarrow v$ 
8:      $\text{neighbors} \leftarrow \text{get\_neighbors}(G, v.\text{node})$ 
9:     for all  $\text{islandin}$   $\text{hierarchy}$  do
10:      if  $\text{island.port.node} \in \text{neighbors}$  then
11:         $\text{island.subislands} \leftarrow \text{island.subislands} \cup [\text{island}]$ 
12:      end if
13:    end for
14:     $\text{hierarchy} \leftarrow \text{hierarchy} \cup \text{island}$ 
15:    for all  $\text{isl} \in \text{island.subislands}$  do
16:       $\text{hierarchy} \leftarrow \text{hierarchy} - \text{isl}$ 
17:       $\text{isl.regular} \leftarrow \text{island.port.weight} > v.\text{weight}$ 
18:    end for
19:  end for
20:  for all  $\text{island} \in \text{hierarchy}$  do
21:     $\text{island.regular} \leftarrow \text{True}$ 
22:  end for
23:  return  $\text{hierarchy}$ 
24: end procedure
```

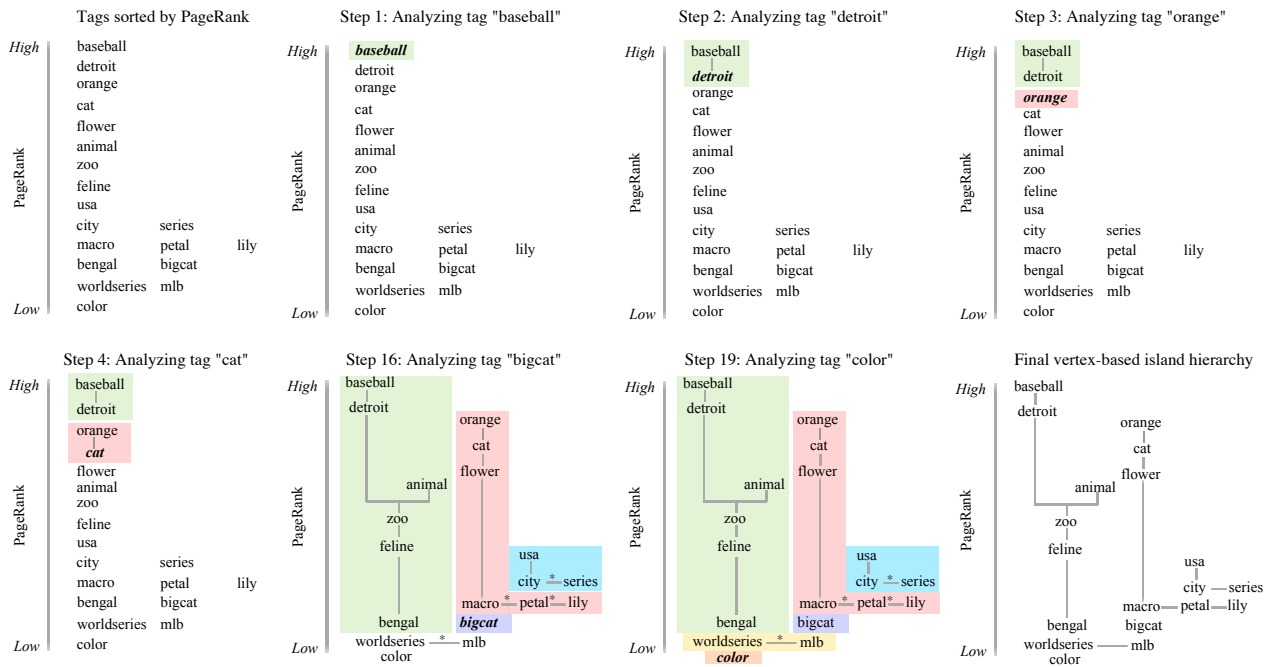


Figure 4.5: Vertex island hierarchy for annotations related to query "tiger" (see Tag Graph in Figure 4.3). Each step processes a tag of the set sorted in descending order by PageRank value. If the tag being analyzed is connected to the port of any of the hierarchies, it is added to such set; otherwise a new hierarchy is built using the tag as root. (*' means that two tags with equal PageRank value are connected)

4.2.4 Adaptive island cuts

Using information from the island hierarchy, we proceed to select sub-sets of nodes connected through the hierarchy (sub-islands) which represent coherent concepts. The selection of suitable islands from the hierarchy is not a straightforward task. There are some trivial islands to be avoided, such as single-vertex islands, and an island that contains all vertices in the graph. To simplify the selection of islands, Zaveršnik & Batagelj [5, 133] proposed a simple criteria for islands of interest using a bound $[k, K]$, where k is a lower bound on the number of vertices of valid islands, and K is an upper bound.

In the context of topic discovery it is reasonable to assume a minimum island size of $k = 3$ given that a smaller size would not carry clear semantics. For example, a community of size = 2 only reflects a co-occurrence of terms, and a community of size = 1 is trivial. Nevertheless, specifying a maximum island size would be very restrictive, and it would constrain the verbosity with which a concept is represented in a given query search results. Instead of setting a fixed value for the upper bound K , we propose to set a threshold on the (sub-)graph density (i.e., the ratio of edges to nodes) for what we consider to be semantically relevant islands. The main intuition behind this idea is that the larger and the closer to a clique an island is, the better: in this sense, there is a trade-off between the density and the size of the island. This trade-off is captured using a density threshold:

$$\delta(x) = \frac{x(x-1)}{2} \times \max\left(\log_2\left(\frac{x+k}{x}\right), t\right)$$

where x is the number of vertices in the island, k is the minimum number of vertices allowed for an island ($k = 3$), and t is a fixed lower bound that we discuss presently. The left term of the product is the number of edges in a clique with x vertices (excluding loops). Assuming $t = 0$, the rightmost term is a logarithmically decaying ratio on the number of vertices in the range $(0, 1]$. When $x = 3$, for example, the ratio is 1, meaning that the island must be a clique. When $x = 6$, the ratio is approximately 0.58, requiring the island to have 9 edges (versus a 6-clique of 15 edges). However, initial tests returned large islands with low density. Thus, we added t as a practical compromise: it offers a parameterizable fixed lower bound on the ratio, to ensure a minimal density for larger islands; for example, if $t = 0.33$, then the right-hand side ratio remains fixed for islands larger than 12, and will not go lower. Figure 4.6 illustrates that our proposed density function, which is an adaptation of Benford’s law², is less restrictive than the well-known Zipf’s law. In the current scenario, we apply Benford’s law as a normalization factor of our edge density coefficient because it will force communities with smaller sizes to have a higher density than those with bigger sizes. Despite the fact that Zipf’s law has a similar graph curve than Benford’s law, the slope of Benford’s law is not so steep at the first positions. Furthermore, Figure 4.7 shows the difference between the number of edges of an unbounded density function with respect to the clique distribution, as well as our t -parameterized bounded density.

²See Appendix A for further information about Benford’s law and Zipf’s law

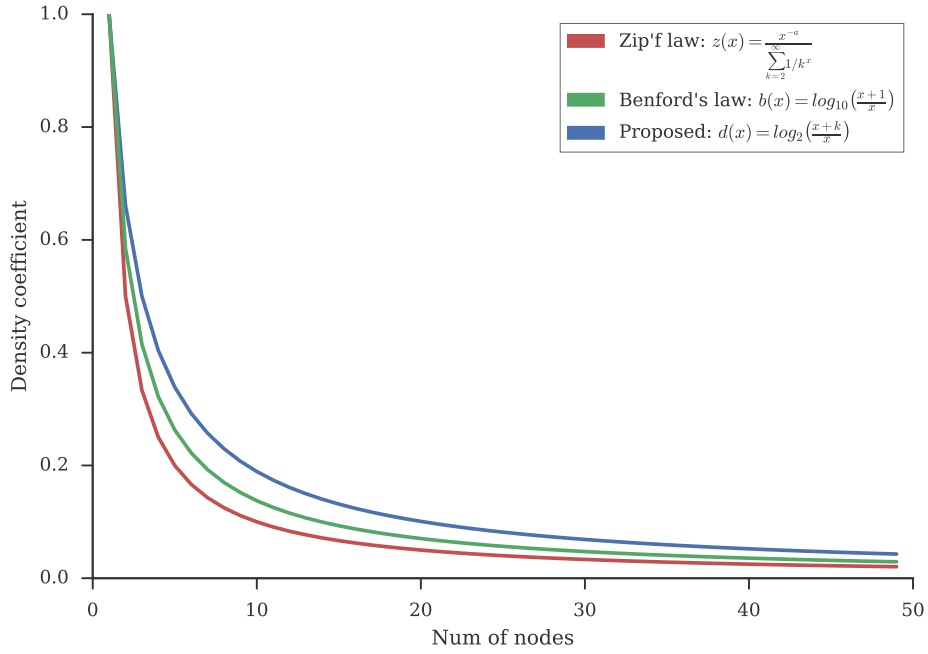


Figure 4.6: Functions to determine expected density coefficients in a sub-graph. We compare the curves for Zipf’s law, Benford’s law, and a proposed function based on Benford’s law. The curve of coefficient values from our adapted version of Benford’s law is not as steep as the others.

Given an island $G' = (V', E')$, in order to avoid *outliers* [127], we also require that all vertices in the island have at least $\log_2(\#V')$ edges for it to be considered a community. Summarizing, we consider an island $G' = (V', E')$ to be a community if:

1. the island is regular
2. $\#V' \geq k$ (where we take $k = 3$)
3. $\#E' \geq \delta(\#V')$
4. no $v' \in V'$ such that $\#n(v') < \log_2(\#V')$

Here, (1) and (2) correspond to the criteria proposed by Zaveršnik & Batagelj [5, 133], whereas we add (3) and (4) to avoid the need for a fixed upper bound K and to correspond with the intuition of a community as discussed previously. Based on these criteria, we select communities from the hierarchy such that they are non-overlapping, but otherwise maximal, while meeting the required criteria. More specifically, we begin at the most general island containing all vertices, and then visit sub-islands, checking that the criteria is met.

Example 4.4. We return to the hierarchy shown in Figure 4.9 where the final communities are highlighted with shaded boxes. These communities were computed by analyzing every island in the hierarchy starting at the top (which corresponds to the bottom of the figure): the first island that we evaluate corresponds to the full graph. We assess the graph using all conditions previously defined, where **Th.** indicates the threshold value, **Ob.** the observed value and **Pass?** whether the condition is satisfied or not.

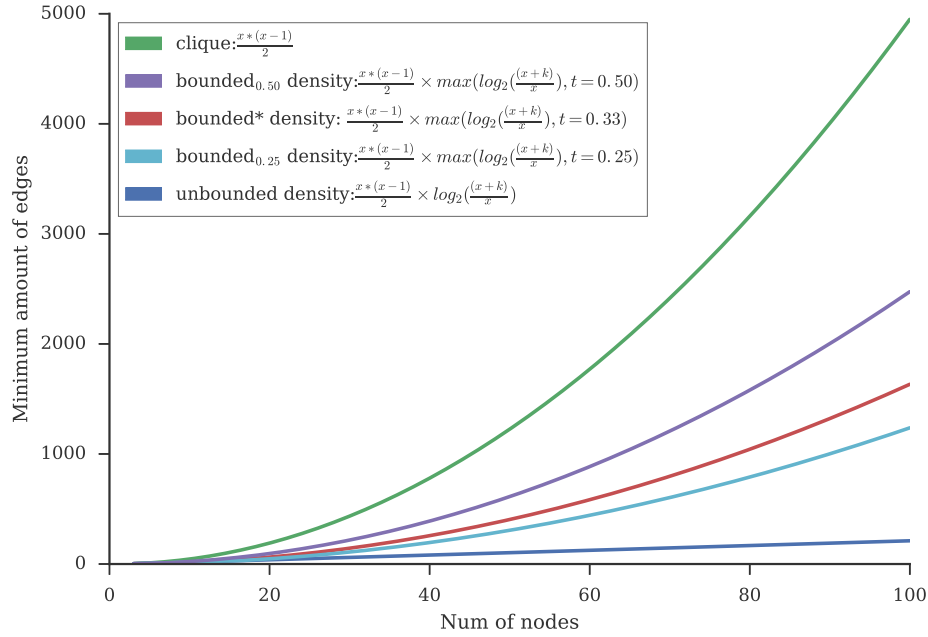


Figure 4.7: Functions to determine minimum number of edges in a sub-graph. The plot shows curves of functions that combine the expected size of clique with a given configuration of our edge density function based on Benford’s law. For our experimental setup, we select the curve “bounded*density”.

	Th.	Ob.	Pass?
REGULAR?	✓	✓	✓
MIN VERTICES (k)	3.0	19	✓
MIN EDGES ($\delta(\#V)$)	56.4	46	✗
MIN CONNECTIVITY ($\log_2(\#V)$)	4.2	3	✗

Since the full graph does not satisfy all conditions, we proceed to analyze its sub-islands independently. We first analyze the left sub-island with the vertices:

$$V' = \{\text{worldseries, mlb, baseball, detroit, series, city, usa}\}$$

	Th.	Ob.	Pass?
REGULAR?	✓	✓	✓
MIN VERTICES (k)	3.0	7	✓
MIN EDGES ($\delta(\#V')$)	10.8	15	✓
MIN CONNECTIVITY ($\log_2(\#V')$)	2.8	3	✓

This sub-island satisfies all conditions, therefore it is accepted as a valid community. This process continues, ultimately resulting in the final set of communities shaded in Figure 4.8. Moreover, Figure 4.9 shows the distribution of the annotations on the actual tag graph. \square

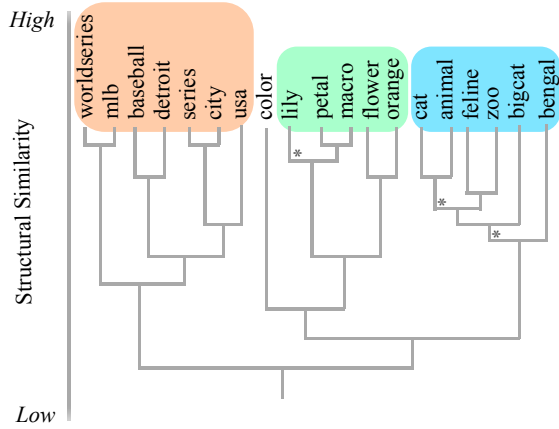


Figure 4.8: Edge island hierarchy showing the final set of topics found for query “tiger”.

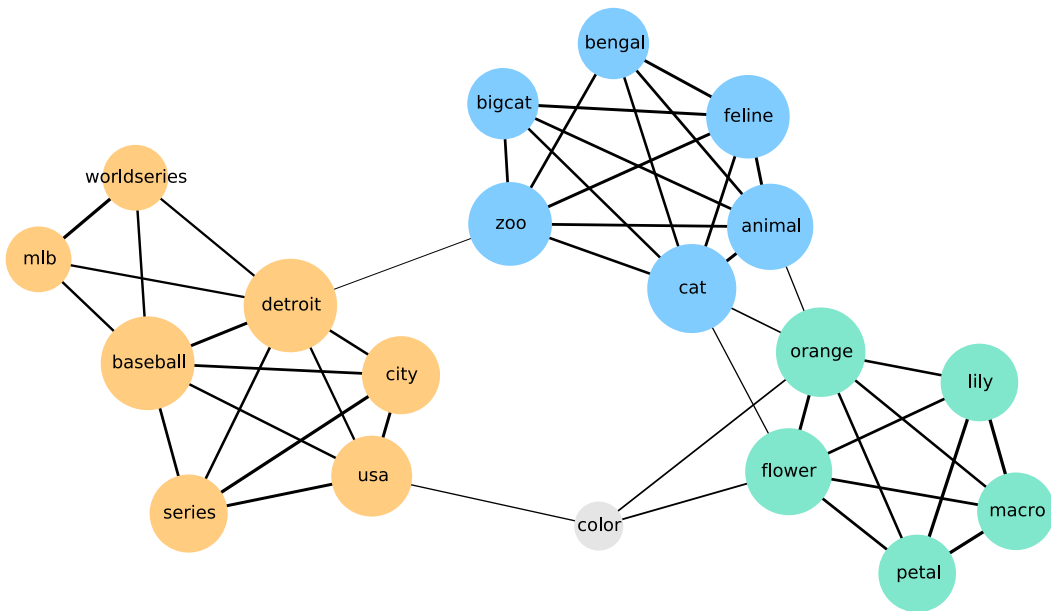


Figure 4.9: Tag graph for query “tiger” colored according to the topics discovered using the edge-based island hierarchy.

Algorithm 3 Algorithm for computing clusters based on a (vertex/edge) island hierarchy.

```

1: procedure ADAPTIVEISLANDCUT( $G, hierarchy, densityThreshold, sizeThreshold$ )
2:    $clusters \leftarrow []$ 
3:   while  $hierarchy \neq []$  do
4:
5:      $island \leftarrow hierarchy[0]$ 
6:      $hierarchy \leftarrow hierarchy - island$ 
7:      $elements \leftarrow get\_elements(island)$ 
8:      $graphlet \leftarrow subgraph(G, elements)$ 
9:      $edgesIsland \leftarrow |edges(graphlet)|$ 
10:     $expectedDensity \leftarrow \log\left(\frac{|elements|+minSize}{|elements|}\right)$ 
11:     $expectedDensity \leftarrow \max(expectedDensity, densityThreshold)$ 
12:
13:     $minEdges \leftarrow \frac{expectedDensity \times |elements| \times (|elements|-1)}{2}$ 
14:
15:     $D \leftarrow node\_attributes(graphlet, degree)$ 
16:     $lowerDegree \leftarrow [(node, degree) \in D \text{ if } degree < \log(|island|)]$ 
17:
18:    if  $edgesIslands \geq minEdges$  and  $|elements| > 2$  then
19:       $clusters \leftarrow clusters \cup elements$ 
20:    else
21:      if  $lowerDegree \neq []$  or  $edgesIslands < minEdges$  or  $island.regular$  then
22:         $hierarchy \leftarrow hierarchy \cup get\_subislands(island)$ 
23:      end if
24:    end if
25:  end while
26:  return  $clusters$ 
27: end procedure

```

4.3 Experiments

In this section we describe the experiments and results of the automatic and user-based evaluation of various well-known methods for community detection, as well as our proposed methods based on adaptive island cuts. First, we describe the dataset and algorithms included in the comparison. Then, we describe an evaluation based on WordNet ontologies and provide the main results and drawbacks. Finally, we present the user study designed to compare the topics detected for the different methods included in the comparison.

4.3.1 Dataset

To assess our method, we need to obtain a dataset that contains a set of query search results in which every element in the list is associated with a set of tags. Social20 [64] is a dataset extracted from Flickr that contains 20 queries, for which the top 1000 results have been re-

Query	Users	Tags	Avg. Tags	Top-5 tags (\neq query)
<i>airplane</i>	430	2,267	9.08	airport, plane, aircraft, flight, aviation
<i>beach</i>	738	2,910	8.03	ocean, sea, sand, water, sunset
<i>boat</i>	661	2,872	9.83	water, sea, ship, lake, boating
<i>bridge</i>	776	5,493	19.51	river, water, night, reflection, hdr
<i>bus</i>	690	5,088	14.18	stop, transit, transport, city, street
<i>butterfly</i>	519	2,197	8.43	nature, flower, insect, macro, 2008
<i>car</i>	619	3,719	11.23	auto, show, 2006, classic, racing
<i>cityscape</i>	645	3,707	15.53	city, building, night, skyline, 2008
<i>classroom</i>	631	3,153	9.97	school, student, teacher, high, class
<i>dog</i>	866	5,721	14.79	pet, animal, puppy, white, the
<i>flower</i>	863	2,898	8.15	macro, nature, garden, yellow, plant
<i>harbor</i>	624	3,940	17.50	harbour, boat, water, lake, bay
<i>horse</i>	169	1,699	16.87	caballo, cheval, paard, pferd, equus
<i>kitchen</i>	769	4,059	10.98	house, home, s, interior, remodel
<i>lion</i>	522	2,582	10.27	zoo, animal, cat, safari, park
<i>mountain</i>	622	2,986	12.26	hiking, snow, nature, lake, landscape
<i>rhino</i>	389	1,898	14.38	zoo, animal, lion, elephant, bird
<i>sheep</i>	557	2,730	9.71	animal, farm, 2008, barn, vermont
<i>street</i>	785	6,061	17.98	city, art, new, night, people
<i>tiger</i>	189	1,448	12.62	sport, ice, icehockey, hockey, csaha

Table 4.1: *Social20* Dataset Summary.

ranked with the state-of-art algorithm for tag ranking, according to the benchmark described by Li et al. [66]. In Table 4.1 we list the queries included in the dataset, and present the main statistics from the dataset.

4.3.2 Algorithms

Given that one of our evaluation frameworks relies on human judgment, we limit our evaluation to two settings, which were chosen based on preliminary experiments:

1. Adaptive Island Cut based on edge property (abbr. AIC-EDGE): With structural similarity as edge property; and $k = 3$, $t = 0.33$ for the density function.
2. Adaptive Island Cut based on vertex property (abbr. AIC-VERTEX): With PageRank (w.r.t. structural similarity weights) as vertex property; and $k = 3$, $t = 0.33$ for the density function.

In order to compare the performance of our community detection algorithm based on adaptive islands cuts, we select a set of well-known algorithms for community detection.

3. Eigenvector [80] (abbr. EIGENVEC) aims to maximize modularity by partitioning the graph based on a *modularity matrix*. After repeated divisions, the algorithm returns a k -partite structure.

4. Infomap [92] (abbr. INFOMAP) uses random walks to emulate information flow in a network. Efficiently coding the information flow is equivalent to finding communities in a graph.
5. Label propagation [90] (abbr. LBLPROP) uses a recursive voting scheme until it reaches a fix-point. First, each vertex is assigned a unique label. In subsequent iterations, each node takes the label that most of its neighbors have, until a fix-point is reached.
6. Multilevel [14] (abbr. MULTILVL) is similar to a hierarchical agglomerative clustering technique. First, vertices for which the modularity gain is maximal are put together in the same community. Next, communities of the graph are re-indexed as if they were vertices, and then edges are re-weighted based on the links between communities.
7. SCAN [127] (abbr. MSCAN) is a variation of the clustering algorithm DBSCAN [32] for graphs. The algorithm defines *cores* (vertices forming a densely-connected neighborhood) as seeds for their clustering process. Core nodes are expanded into their neighborhoods, identifying communities.

Every algorithm in the dataset receives the same data as input, and do not require any additional information nor setup tuning to perform the community detection task. The implementation employ for algorithms (3) to (6) is available in the library *igraph* for the Python programming language³. For our experiments, we used a java implementation of the SCAN algorithm⁴ developed by Papadopoulos et al. [83]. Table 4.2 shows statistics for each algorithm on the communities identified for the tag graphs. After applying the community detection algorithms in our dataset, we see a split of the algorithms into two groups based on an order of magnitude difference in the average community size computed: LBLPROP, EIGENVEC and MULTILVL produce much larger/fewer communities than AIC-*, INFOMAP or MSCAN. In fact, the former algorithms tended to produce one “super-community” with the vast majority of tags, and a few other small communities.

Method	Total	Community Size			
		Max	Min	Avg	S.Dev
AIC-EDGE	649	216	3	7.33	17.94
AIC-VERTEX	533	142	3	6.47	15.27
EIGENVEC	157	542	4	81.91	107.43
INFOMAP	906	79	4	6.54	17.93
LBLPROP	123	1446	4	104.58	4.87
MULTILVL	189	393	4	68.17	82.80
MSCAN	1,269	166	4	7.31	8.24

Table 4.2: Statistics about communities computed by different methods across all 20 queries

Runtimes: Community detection was run on a MacBook Pro, with an Intel Core i7 (3 GHz) processor and 8 GB of RAM. Taking the mean and standard deviation of runtimes of each algorithm across the twenty queries, the fastest were LBLPROP (11.3 ms \pm 10.2), MULTILVL

³Documentation at <http://igraph.org/python/doc/igraph.Graph-class.html> (accessed on June, 2018)

⁴Code available at https://users.dcc.uchile.cl/~tbracamo/mscan_dawak2010/sourcecode.zip

Query	AIC-EDGE	AIC-VERTEX	EIGENVEC	INFOMAP	LBLPROP	MULTILVL	MSCAN
<i>airplane</i>	204.60	46.05	42.00	220.00	2.73	3.13	328.00
<i>beach</i>	1729.70	428.33	116.00	363.00	7.08	9.42	566.00
<i>boat</i>	574.70	100.99	76.30	254.00	4.32	4.29	367.00
<i>bridge</i> ↑	10926.00	1259.62	380.00	1750.00	37.80	35.90	13590.00
<i>bus</i>	3975.00	549.52	209.00	1380.00	20.40	18.50	2423.00
<i>butterfly</i>	561.50	93.83	97.20	164.00	4.40	4.47	403.00
<i>car</i>	2690.20	747.78	61.60	325.00	7.98	12.80	534.00
<i>cityscape</i>	5203.00	683.85	266.00	927.00	17.30	19.60	5376.00
<i>classroom</i>	1197.60	192.02	83.10	489.00	7.65	7.26	773.00
<i>dog</i>	6111.00	776.98	317.00	1740.00	26.70	26.70	4388.00
<i>flower</i>	2166.90	653.30	67.30	268.00	6.69	9.58	658.00
<i>harbor</i>	4034.00	504.38	184.00	1810.00	19.00	19.90	4010.00
<i>horse</i> ↓	78.13	21.40	167.00	89.50	1.41	1.43	171.00
<i>kitchen</i>	1731.60	258.37	193.00	773.00	12.90	11.60	1203.00
<i>lion</i>	510.80	100.14	222.00	324.00	6.11	4.76	469.00
<i>mountain</i>	1991.70	645.86	72.90	362.00	6.75	9.04	865.00
<i>rhino</i>	258.90	54.59	87.90	186.00	2.97	3.23	313.00
<i>sheep</i>	448.20	89.12	148.00	250.00	4.59	4.49	451.00
<i>street</i>	7106.00	763.15	382.00	2210.00	28.10	25.70	9141.00
<i>tiger</i>	121.31	29.07	76.00	98.00	1.80	1.75	204.00
Mean	2581.04	399.92	162.42	699.13	11.33	11.68	2311.65
St.Dev	2880.86	350.31	105.73	683.45	10.22	9.67	3526.64

Table 4.3: Runtime (in ms.) for the algorithms included in the user study. *horse* and *bridge* are the queries with lowest, and highest runtime, respectively. AIC-EDGE and LBLPROP are the algorithms with lowest, and highest average runtime across queries, respectively.

(11.9 ms \pm 9.7), followed by EIGENVEC (162.4 ms \pm 105.7), AIC-VERTEX (399.9 ms \pm 350.3, incl. s-weighting and PageRank computation) and INFOMAP (699.1 ms \pm 683.5). The slowest method was AIC-EDGE (2,581.0 ms \pm 2,880.9, incl. s-weighting), where the worst case was a query that took 10.9 seconds to run; we found that most of this cost came after computing the hierarchy when selecting the islands (which took 9.8 seconds in the worst case). This would be prohibitive for online clustering in particular, but we are not focused on optimizing our methods. However, revising the implementation of the traversal of the edge-island hierarchy in future work is something we can identify as a priority from these results, especially for online scenarios.

4.3.3 User study design

Task: We designed a user evaluation to measure how well community detection algorithms identify topics from the tag graphs in query results. Each user is presented with a set of tags from a cluster that has been created using one of the community detection algorithms. Users are given the option to remove terms that they do not understand. The user is then asked to choose the largest subset of tags that they (subjectively) consider semantically related.

Assessment of Groups of Related Terms

INSTRUCTIONS: Select the terms that you feel belong to a same category or semantic meaning. If you find more than one category or meaning, select the terms that correspond to the largest group only. If one of the term's meaning is not clear click on

You've completed 9 questions.
[Click here for an example](#)

fish
 yacht
 new
 painting
 sailing
 island
 bird

I do not find a relevant concept among the terms that I understand.

[Next](#)

Figure 4.10: Example of task shown to assessors during the user study. For instance, an assessor could select the terms “sailing” and “yacht” to represent the concept **sailing**. The UI allows users to remove terms that they do not know, and therefore cannot judge by clicking on the sign to the left of each terms. To select terms associated with a concept (not explicitly stated), users should click on the terms.

In addition, the user can state that they do not find any of the terms to be related. We randomize the order in which tags are shown to remove any bias due to position. Figure 4.10 shows an example from which users could select different sets of tags based on the concept they infer. Internally, every assessment assigns a category (and value) to each tag in the current set:

- Relevant (1): if the tag is relevant for the detected concept.
- Unknown (0) : if the user does not know the meaning of the tag.
- Not relevant (-1): if the tag is not relevant for the detected concept.

Sample: As we mentioned earlier, some of the resulting clusters contain over a hundred tags, and the complete set of clusters produced by all algorithms is too large for human evaluation. To mitigate this problem, we designed the following sampling method that we applied to the results of all algorithms:

1. For each query we identify the subset of tags assigned to a cluster for each of the algorithms (not all the tags are assigned to clusters). We refer to these tags as the seed set for the query. In addition, we remove tags that have one character and those that have non-ASCII encoded characters.
2. We randomly select 10 terms from the seed set and retrieve the community that they belong to, according to each algorithm.

- For each community we show at most 10 tags for user evaluation. If the community contains more than 10 elements, we first add the tags that appear in the seed set, and then randomly select from the remaining tags until we reach a total of 10. With this sampling approach, we ensure that we are evaluating similar topics for each algorithm.

In Table 4.4, we show some of the characteristics of the sample dataset that is evaluated by users. In total, we are left with 660 communities (17.3% of all communities), of which 633 correspond to unique sets of tags (27 identical communities were identified by more than one algorithm).

Method	Count	Community Size			
		Max	Min	Avg	S.Dev
AIC-EDGE	89	10	3	6.06	2.93
AIC-VERTEX	81	10	3	5.64	3.00
EIGENVEC	66	10	4	9.67	1.27
INFOMAP	167	10	4	6.05	2.20
LBLPROP	39	10	4	9.33	1.69
MULTILVL	86	10	5	9.80	0.87
MSCAN	132	10	4	7.24	2.34

Table 4.4: Statistics about communities sampled for the user study across all 20 queries

Assessors: We recruited 40 students from two engineering schools in Santiago, Chile. Most of the tags in our evaluation were in English; therefore, participants were required to have at least an intermediate level of English (i.e., to normally read and understand news and non-technical books in English). Evaluations were split into 3 sessions. We collected 3,165 evaluations, averaging 79.1 (± 40.4) assessments per user, and 23.8 (± 22.2) seconds per question. We found that 50% of users knew the meaning of at least 90% of the tags in the communities they evaluated, whereas all users knew the meaning of more than 67% of the tags. Figure 4.11 shows the distribution of users versus the rate of terms (in the user study) they knew.

4.3.4 Inter-assessors agreement

A key question is how consistently different users agree on which terms are related given the same question but with the terms in a randomized order. We measure agreement with Krippendorff’s alpha (α) [61] since, unlike other agreement metrics such as the more well-known Fleiss’ kappa (κ) [33], Krippendorff’s α can be computed in cases where some responses are blank (in our case, where users did not understand a term). We compute the coefficient as

$$\alpha = 1 - \frac{D_o}{D_e}$$

where D_o is the observed disagreement, and D_e is the expected disagreement based on an interpretation of chance (see [61] for more details).

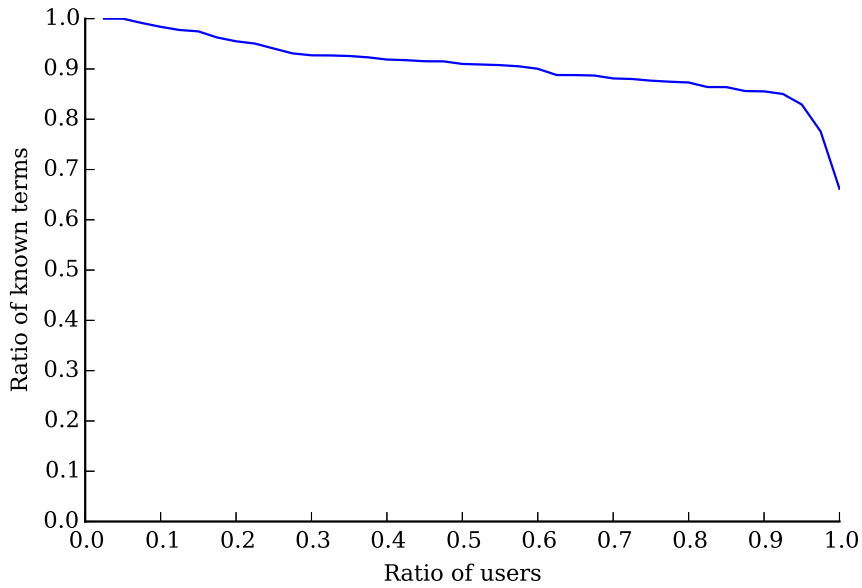


Figure 4.11: Distribution of rates of terms known with respect to rate of assessors. 50% of assessors in our user study knew the meaning of at least 90% of the tags in the communities they evaluated, whereas all users knew the meaning of more than 67% of the tags.

Tag	Unkown	Tag	Unknown
xc3	29	petronas	13
bw	27	d300	13
wm	22	dundee	12
vlinder	21	wasser	12
nyclpc	20	ii	12
pupazzo	19	a38	11
abigfave	18	xaate	11
pferd	18	lca	11
lon	18	xa0	11
s	17	thed	10
hob	16	etsy	10
pecora	16	xa9	10
ca	16	ed	10
vitulina	16	d80	10
s5is	13	kl	10
d200	13	xb6we	10
t	13	350d	10
f	13		

Table 4.5: List of tags marked as unknown 10 or more times.

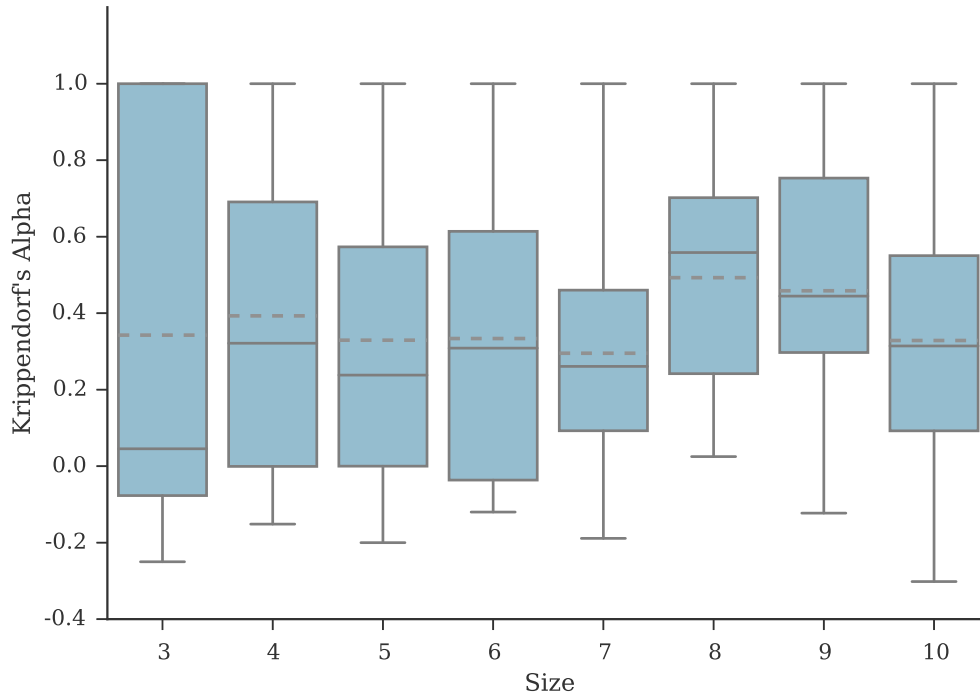


Figure 4.12: Krippendorff’s α values by size of sampled community shown to assessors. Samples with fewer terms have a higher variance of Krippendorff’s α values. Assessors’ agreement is higher for concepts inferred from samples of 7 terms.

We show box-plots⁵ for Krippendorff’s α values distributed by size of communities in Figure 4.12. We notice that for most community sizes the median value of α fall into the range $0.2 < \alpha \leq 0.4$. Agreement is poorer for communities of size 3, presumably because it is more difficult for users to pick up a consistent “theme” with fewer terms. We noted that the mean and median values of Krippendorff’s α do not vary significantly across different methods.

In our user study, participants can indicate if they do not find at least two related terms. In Table 4.6 we show the rate of clusters for which users find at least two related terms. For comparison, we also provide the average size of communities from Figure 4.12, since the probability that a user finds at least two related terms from a set of three is much lower than from a set of ten, even if said terms were selected randomly. Thus, we see an expected correlation between algorithms returning larger communities and having higher rates. We list the most unknown terms in Table 4.5.

⁵The box-plots of this paper are Tukey box-plots where the solid line denotes median, the dashed line denotes mean, box-edges denote quartiles, whiskers denote the lowest/highest observation with 1.5 IQR of the box-edges, and other points denote outliers.

Method	Rate	Avg Size
AIC-EDGE	0.79	6.06
AIC-VERTEX	0.79	5.64
EIGENVEC	0.95	9.67
INFOMAP	0.69	6.05
LBLPROP	0.86	9.33
MULTILVL	0.92	9.80
MSCAN	0.84	7.24

Table 4.6: Rate of communities for which the majority of assessors found at least two related terms

4.3.5 Majority-voting Precision

To compute the majority-voting precision of a set of terms T sampled from a community relative to a single user assessment, we consider each term as: *relevant* (T_R or 1: selected as related), *unknown* (T_U or 0: marked as not understood) and *irrelevant* (T_I or -1 : not selected or no pair of related terms found). We then consider T_R as true positives, T_I as false positives, and discard T_U ; thus we compute the majority-voting precision as:

$$MVP(T) = \frac{\#T_R}{\#T_I - \#T_U}$$

When considering multiple assessors, we apply a majority voting for which terms are relevant, unknown, or irrelevant. Majority-voting precision is calculated on the consensus assessment.

Example 4.5. Let's assume we have the following assessments from users u_1 to u_5 for a set of tags $T = \{t_1, \dots, t_5\}$

	t_1	t_2	t_3	t_4	t_5
u_1 :	1	0	-1	-1	0
u_2 :	1	1	0	-1	0
u_3 :	1	0	0	-1	1
u_4 :	1	1	1	0	0
u_5 :	1	1	1	1	0
<i>consensus</i> :	1	1	1	-1	0

The majority-voting precision is $MVP(T) = \frac{3}{5-1} = 0.75$. □

In Figure 4.13 we show the precision values of all sampled communities for each algorithm. For this figure, we consider all communities assessed in our user study. This means we include communities with low agreement, and also communities for which users do not find a relevant concept. We see that the proposed methods based on Adaptive Island Cuts (AIC-*) perform better on average than those we compare against. Also, the maximum and minimum values

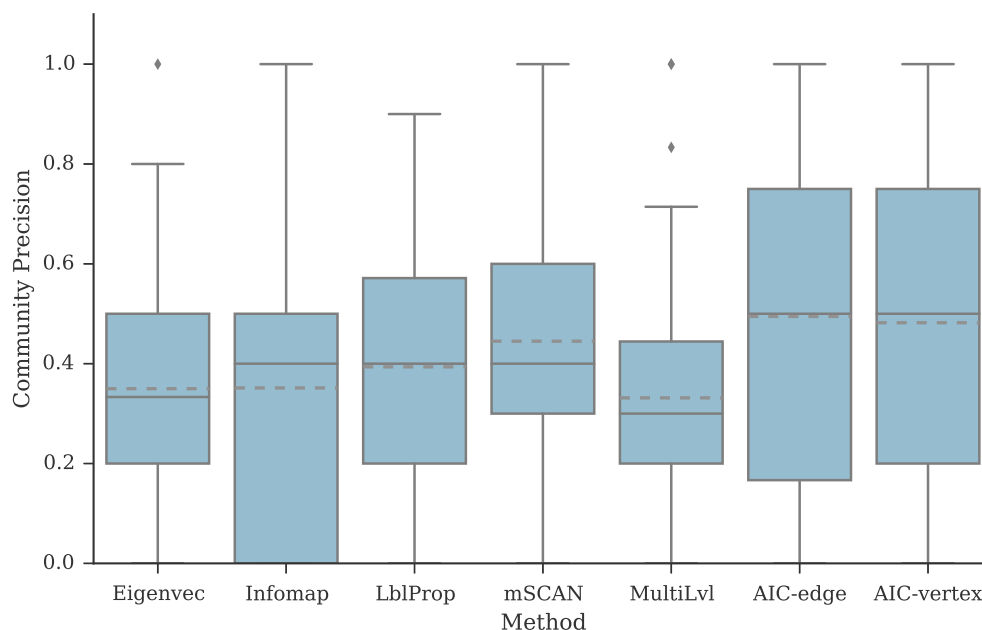


Figure 4.13: Comparison of Majority-voting Precision for the methods considering all sampled communities. AIC-* methods have a higher majority-voting precision than others. mSCAN is the method with second highest majority-voting precision, but its variance is lower than AIC-* methods. This means that communities detected using mSCAN are less likely to include terms that assessors consider irrelevant for the detected concept.

of their quartiles are higher than the values returned by other methods. However, the mean majority-voting precision of all methods is lower than 0.6, which implies that many of the terms in the computed communities are noisy, regardless of the method employed.

Nevertheless, the majority-voting precision measure has some weaknesses. First, it does not distinguish cases where three users found a term (ir)relevant vs. cases where five users found a term (ir)relevant. Second, even if the terms selected as relevant by different users overlap, they may still have different topics in mind; subsequently taking the consensus assessment may then not make sense for any topic. Thus, we performed the same majority-voting precision analysis, but only considering sets of terms with agreement ($\alpha > 0.4$), where we consider 263 (39.8%) of the 660 sampled communities. Figure 4.14 compares the majority-voting precision values for all methods. We notice that after removing low-agreement communities, mean and median majority-voting precision values tend to increase and the quartiles shrink. However, the relative performance of methods tends to stay about the same.

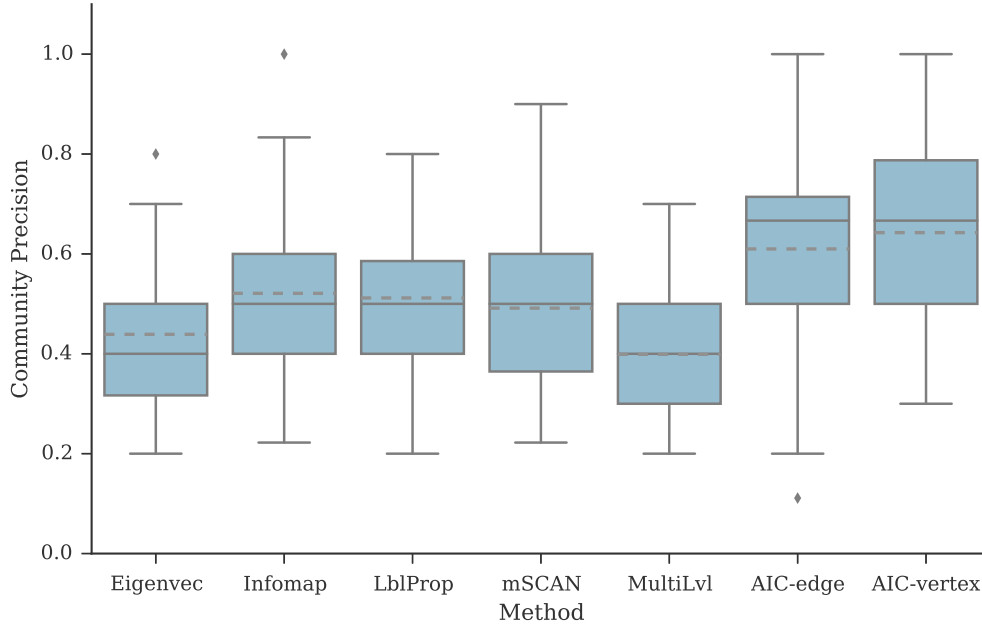


Figure 4.14: Comparison of Majority-voting Precision for the methods considering only sampled communities for which an agreement of $\alpha > 0.4$ (fair agreement) was found. Our proposed methods AIC-* have a higher majority-voting precision than others, and its variance is comparable to other methods. There is a fair agreement that at least 60% of the terms included in the communities detected using AIC-* represent a concept.

4.3.6 Data-driven Recall

An algorithm that creates smaller communities will tend to have higher majority-voting precision, but will also tend to split related terms into different communities. Hence, majority-voting precision only tells one side of the story: we must also consider *recall*. However, measuring true recall appears complex in this setting. Instead we consider a *relative recall* measure, where we take pairs of terms that users agree to be related and then, for each algorithm, we check what ratio of these pairs appear in the same community (*true positives*) versus the ratio of all such pairs (*true positives* and *false negatives*).

To identify pairs of related tags, given a user assessment for a set of tags $T = \{t_1, \dots, t_n\}$, we first use the following function:

$$\text{rel}(t_i, t_j) = \begin{cases} 1 & t_i = 1 \wedge t_j = 1 \\ 0 & t_i = 0 \vee t_j = 0 \vee (t_i = -1 \wedge t_j = -1) \\ -1 & \text{otherwise} \end{cases}$$

where:

1 indicates terms t_i and t_j are related,

0 indicates it cannot be determine if terms t_i and t_j are related or not, and

-1 indicates terms are not related.

Example 4.6. Let us assume we have the following assessments from users u_1 to u_4 for a set of tags $T = \{t_1, \dots, t_4\}$

	t_1	t_2	t_3	t_4
u_1 :	1	0	-1	-1
u_2 :	1	1	0	-1
u_3 :	1	0	0	-1
u_4 :	1	1	1	0

The $rel(\bullet, \bullet)$ values for all the combinations of tags in T are:

	$rel(t_1, t_2)$	$rel(t_1, t_3)$	$rel(t_1, t_4)$	$rel(t_2, t_3)$	$rel(t_2, t_4)$	$rel(t_3, t_4)$
u_1 :	0	1	-1	0	0	0
u_2 :	1	0	-1	0	-1	0
u_3 :	0	0	-1	0	0	0
u_4 :	1	1	0	1	0	0
<i>consensus</i> :	2	2	-3	1	-1	0

In this example, we show the relatedness between pairs of terms in a specific cluster. \square

We then take the sum of this $rel(\bullet, \bullet)$ for all pairs across all user assessments for a specific query and algorithm. To compute the relative recall of a particular algorithm and query, we take the sum of all such pairs for all *other* algorithms and select those with a positive score (> 0) as related pairs by consensus.⁶ We compute the relative recall for that algorithm and query as the ratio of related pairs appearing in the same community vs. all such pairs.

One concern using related terms selected by consensus is again that users may have different topics in mind for why terms are related. To help mitigate this issue, we compute relative recall on a per-query basis. We also compute a weighted version of relative recall where we take the sum of the rel function for all positive pairs appearing in the same community, divided by the sum for all such pairs. This way, we give more weight in the recall measure to pairs that were repeatedly considered related by different users for that query. In the end, both the weighted and non-weighted results were very similar, hence we present only the weighted results.

We present weighted relative recall for each algorithm across all queries as a box-plot in Figure 4.15. Unsurprisingly we see that algorithms producing much larger average community sizes (see Table 4.2) have much higher relative recall: LBLPROP (average community size 104.58), EIGENVEC (81.91) and MULTILVL (68.17), have larger communities and higher relative recall than AIC-EDGE (7.33), AIC-VERTEX (6.47), MSCAN (7.31), and INFOMAP (6.54). On the other hand, amongst the four algorithms producing smaller communities, we see that AIC-EDGE and AIC-VERTEX have better recall (and precision) than MSCAN or INFOMAP.

⁶Unlike typical relative recall measures in IR, we do not include the pairs of the algorithm under testing, since different algorithms may have different numbers of related pairs associated with them, and each algorithm has a recall of 1 for its own pairs.

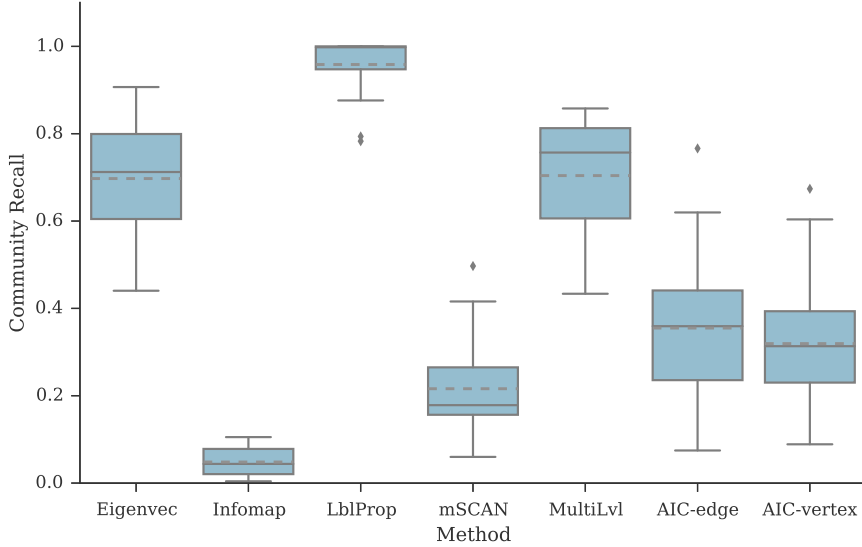


Figure 4.15: Comparison of relative recall for the methods in our user study. Methods that tend to return communities with bigger sizes (EIGENVEC, LBLPROP, and MULTILVL) have higher relative recall values than those that tend to return communities with smaller sizes (INFOMAP, MSCAN, and AIC-*). Proposed methods AIC-* have higher relative recall than other methods that return communities of similar characteristics.

4.3.7 Ontology-based Cohesion

For the automatic evaluation, our goal is to measure how semantically close the terms included in each clusters are. To achieve our goal, we define **cohesion** as our main metric. Cohesion is the average similarity between pairs of different terms in a set. More formally,

$$cohesion_S = \frac{\sum_{t_i \in S} (avg_{t_j \in S, t_i \neq t_j} sim(t_i, t_j))}{n_S}$$

where: S is the set of terms, t_i is a term in the set, n_S is the size of the set.

For our ontology-based cohesion, we measure how similar two terms are using the Wu-Palmer Similarity [125]. Specifically, we employ the implementation of Wu-Palmer Similarity in WordNet. We choose this metric because it is corpus-independent, which means that we do not require to train the similarity measure using a specific corpus. Furthermore, the WordNet implementation of Wu-Palmer allows comparing synsets⁷ from different part-of-speech⁸. Wu-Palmer similarity value is normalized by definition, which makes it easy to compare and aggregate similarity values between different pairs of terms. The Wu-Palmer similarity calculates relatedness by considering the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS (Least Common Subsumer). Given two synsets s_i and s_j , the Wu-Palmer similarity is computed:

$$Wu-Palmer(s_i, s_j) = \frac{2 * depth(lcs_{s_i, s_j})}{depth(s_i) + depth(s_j)}$$

⁷Set of synonyms of a given term.

⁸A category to which a word is assigned in accordance with its syntactic functions. The parts-of-speech recognized by WordNet are noun, adjective, verb, adverb.

		Cohesion	
		High	Low
Coverage	High	(animal, gull, mammal, nature, seal) (britain, england, thames, uk) (backpacking, countryside, district, walking)	(airway, american, international, taxiing) (all, reserved, rights, xa9) (brown, canon, canonpowershot, scary)
	Low	(awesomeshot, animal, flickrdiamond, superplus, wildlife) (busch, buschgardens, gardens, tampa) (747, aeroplane, flight, flughafen, flugzeug)	(cambridge, platinumaward, superaplus, walking) (aula, luz, shadow, spain, ventana)(california, diego, san)

Table 4.7: Example of communities with different degree of coverage and cohesion according to WordNet. Communities with high coverage are the ones for which more than 50% of terms are recognized by WordNet. Similarly, cohesion values above 0.5 are considered high.

This means that $0 < Wu\text{-Palmer} \leq 1$. The Wu-Palmer similarity can never be zero because the depth of the LCS is never zero (the depth of the root of a taxonomy is one). The Wu-Palmer similarity is one if the two input concepts are the same.

The $sim_{WordNet}$ is the similarity measure apply on the *cohesion* equation we defined above. The similarity measure based on WordNet aims to maximize the similarity with respect to the synsets associated with two given terms.

$$sim_{WordNet}(t_i, t_j) = \max_{s_i \in t_i, s_j \in t_j} Wu\text{-Palmer}(s_i, s_j)$$

Vocabulary Coverage: Since the terms in our set are originally annotations from an online social network, we measure the coverage of WordNet dictionary over the vocabulary of the clusters we detect using the algorithms we list in Section 4.3.2. We define **coverage** as follows,

$$coverage_S = \frac{\sum_{t_i \in S} find_{WordNet}(t_i)}{n_S}$$

where $find_{WordNet}(t_i)$ returns 1 iff $t_i \in WordNet$, else returns 0.

Table 4.7 shows examples of communities that have high or low coverage. From these examples we notice that WordNet does not contain terms specific to the Flickr community, such as 'flickrdiamond', and 'wesomeshot'. Also, there are sets of terms that human assessors could identify easily as related, but WordNet taxonomy does not recognize because those sets include multi-term entity names, such as ('california', 'diego', 'san') which refers to the city "San Diego, California".

Correlation between automatic evaluation and user study: Figure 4.17 shows that there is no correlation between the metric proposed for assessing the quality of clusters of terms based on user opinion, and the attempt of automatic assessment performed using

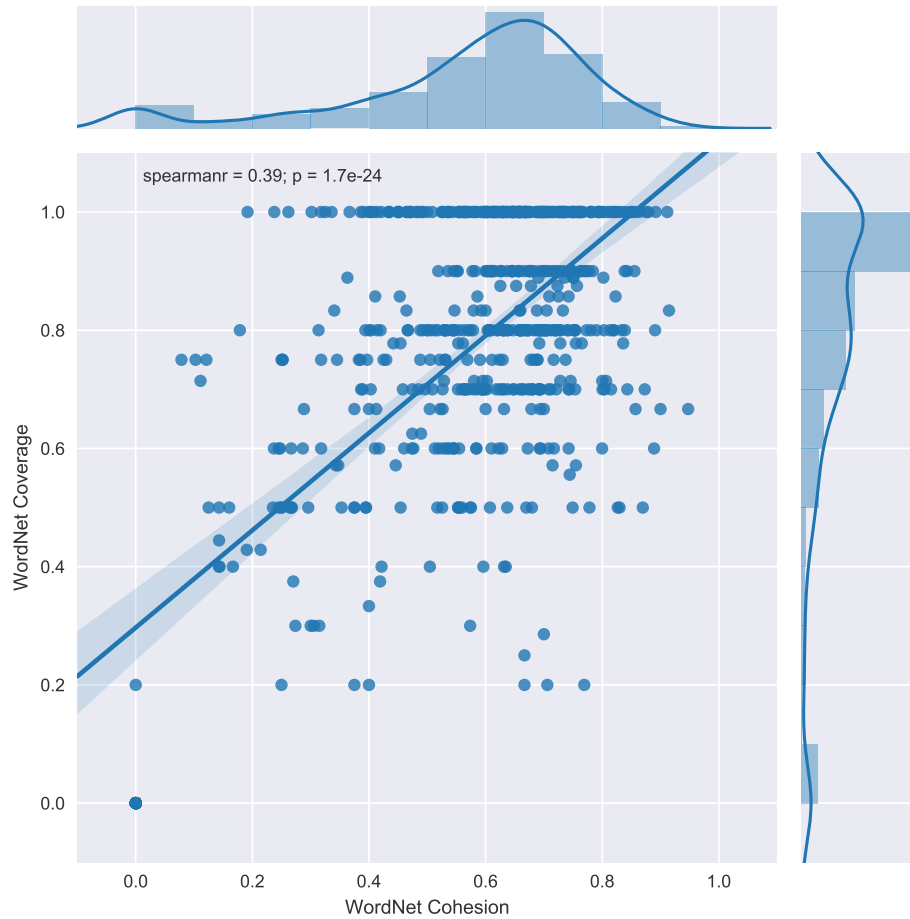


Figure 4.16: WordNet Cohesion vs. WordNet Coverage. There is a low positive correlation between cohesion and coverage values. The majority of communities have a coverage higher or equal than 0.7, and a cohesion between 0.5 and 0.8.

WordNet similarity metrics. WordNet offers different similarity metrics, from which we chose the corpus-independent metric Wu-Palmer. We opted to analyze only the results returned using this metric because all corpus-independent metrics follow similar strategies to compute similarity between terms. From our initial inspection on WordNet results, we noticed that it does not provide a full coverage of the terms. WordNet coverage is lower than the rate of terms assessors recognized in the user study. The same experimental setup with a different source of information would lead to a higher WordNet coverage, which would give more confidence on the similarity values obtained using this resource.

We infer the lack of correlation as a result of the different nature of the metrics employed in both scenarios, and the source of information. In the user study, assessors represent a knowledge base able to effectively identify polysemic terms and to pick the meaning of a term that fits better the current topic. On the other side, WordNet vocabulary is fixed and the list of meanings is bounded to a standard dictionary which does not include slang. Also, many tags are proper nouns, which WordNet does not cover either. Hence, designing an automatic assessment on the cohesion of terms that comprise a concept is not a trivial task.

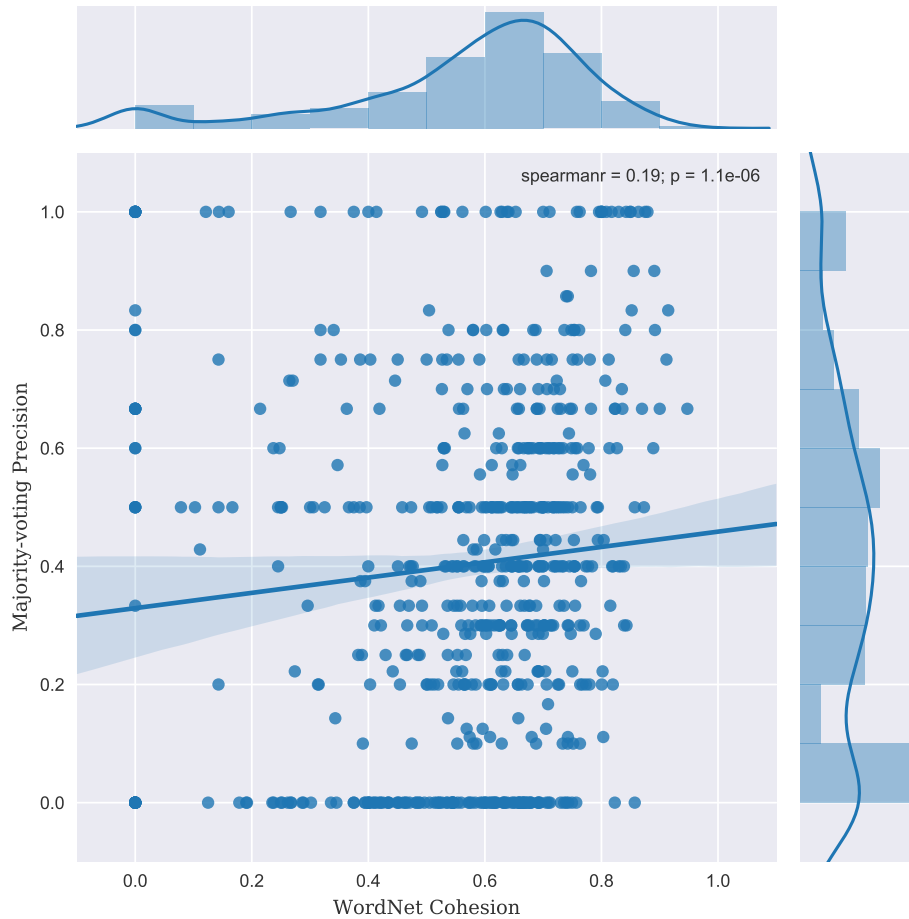


Figure 4.17: Correlation between WordNet Cohesion and Majority-voting Precision. There is no correlation between the Majority-voting Precision computed based on user opinion, and the cohesion values computed using Wu-Palmer implementation in WordNet.

4.3.8 Correlation between User Opinion and Tag Graph

We employ the Spearman’s ρ correlation metric to determine if a correlation exists between the co-occurrence of terms in the search results, the structural similarity of their neighborhoods based on the Tag Graph structure, and the opinion of users obtained from the user study. Table 4.8 shows the correlation values between user opinions, and the Tag Graph structure. Results show that user opinion with respect to pairs of terms is not correlated with the co-occurrence of terms ($\rho = 0.131$, p -value < 0.001), nor with the structural similarity ($\rho = 0.135$, p -value < 0.001). Low correlation between user opinion and term co-occurrence is due to computing user opinion requiring that we aggregate opinions for and against the topical relationship between terms, while term co-occurrence only considers explicit positive opinions. Besides we found that structural similarity and co-occurrence values are highly correlated ($\rho = 0.966$, p -value < 0.001).

		Spearman’s ρ^*	
User Opinion	~ Co-occurrence	=	0.131
User Opinion	~ Str. Similarity	=	0.135
User Opinion	~ Co-occurrence (Prob.)	=	0.146
User Opinion	~ Str. Similarity (Prob.)	=	0.144
Co-occurrence	~ Str. Similarity	=	0.966

* p -value < 1e-6

Table 4.8: Spearman’s ρ correlation values between users’ opinions and Tag Graph properties value. Co-occurrence values and structural similarity are highly correlated because structural similarity values are computed based on co-occurrence. Low correlation between user opinion and metrics from the Tag Graph is because user opinion is obtained by aggregating positive and negative opinions (by majority-voting), while the other metrics only consider positive opinions.

4.3.9 Discussion

One of the benefits of community detection algorithms is that one need not provide a number of expected clusters *a priori*. However, without this fixed criterion, different algorithms can produce very different results; particularly, we found two well-distinguished types of algorithm: three algorithms that produced large communities (avg. size > 68), and four algorithms that produced small communities (avg. size < 8), including the two we propose. From our user study, we found that evaluators had mixed agreement on which tags were related in the provided set. Looking at first at all assessments, and then only to those with agreement ($\alpha > 0.4$), in both cases the two methods we propose had the highest mean precision. Considering relative recall, our methods were beaten by those producing large communities, but our methods outperformed the other two that produce equivalently-sized communities.

In this sense, our results show that for the twenty queries considered, our methods outperform MSCAN and INFOMAP for grouping terms that users consider related into the same communities, both in terms of precision and recall. However, the comparison with large-community methods is inclusive; our methods are better in precision but much worse in recall. Thus, the question of which is better depends on the relative importance of precision vs. recall for the application in mind. We do note, however, that the LBLPROP method has a small cost in precision relative to our methods when considering its gain in recall. When comparing our two proposed methods based on edge-islands and vertex-islands, the latter has a slight edge in precision, but a slight cost in recall. Nonetheless, overall results are quite similar, with the single exception that our current edge-island implementation incurs a significantly higher runtime cost than that of vertex-islands.

4.4 Conclusions

The work presented in this chapter is motivated by the goal of producing a topical clustering of multimedia resources based on tags. We focus on community detection techniques, since there is a variety of established methods proposed in the literature, and they have the significant benefit of not requiring a fixed number of clusters to be provided beforehand. However, without this fixed criterion, different algorithms can produce very different results. We found two well-distinguished types of algorithm: three algorithms that produced large communities (avg. size > 68), and four algorithms that produced small communities (avg. size < 8) including the two we propose.

One major obstacle faced in this work was deciding on appropriate methods for evaluation. Our user evaluation was a costly process in terms of manual effort performed by human assessors. Our evaluation methodology limits the variety of algorithms, configurations, and datasets, which can be considered for assessment. On the other hand, it is not clear how a gold standard for tag clusters could be created *a priori*, particularly when users may often disagree on the relatedness of sets of terms, as we have seen from the results herein. From our user study, we found that assessors had mixed agreement on which tags were related in the presented set.

Query-dependent annotation clustering has some unique benefits: it is applied on smaller graphs, and since it can be applied client-side, it could reduce server load, and could even be used to aggregate results from multiple servers. Also, by only clustering resources relevant to a specific topic, it is possible that the quality of clusters is improved with respect to that topic, given that polysemous tags are more likely be used in the sense captured by the query.

There are still some open questions about which resolution of community detection is the most desirable for clustering multimedia resources. Our results are still inconclusive as to which community size is more useful when clustering search results themselves—whether smaller communities with better precision, or larger communities with better recall. Also, when considering online clustering, an important consideration is the number of results returned by the search engine.

Our next major step is to use the results of our tag-clustering methods to investigate clustering on the level of resources, asking users to evaluate the topic relatedness of image clusters or video clusters rather than the tags with which they are annotated. We also plan to investigate the effectiveness of community detection techniques for identifying topics in other datasets with tagged multimedia search results. As for our current research, we make datasets and evaluations, available for download at: <http://dcc.uchile.cl/tbracamo/communities/>.

Chapter 5

Automatic Tagging of Multimedia Resources

Searching multimedia resources using query-by-keywords is one of the most common tasks users perform on the Web. Although the nature of a query and the content being searched are not directly comparable, users still see query-by-keyword as a natural way to express their information needs. In recent years, search engines have significantly improved the accuracy of retrieved results for multimedia searches, driven by state-of-the-art classification algorithms that reduce the search problem to a labeling approach. This boost in the accuracy of results is also supported by the fact that current search engines count on more data to train better models. Nevertheless, many state-of-the-art algorithms still fail on returning relevant multimedia results for queries that represent complex ideas, as well as novel concepts.

To address this problem, in this thesis we focus on the automatic labeling problem that supports multimedia search. Traditionally, algorithms that automatically label items rely on a curated training dataset. Nevertheless, most public datasets only cover a small subset of the concepts found on the Web. Hence, we propose a framework that leverages user-generated data (a.k.a. context data) as well as audio-visual features of multimedia content, in order to build datasets with high quality that are not restricted to predefined concepts. Specifically, we propose using a graph representation to model content similarity and semantic relationships that exist among multimedia resources found on the Web. Our representation aims to combine seemingly unrelated metrics into a unique graph structure, to support a framework for the automatic tagging of multimedia resources based on information propagation.

The main features of our framework are:

- **Multimedia-type independence:** We do not employ specific content-based features in the graph construction process. Our model could potentially use a single feature, or combine multiple audio-visual features to represent relationship between multimedia resources.
- **Language independence:** Our framework does not require any training data, and it is not fixed to language nor domain.

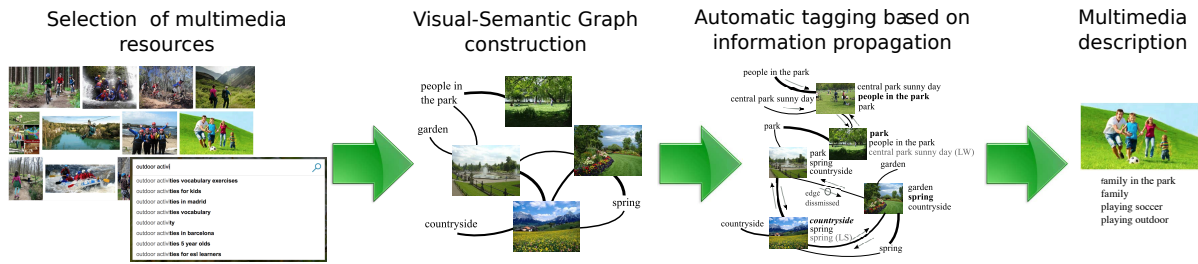


Figure 5.1: Framework for automatic tagging:: Given a set of queries and their clicked images from a search engine log, a Visual-Semantic graph is built, and a tag propagation algorithm is applied over it. Each query is assigned to multiple images based on their audio-visual similarity with respect to the originally clicked images.

- **Multilabel assignment:** Our framework is able to assign more than one label to the same multimedia resource. Furthermore, these labels might be complete descriptions on natural language, instead of sets of one-term labels.

5.1 Framework for Automatic Multimedia Tagging

In this section we describe the framework we propose to perform automatic multimedia tagging based on information propagation over a graph structure. First, we introduce the general framework, pointing out the input and output of each step. In subsequent sections, we describe in detail each step of the framework.

5.1.1 Overview

We introduce a framework to automatically label multimedia resources that have already been indexed by a search engine. Our proposal is motivated by the idea of enhancing search engine results by leveraging user-generated content (UGC), such as click-through data. In Figure 5.1 we show the proposed framework, which consists of the following stages:

1. **Selection and gathering of multimedia+UGC source:** Our starting point is multimedia data that has had some user interaction. For example, videos posted on multimedia sharing platforms and tagged by their publishers, or images indexed by search engines with some associated searches.
2. **Construction of the Visual-Semantic graph:** The Visual-Semantic graph is the key structure over which our framework works. Given a finite set of multimedia resources R , where each resource is represented by an audio-visual feature signature, and a set of labels, the Visual-Semantic graph is the union of the graph that represents the visual similarity between the resources, and the graph that represents the semantic relationship between labels and resources.

3. **Automatic Tagging Algorithm:** We employ an information propagation approach to broadcast labels initially associated with some multimedia resources towards new resources that are visually similar. Under this approach, we do not need additional information to the one represented on the Visual-Semantic graph. Given that initial runs on propagating information show noise propagation (i.e., irrelevant labels are propagated through the graph), we propose a boosting heuristic that exploits the Visual-Semantic graph structure in order to reduce the effect of what we call “*stop images*”.
4. **Multimedia description using tags:** Ideally, it would not be necessary to apply any additional process to the output of the automatic tagging algorithm. However, the labels propagated might contain redundant information, which should be filtered before re-indexing the results in the search engines.

5.1.2 Visual-Semantic graph

The Visual-Semantic graph is built upon two main components: (1) a visual similarity graph, which represents the relationship between images based on visual descriptors; and (2) a semantic similarity graph, which depicts the relationship between images and candidate annotations. In our specific case, we use *click-through* data to define this relationship.

Visual Similarity Graph: The *visual similarity graph* represents content-based similarity relationships in a collection of images. Each image is represented using a visual descriptor, and the similarity between images is computed using a measure δ . In this graph, the nodes and edges represent the images and their similarity, respectively. The weight of the edges is high if the images connected are similar. To normalize the distance between images for a given descriptor, we compute its associated maximum distance M (the largest distance between two images using that descriptor). Also, we define a threshold value τ that indicates which images must be connected.

Finally, we define the visual similarity graph $G_\nu = (I, E)$, where I is the set of images and E is the set of edges of G_ν . An edge $(i, j) \in E$ is defined if $\delta(i, j) \leq \tau$. To each edge (u, v) , we associate a weight $w(i, j) = \delta(i, j)$ that represents the content-based similarity between both images. For our study, we compute the descriptors listed in Table 5.1.2, but our approach is not limited to them.

Semantic Similarity Graph: We define the *semantic similarity graph* as an undirected bipartite graph that represents semantic-based similarity relationships between a collection of term-sets (sets of words) and a collection of images. The nodes in this graph represent two types of objects: term-sets and images. The edges in this graph connect term-sets with images that have a semantic relationship with them. Each edge has a weight associated to it which is a measure of the relevance of the term-set to the connected image.

For this work in particular, we consider the click graph as our semantic similarity graph. The click graph is a bipartite graph of queries and images which denotes user searching behavior extracted from a search engine query log. Therefore, in our bipartite

Visual Descriptor	Description	Size (bytes)	Similarity Measure
Edge Histogram (EHD) [75]	It represents the spatial distribution of the direction of the edges.	80	Manhattan (L_1)
Color Histogram (HSV) [75]	It quantifies the HSV (Hue-Saturation-Value) color space.	256	Manhattan (L_1)
Ordinal Measure (OMD) [11]	It represents the order of the average intensities in descending order.	81	Hamming (H)

Table 5.1: Descriptors employed to represent images in the Visual Similarity Graph.

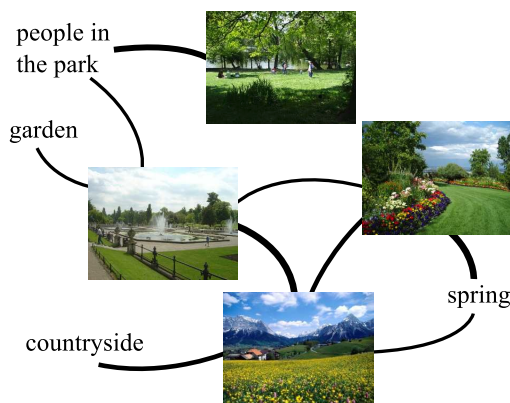


Figure 5.2: Visual-Semantic graph [16]: The edges between queries and images are added based on aggregated data from query logs, and edges between images are added based on their visual similarity.

semantic graph structure, search engine queries are the term-set type nodes and the images clicked by users are the image type nodes. Edges in this graph connect queries to the images selected by users in their searches. The weight of an edge corresponds to the number of clicks that an image registers in a specific period of time for a given query.

It should be noted that other types of user generated annotations of images can be used to generate a semantic similarity graph; for example, terms in metadata, such as user tags. Our selection of the click graph in this case is related to two characteristics which make it appropriate: 1) It gives a measure of relevance of term-sets to images, which is the click frequency, 2) it conveys user-relevance feedback, i.e. users select (click) on images which match their information need.

The Visual-Semantic graph We define the Visual-Semantic graph $G_{v,s}$ as the union of the visual similarity graph and the semantic graph. There is an undirected weighted edge between two images i_1 and i_2 of weight $w(i_1, i_2)$ if both images are similar according to the visual similarity graph. There is an undirected weighted edge between a term-

set t and an image i if there is a user defined semantic relationship between q and i . The weight of this edge is given by $w(t,i)$. In Figure 5.2, we show an example of Visual-Semantic graph.

For simplicity, in this work we have only considered images that register at least one click in the query log. Additionally, for computational complexity reasons, we only consider a random partial cover of the visual similarity graph. We also consider three image descriptors, so in fact we obtain three different similarity graphs. To combine these graphs, we perform a union between each similarity graph and the click graph, achieving three possible unified graphs. This combination will be explained with more detail in our experimental evaluation.

5.2 Characterizing Visual Information Needs

In this section we focus on the potential of the Visual-Semantic graph to address the problem of understanding user visual information needs. We also describe the importance of including the analysis of visual information needs in order to diversify the test cases in the evaluation process.

We define a *User Visual Information Need* as a user requirement to obtain visual information, which is biased by a main physical attribute such as color, texture or shape, or a combination of them. Thus, it is possible to establish a relationship between computational representations of objects selected as relevant, for visual information needs extremely linked to physical attributes. In addition, it is possible to determine whether a request is associated to a visual information need.

We introduce the notion of visual information need in the Visual-Semantic graph structure, based on the distribution of connected components comprised by queries and images. We analyze the distribution of query terms and the neighborhood of images inside each connected component. Intuitively, we establish 4 different categories of queries and images:

Queries with narrow visual information need: Queries related to a set of images with strong visual similarity that belong to the same neighborhood¹ in the Visual-Semantic graph.

Queries with wide visual information need: Queries related to sets of images that belong to different neighborhoods in the Visual-Semantic graph.

Images with accurate textual description: Images related to queries for which neighboring images were also clicked.

Images with loose textual description: Images related to queries for which no neighboring images were clicked.

¹Connected components are referred to as *neighborhoods*.

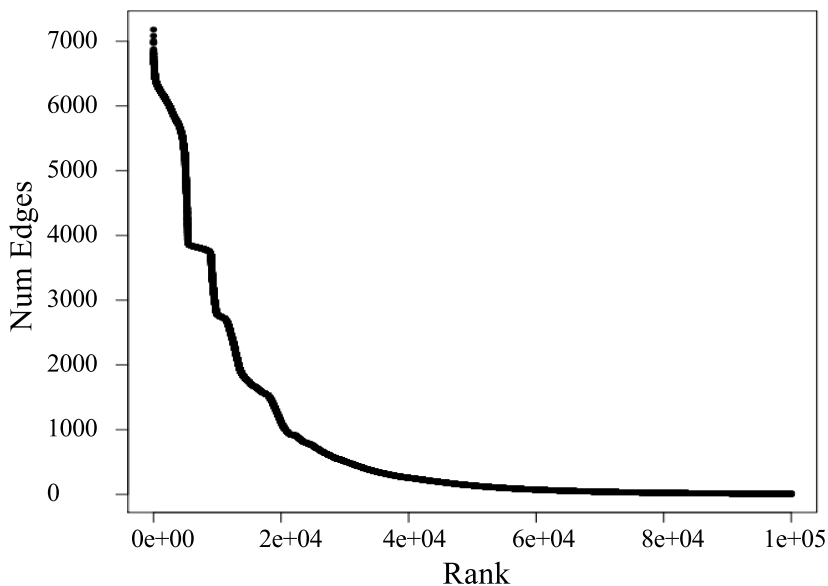


Figure 5.3: Visual-Semantic graph image nodes ranked by connections (edges). The image nodes ranked by amount of edges follow Zipf’s law. We infer that nodes with a high number of edges have a behavior similar to that of “stop-images”, so we define them as “stop-images”.

5.3 Propagating Tags on the Visual-Semantic graph

In this section we describe how to employ the Visual-Semantic graph structure to propagate the semantics of the queries to the images. We use a scheme based on information propagation to automatically tag unlabeled images. We work under the assumption that the queries and their corresponding clicked images are highly related. Thus, we have a strong semantic relationship between a textual description (query) and an image. Note that the relationship between images is not as reliable as the relationship between queries and clicked images.

To tag images, we propagate queries through the Visual-Semantic graph by following a breadth-first traversal. In our scheme, we assume that each query node q is a source of energy ξ which is propagated to image nodes i . We state that at each step of the breadth-first traversal the query q loses an amount of energy. Also, the new energy value is proportional to the weight of the edge that connects the nodes that are traversed. We provide more details of this process in the following sections.

5.3.1 Stop-images

In a previous study [16], we show that every visual descriptor has a Zipf-like distribution of connections between images that appear in the Visual-Semantic graph. Indeed, we define *stop-images* as images within large image neighborhoods. The main characteristic of stop-images is that, similar to stop-words, they do not carry any relevant semantics. Structurally, we can identify stop-images by filtering image nodes with a number of connections higher than a given threshold. We consider that stop-images are likely to propagate noisy tags,

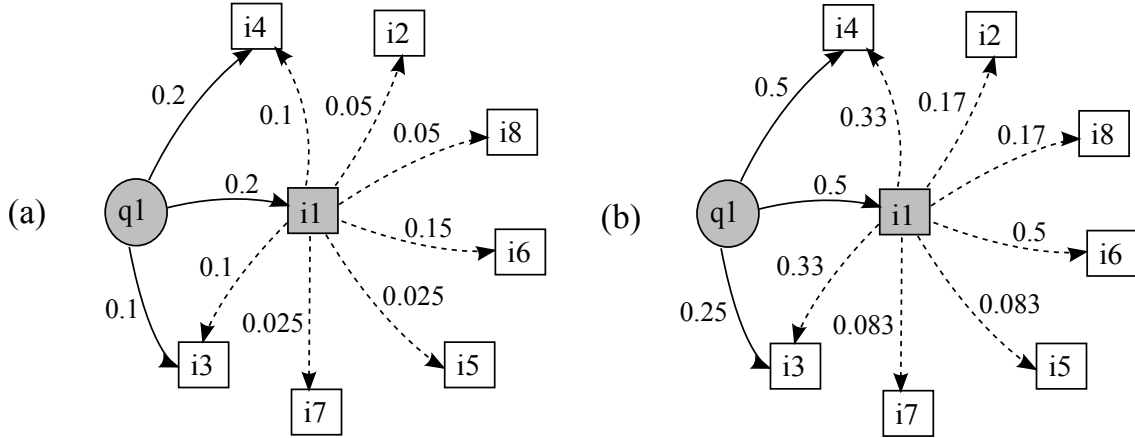


Figure 5.4: Graph pre-processing: (a) original edges, (b) re-weighted edges. Original edge weights are result of stochastic normalization, while re-weighted edges of a node are normalized with respect to the node highest edge weight.

and are in fact the biggest problem in our propagation scheme. Figure 5.3 shows the ranked number of edges in a Visual-Semantic graph. As we can see, there is a huge difference between the number of connection from images in the first 10% and images in the last 10% of the curve.

5.3.2 Weighting schema and pruning indicators

Since the original weight of a nodes' output edges is stochastic-normalized [86], there is an inversely proportional relationship between the maximum weight of the output edges of a node, and the number of output edges of a node. We re-weight the edges of the graph in order to reduce the effect of edge weight minimization in nodes with many connections.

Let $w_{i,j}$ and $w'_{i,j}$ be the original and new weight, respectively, of the edge that connects nodes i and j . $w'_{i,j}$ is inversely proportional to the max value of all output edges of node i ($w_{i,k}$), and proportional to the sum of all the output edges of node i . Formally, each node's output edge is normalized using:

$$w'_{i,j} = \frac{w_{i,j} \cdot \sum w_{i,k}}{\max w_{i,k}}$$

This equation maintains the idea that the edges' weights are probabilities and it adds weight independence to edges. Weight independence is required to propagate tags uniformly. Figure 5.4 shows the application of our re-weighting scheme. Once we have re-weighted the Visual-Semantic graph, we perform our method.

Since our approach focuses on providing an approximation to the automatic tagging problem, in our algorithm we define two *pruning variables* which allow us to filter edges which do not contribute to a good tag assignment.

Edge probability threshold(α): This variable allows to determine the number of nodes connected to a current node that are suitable for the next propagation level. A low

value is not recommended for this variable, to avoid propagating tags between images with low similarity. Since the Visual-Semantic graph’s edges represent the normalized jumping probability from one node to another, this probability does not reflect an accurate similarity metric. We cannot assure that a higher probability implies higher similarity, but a higher similarity metric can assure higher probability.

Tagging energy threshold(ε): The Visual-Semantic graph contains cycles so that a stop condition is necessary to finish the breadth-first traversal our algorithm requires. This variable represents the minimum energy value required to propagate a tag from the current node to its descendants. The value of ε is inversely proportional to the propagation path length. Therefore, lower values of ε imply longer paths.

To reduce the noise generated by the propagation of tags through stop-images, we propose a weighting scheme based on two aspects: (1) the likelihood of two nodes representing the same semantics, and (2) the similarity of two nodes based on their neighborhoods. For the first aspect, we denormalize the stochastic weight between two nodes, so the probabilities do not decrease while the number of neighbors of a node increases. For the second aspect, we use the structure similarity [83] to measure the percentage of neighbors shared by two nodes. We combine these assumptions in the following equation:

$$\psi_{i,j} = \frac{\omega_{i,j}}{\max_k \omega_{i,k}} * \frac{|nv(i) \cap nv(j)|}{\sqrt{|nv(i)| * |nv(j)|}}$$

where:

$\omega_{i,j}$ and $\psi_{i,j}$ are the original and new weight of the edge that connects nodes i and j ;
 $\max_k \omega_{i,k}$ is the maximum value of all output edges of node i ;
 $nv(x) = n(x) \cup x$; and $n(x)$ is the set of neighbors of x .

In Figure 5.5-5.7, we show the distribution of the mean weight according to the degree of nodes. We notice that for the HSV (Figure 5.6) and OMD (Figure 5.7) descriptors there is a crosspoint between the curves of the original weight and the new weight. We define the crosspoint value as the maximum degree for nodes with information suitable to be propagated (i.e. not stop-images). We observe that the structural similarity influences the weight of the edges, by increasing the value of nodes with similar neighbors and penalizing the weight of edges from potential stop-images.

5.3.3 Bounded propagation of tags

Once we have re-weighted the Visual-Semantic graph, we execute our proposed method, which consists on traversing the graph. This traversal process starts at each query and is independent of other queries’ traversal processes. Our method can be easily parallelized since the propagation of every query is independent. For each query q , we propagate it to its connected images i , passing as much energy as the edge’s weight. Let, $t_{q,i}$ be the energy of tag q with respect to an image i , then the first propagation step consists of:

$$t_{q,i} = w'_{q,i}$$

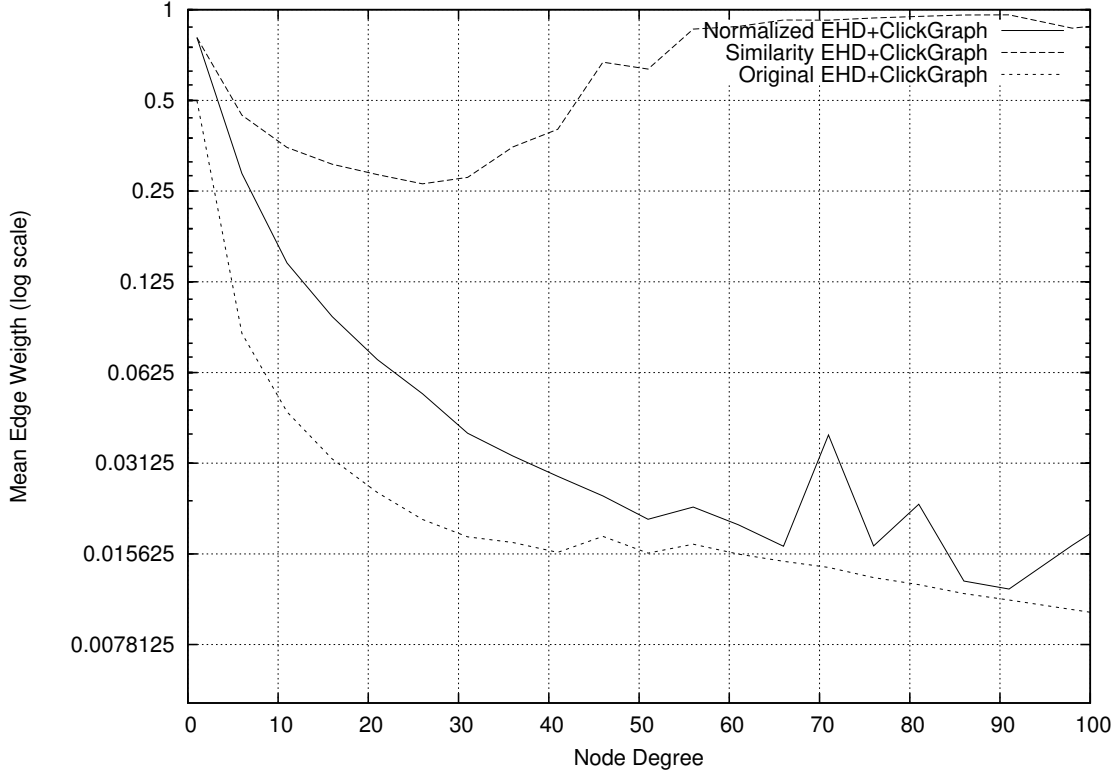


Figure 5.5: Weight comparison for EHD descriptor. The dash curve on the bottom represents the original edge weights for the Visual-Semantic graph that uses EHD. The dash curve on the top represents the structural similarity between image nodes. The line in the middle represents the final weights of the edges connecting images that result from multiplying the original weight and the structural similarity.

Then, we propagate the tag q associated to an image j to all its neighbors i . We choose only the neighbors that fulfill the *edge connectivity threshold* (α). If there are nodes that have been already tagged with q , we keep the one with the maximum value. Formally, each image i is tagged with q using the following expression:

$$t_{q,i} = \max(t_{q,j} \cdot w'_{j,i}, t_{q,i})$$

We repeat this propagation scheme following a breadth-first traversal until the energy of the tag associated to an image does not reach the *tagging energy threshold* (ε). The variables α and ε are employed to prune the graph's traversal path. Figure 5.8 shows the propagation process performed by our algorithm. Tags marked with LS are discarded because they have already been assigned to the target images with an energy value higher than the one in the current propagation step. Additionally, tags marked with LW are not propagated a step further, because their energy value is lower than the threshold ε .

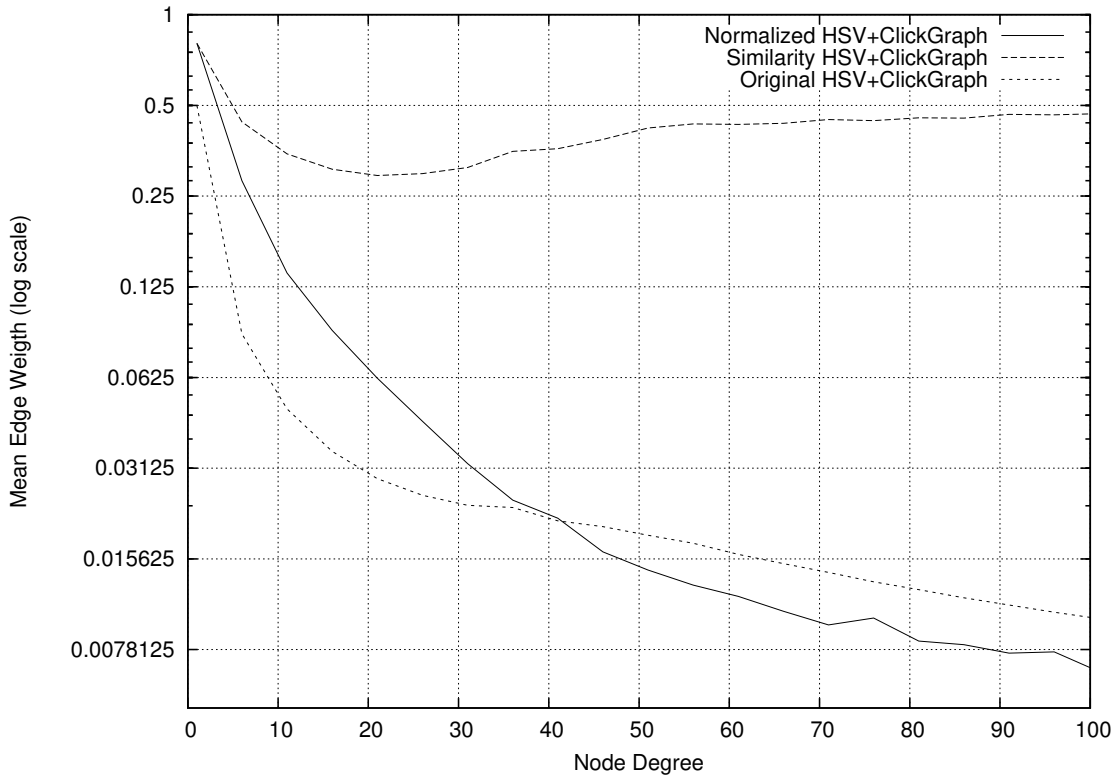


Figure 5.6: Weight comparison for HSV descriptor. The dash curve on the bottom represents the original edge weights for the Visual-Semantic graph that uses HSV. The dash curve on the top represents the structural similarity between image nodes. The line in the middle represents the final weights of the edges connecting images that result from multiplying the original weight and the structural similarity. The cross-point between the curves of the original and the normalized weights represents the threshold number of edges from which an image is considered a “stop-image” for the HSV descriptor.

5.4 Experiments

In this section we describe the exploratory analysis performed to determine whether the initial algorithm proposed to propagate tags and the respective pruning variables had an actual impact on the quality of the information propagated with respect to a naïve approach based on breadth-first traversal. The goal of our initial exploratory analysis is assessing the descriptiveness of the tags generated by our algorithm against a baseline approach. Specifically, we aim to measure the precision² of the queries automatically assigned to multimedia resources.

5.4.1 Dataset

To assess the performance of our proposed solution inspired by the Visual-Semantic graph, we use the following dataset:

²We define Precision as the rate of relevant queries over the total amount of queries associated with an image.

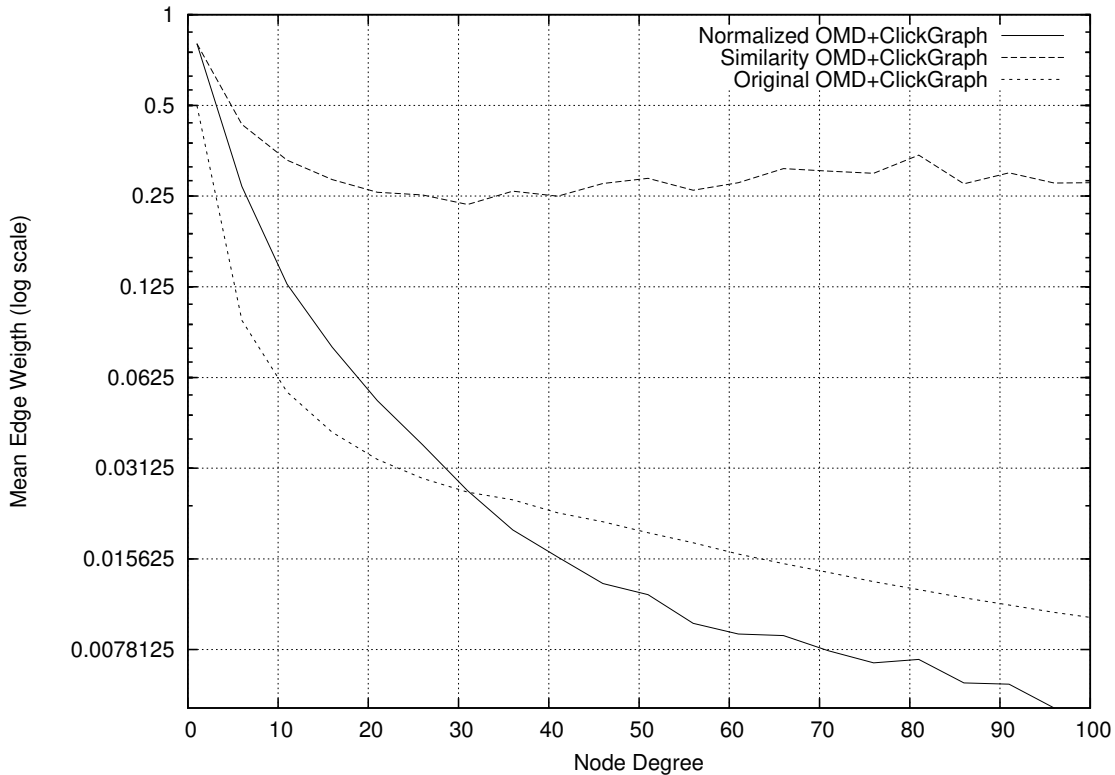


Figure 5.7: Weight comparison for OMD descriptor. The dash curve on the bottom represents the original edge weights for the Visual-Semantic graph that uses OMD. The dash curve on the top represents the structural similarity between image nodes. The line in the middle represents the final weights of the edges connecting images that result from multiplying the original weight and the structural similarity. The cross-point between the curves of the original and the normalized weights represents the threshold number of edges from which an image is considered a “stop-image” for the OMD descriptor.

- Yahoo! Image Search query log: This collection contains a two-weeks period of activity, from March 1st 2010 to March 13 2010. In addition, each week contains approximately 7 million unique clicked images (images clicked in at least one session) and 11.2 million unique-session clicks on images. For further details on the original collection we refer to Poblete et al. [86].

We compute the number of connected components and the bucket sets in order to understand how the queries and images distribute along our graph structure. In Table 5.2, we show the total number of connected components of the Visual-Semantic graph and its sub-graphs (OMD graph and ClickGraph). We also show the total number of non-trivial components for each part of the graph. In the visual graph, a trivial connected component has only one element (image). And in the semantic and Visual-Semantic graphs, a trivial component has only two elements (one image and one query).

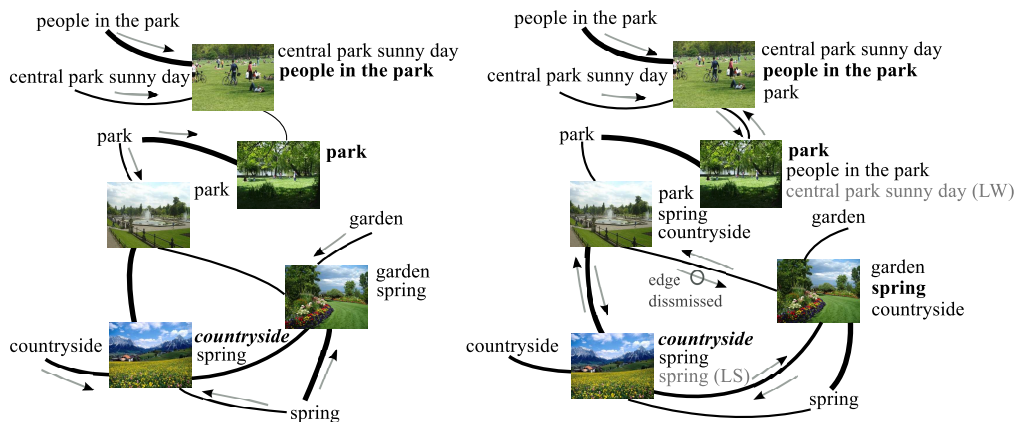


Figure 5.8: Automatic tag propagation on the Visual-Semantic graph: The image on the left shows the result of the first step, in which queries are passed to their corresponding clicked images. The image on the right shows the second step, in which the energy maximization, and energy threshold (ε) validations are applied (LS: propagated tag does not maximize energy for target image, LW: energy value of propagation is lower than the energy threshold). Edges with a weight lower than the edge probability threshold (α) are dismissed.

Graph	Connected Components	
	Total	Non-Trivial
Semantic Graph (ClickGraph)	1,994,335	660,028
Visual Graph with EHD	7,097,172	3,364
Visual Graph with HSV	7,018,763	13,710
Visual Graph with OMD	6,926,696	35,237
Visual-Semantic graph (EHD+ClickGraph)	1,979,629	652,731
Visual-Semantic graph (HSV+ClickGraph)	1,926,628	625,622
Visual-Semantic graph (OMD+ClickGraph)	1,866,981	598,779

Table 5.2: Number of connected components in the Visual-Semantic graphs and the independent Visual graphs, and Semantic graph. The number of connected component decreases after filtering out trivial components, which correspond to graphs with number of nodes less than 3. The ratio between trivial and non-trivial components in Visual graphs gives an insight of the sparseness of such graphs.

5.4.2 Sample selection

Manual validation consisted in: (1) extracting a sub-graph from the Visual-Semantic graph, (2) applying our algorithm over some queries of this graph, and (3) evaluating the main outcomes in the case of study.

First, we select an image from the dataset as a seed and extract the queries directly related to the image. Then, we extract the images connected to the seed and also extract the queries related to these images. Based on visual similarity, we extend the maximum path length between neighbors to up to three nodes. In order to select a seed that allows a representative case of study, we randomly select an image connected with up to other five images. Our goal is to extract and assess the annotations from a cluster of content-related images.

We choose a subset of queries from the extracted sub-graph. We select content-related and content-unrelated queries with respect to the content of the seed image. We perform our propagation algorithm using these queries as starting points. We aim at testing the efficiency of our pruning criteria and the soundness of the propagated tags.

5.4.3 Automatic Tagging Exploratory Analysis

Table 5.3 summarizes the Precision values obtained when evaluating the accuracy of the tags associated to a sample of images, after executing our algorithm and the naïve approach. First, we notice that the pruning variables have an immediate effect improving the Precision of the tags assigned. Also, we notice that as the values assigned to the pruning variables increase, the effect of these variables in the Precision measure decreases. The maximum Precision value our approach achieves is about 0.8. Even increasing the pruning of the propagation of tags set we do not improve the Precision.

We notice that increasing α improves Precision quicker than increasing ε . However, ε provides good improvements when the value of α is small. As a matter of fact, these pruning variables represent a suitable complement for effectively propagating tags in a graph structure. In our propagation algorithm, α controls the width of the spanning tree resulting from the breadth-first traversal, and ε controls the depth of the propagation paths.

Variables	$\varepsilon = 0.125$	$\varepsilon = 0.1875$	$\varepsilon = 0.25$
$\alpha = 0.25$	0.398	0.415	0.488
$\alpha = 0.375$	0.526	0.717	0.810
$\alpha = 0.5$	0.730	0.798	0.799
<i>Baseline</i>	0.357		

Table 5.3: Precision values of the baseline algorithm vs. our propagation algorithm with different combination of values for variables α and ε . The highest Precision value correspond to $\alpha = 0.375$ and $\varepsilon = 0.25$. Increasing α improves Precision quicker than increasing ε . The baseline algorithm consists on a 3-step breadth-first propagation.

5.5 Conclusions

The framework described in this chapter is motivated by the goal of providing long natural-language descriptions to multimedia resources. Our framework is flexible enough to support annotations on different multimedia data types, as well as using different sources for multimedia context data. We specifically develop the use case of using queries to tag images found on the Web. We use click-through data from query logs as a knowledge repository for describing the relationship between descriptions and images.

In addition to our framework, we introduce an automatic tagging algorithm based on tag propagation over the Visual-Semantic graph. Our method considers that queries from query logs are good candidate annotations for the images that users clicked for such queries.

The propagation scheme starts on the semantic section of the Visual-Semantic graph, and propagates queries to the visual section based on visual similarity. We perform an initial exploratory analysis to determine the impact of the pruning criteria applied on our propagation scheme.

We conclude that the use of good pruning criteria improves the quality of the results. Therefore, the values of α and ε are important for the efficiency of our algorithm. We propose a re-weighting scheme that simplifies the generalization of the variable values, depending on the visual similarity measure used to build the graph. We believe that the Precision values obtained can be improved by using a combination of visual descriptors to determine edges between images, as opposed to only one descriptor. Additionally, the combination of descriptors could contribute to smoothing the long tail distribution we found in the number of image edges.

As a result of the analysis of query logs and their conversion to Visual-Semantic graphs, we found that queries and images connect to each other following different trends based on the specificity of their semantics. For example, queries with broad semantics pointing to abstract ideas are prone to be related to images that are not visually similar, and hence disconnected in the Visual-Semantic graph. On the other side, very specific queries tend to be related to neighboring images in the Visual-Semantic graph. An open question around this topic is how to use the graph structure to automatically determine which class of semantics (broad or specific) are carried by queries and images.

One of the main challenges to properly assess our proposed algorithm for propagating context data on the Visual-Semantic graph is determining the optimal configuration setup of pruning variables, as well as a suitable visual descriptor. Currently, the descriptor with better performance is OMD, which is a global descriptor that has been mainly used to detect near duplicates. Nevertheless, we have not explored the potential of local descriptors to build the visual part of the graph. We believe that despite the higher cost of computation, local descriptors are a good option, given that the Visual-Semantic graph is meant to be built off-line.

Besides determining the optimal configuration for our algorithm, another main challenge is assessing the performance of our proposal with respect to other approaches that also address the problem of generating long descriptions for multimedia resources (similar to dense annotations). For the assessment procedure, it is important to design a methodology that includes user opinion, as well as selecting a diverse set of images and queries (with different levels of specificity). Thus, the methodology would allow to compare approaches under different scenarios and also determine which approach would be better in each case.

The next major step to advance in our research is to design a human-centered assessment, and to investigate state-of-the-art approaches for generating dense annotations in order to design a methodology that supports assessments of large collections using only a representative sample. Furthermore, we plan to analyze other datasets to determine common behaviors between multimedia content and context data that can be exploited to improve the generation of long descriptions for multimedia resources.

Chapter 6

Conclusions and Future Work

In this PhD. thesis we address two important aspects related to the challenge of retrieving multimedia documents on the Web. First, we propose *a framework to detect concepts associated to multimedia-related queries* based on community detection techniques. Second, we propose *a framework to combine content and context data to automatically tag multimedia resources* using a graph-based structure. We only assess the frameworks using images. In addition to the main goal, we *designed and deployed a user study that allows us to assess the quality of concepts* obtained from the first framework application.

This section provides a discussion of the main results obtained in this thesis work. In addition, we detail the limitations of the proposed solutions, pointing out the impact of each limitation on our research. Possible ways to overcome these limitations are described in future work.

6.1 Main Results

As we stated early on this document, this thesis is focused on achieving two main goals and a secondary goal, which we managed to accomplish in the following fashion:

- **Goal 1:** *Identify concepts in multimedia search results (Chapter 4)*

We leveraged manually-generated annotations to discover concepts derived from a specific query, by proposing a framework that takes a set of multimedia resources \mathcal{M} with some context data, in order to build a graph of co-occurrences of terms (a.k.a. *Tag Graph*). This framework then mines the structure of the Tag Graph to get a set of groups of terms \mathcal{C} that represents the concepts derived from \mathcal{M} , in which each concept c is represented by a set of terms extracted from the context data.

The main advantage of our framework is its flexibility, because it is language and multimedia type independent. Furthermore, it exploits the idea of islands [133] as a community detection algorithm, which is a completely different approach to those found

on the state-of-the-art. Using islands as a basis also provides flexibility with respect to the metric employed to describe the relevance of each vertex or edge in the Tag Graph.

We demonstrate that by using our framework, it is possible to extract ad-hoc topics for search results using community detection. In particular, our proposed island-based methods produce more compact and less noisy clusters, as well as a lower relative recall when compared to methods that produce much larger clusters. Nonetheless, when we compared against methods of similar clustering resolution, our approach yielded the highest recall.

Therefore, our proposal is able to effectively find coherent groups of terms that describe different topics related to a given query. Although our assessment includes only annotated images, our methodology for processing data and assessing the results can be applied to other multimedia types, as well as other sources of contextual data.

Unlike several state-of-the-art approaches oriented to detect relevant concepts in multimedia resources, our framework does not require knowing the full multimedia collection, since we do not perform any pre-computation over the full dataset. Our framework operates under the assumption that the results obtained for a given query are representative for the most relevant concepts related to the query.

- **Goal 2:** *Effectively annotate multimedia documents using queries (Chapter 5)*

We leverage the relationship between multimedia content and context data to generate relevant annotations to multimedia resources. We propose a framework that exploits implicitly user-generated content (e.g., click-through data) as context data, along with visual features extracted from images on the Web. Specifically, we use the structure of the Visual-Semantic Graph [86] to propagate context data through edges that connect visually similar images.

Our framework is characterized by its flexibility, which allows to use different types of multimedia features to represent similarity between multimedia documents. For our assessment, we used only images to build Visual-Semantic graphs. However, we detected that not all images are good candidates for propagation of information given specific features; we refer to these images as *stop-images*. In order to make our framework more robust, our automatic tagging algorithm discards these stop-images, so that they do not propagate noisy information across the graph. Although our first approach to detect stop-images was strictly empirical, using images' neighborhood information may lead to detected stop-images. The definition of stop-images, as well as the mechanism to detect them, are novel, and could be employed to boost other well-known approaches for automatic tagging.

Moreover, unlike most current automatic tagging approaches, our framework is not focused on annotating multimedia resources with single-term tags. Instead, we aim to generate large descriptions similar to dense annotations. In our preliminary assessment, we found that our proposed Visual-Semantic graph information propagation algorithm outperforms a naïve approach based on breadth-first graph traversal. The difference in performance is obtained thanks to the application of our pruning variables, which

reduces the propagation of noisy descriptions (i.e., queries) between images in the Visual-Semantic graph.

As a result of our analysis of query logs, we learned that queries provide a natural way to describe image content in a concise fashion. Nevertheless, not all queries carry the same level of specificity, nor are all images descriptions bounded to a small set of concepts. On the query-side, we found that queries pointing to visually diverse images are prone to represent abstract concepts, whereas queries related to visually similar images are prone to represent specific physical entities. Regarding the multimedia-side, specifically images, we found that images with broad semantics are usually related to a large set of terms, while images with very specific semantics are related to queries whose terms are likely to overlap. As far as we know, there’s no previous work that presents this taxonomy to classify visual information needs.

- **Goal 1.A:** *Design and deploy a human-centered evaluation (Chapter 4)*

We focus on the evaluation of concepts extracted from a given query. Specifically, we aim at gathering the opinion of users regarding sets of terms that are potentially related to each other and that may represent a concept. Given the complexity of the task, we designed a controlled user study following guidelines from crowdsourcing. For instance, we assigned the questions in a random order, and also the terms were sorted differently for different users. Furthermore, every task was assigned to multiple users, so we could gather data about user agreement. We found that this task is not so easy for humans, which leads to a low inter-assessor agreement.

Given that the amount of clusters returned by the algorithms in our evaluation is in the order of hundreds, and that we aim to gather more than a single opinion per cluster, we design a sampling schema to select comparable clusters returned by different algorithms. Unlike previous human-centered evaluations aiming to measure the quality of automatically generated clusters of terms, we do not ask users to manually simulate the best partitioning of a set of terms. Based on our experience, this could have led to an even lower agreement value.

It is important to remark that we could easily apply the same methodology in other domains, given that our assessments are not bound to a specific multimedia type (i.e., we do not show any image or video that could biased the opinion of users with respect to a concept), and we only rely on the expressiveness of the terms representing concepts. In such a case, which is similar to what we do in our user study, we would only need to instruct users about the resources from which the terms were taken, without listing the queries we employed to gather them. Furthermore, to aggregate the opinion of multiple users, we propose to use metrics inspired by well-known Precision and Recall.

Besides performing a human-centered evaluation, we designed an automated assessment, which led to inconclusive results. We noticed that automated computation of coherence between sets of terms does not correlate with human assessors’ opinions. With current state-of-the-art knowledge bases, such as WordNet, mapping set of terms to concepts is not feasible to assess, since most knowledge bases are not constantly updated with the latest vocabulary and slang popular between Web user communities.

Overall, in this PhD. thesis we show that multimedia context data plays an important role for enhancing the quality of multimedia description. In addition, it becomes clear that graph structures are key when developing techniques that seamlessly represent the relationship between multimedia content and context. By using graph representations instead of vector representations, we manage to bridge the curse of dimensionality.

6.2 Limitations

Despite the results we obtained for the topics covered in this thesis work, this research still suffers from some limitations, which can be inferred by the assumptions that make feasible the assessment of our proposed frameworks for concept detection and automatic tagging.

Regarding our framework for *identifying concepts on multimedia search results*, we assume there is a large size of search results (for example, 1,000) available to build our Tag Graph. Although this gives a broader view of the different concepts we could find related to a query, it does not simulate the amount of results common users would inspect on their daily requests to any search engine. Note that we do not explore different setups to determine the optimal amount of search results needed to build a relevant Tag Graph. Additionally, for our *community detection algorithm based on islands*, our assessments only show results in which we employ similarity structure and PageRank as edge and vertex properties, respectively. We do not study the impact of other centrality metrics for graphs.

Apart from the limitations of our framework and algorithm for concept detection in multimedia search results, it is important to mention that clustering multimedia documents using the concepts found is out of our scope. The main reason for this distinction is that, although our community detection algorithm returns non-overlapping communities, this does not assure that the multimedia results will be partitioned without overlaps. Mapping multimedia documents to concepts may generate one-to-many relationships for cases in which a multimedia document is associated to terms from different concepts.

Related to our framework for *annotating multimedia documents using queries*, we do not consider the effect of coordinated click-spam attacks on the performance of our automatic tagging algorithm. However, we assume that multimedia resources that have been clicked only once are not relevant and thus are not included in the dataset. Moreover, in the last stage of our framework, in which we have a set of queries that describes a multimedia document with a certain level of confidence, we do not remove near-duplicate descriptions. Although removing duplicate information could improve the presentation and interpretation of our results. Nevertheless, these redundant descriptions could be used as input data for more advanced deep learning algorithms able to provide a more complex description of the multimedia documents.

The main limitation of our framework on the annotation of multimedia documents is that our propagation algorithm only considers a phase in which we propagate information over the full Visual-Semantic graph. We still need to extend our algorithm to support small incremental changes in the Visual-Semantic graph. Under the current scenario, our framework

would not be the best option to perform on-line automatic tagging, since it has been initially designed for batch processing of queries and multimedia documents.

Finally, with respect to the *human-centered evaluation designed and deployed* to test multimedia-related tasks, we focused on deploying a user-study for assessing the quality of the clusters of terms obtained for a given query. We do not study the same task in an open environment, such as a real crowdsourcing platform. Thus, we do not know the impact on the complexity of this task in an environment prone to click-spam. Besides, in our assessment we do not rank users by reliability, which means that the opinion of every user is considered equally accurate regardless of their expertise. Deploying our assessment framework in environments such as Amazon Mechanical Turk or CrowdFlower might require an additional step to validate that the user is competent enough to understand the task and complete the assessment.

6.3 Future Work

Future work involves researching how our frameworks for multimedia context data enrichment impact the performance of search engines, measured in terms of Precision and User Engagement. We aim to develop the architecture of a meta-searcher that provides more accurate results, showing them in a more friendly fashion that does not expose users to overwhelming amounts of multimedia.

To reach that long-term goal, we believe it is necessary to address the following challenges:

About concept discovery

- Determining the optimal number of search results required to build a representative Tag Graph for a given query.
For this thesis, we assume that for every query in the dataset, there is a fixed large amount of items returned by a search engine available to build a Tag Graph. In a scenario in which we would have to gather query results in an online fashion, gathering a large amount of search results is not adequate. Thus, it becomes relevant to determine optimal amount of items to retrieve in order to build a Tag Graph that represents relevant concepts in a query without requesting unnecessary elements.
- Determining the optimal resolution of clusters to represent concepts.
In our comparison of community detection approaches, we detect that most algorithms follow one of two trends: (1) returning many small communities, or (2) returning a few large communities. For assessment purposes, we sample communities to get summarized versions that avoid that our assessors get unmanageable lists of annotations.
- Designing an algorithm to map multimedia documents to the concepts related to a query.
The annotations related to a single document in the search result list may be distributed in different concepts. The concepts detected do not contain overlapping terms, but a single multimedia content may belong to multiple concepts, based on its annotations.

About automatic annotations

- Designing a complementary mechanism to annotate small batches of multimedia documents requiring minimum re-computation of the Visual-Semantic graph.
The framework we propose for automatic tagging assumes that we know in advance the full dataset to be tagged. Thus, the Visual-Semantic graph is built once, and then used to propagate candidate labels to multimedia content. However, extending the dataset with small batches of multimedia content has not been explored on this thesis. Given that Web users are continuously publishing new content on the Web, it is extending the algorithm becomes relevant, as well as formalizing a mechanism to annotate multimedia content included after the first Visual-Semantic graph has been built.
- Using ImageNet as a complementary source to build the Visual-Semantic graph.
Query logs are a valuable source of data, and in this thesis work we exploit them in order to build the Visual-Semantic graph that we use to automatically propagate tags. Nonetheless, access to query logs is restricted, and sometimes only partial snapshots are available. Thus, it is necessary to employ additional public resources that allow us to boost query logs when they are limited to the general public.
- Designing an algorithm to generate long descriptions based on queries.
Creating verbose descriptions of images is a field that has gained more adepts in the last few years. In this thesis, we explore the potential of queries to describe multimedia content. Using visual similarity to propagate queries allows images to be tagged with a diverse vocabulary, which could be employed to describe many different aspects of the same document based on the aggregated perception of multiple users.

The most challenging aspect of future work is exploring other modalities of multimedia data different to images.

Bibliography

- [1] Omar Alonso and Matthew Lease. Crowdsourcing 101: Putting the wisdom of crowds to work for you. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 1–2, New York, NY, USA, 2011. ACM.
- [2] Ioannis Antonellis, Hector Garcia-Molina, and Jawed Karim. Tagging with queries: How and why? In *Second ACM International Conference on Web Search and Data Mining WSDM 2009, Late Breaking Results Session*. Infolab, February 2009.
- [3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM press New York, 1999.
- [4] Lamberto Ballan, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Data-driven approaches for social image and video tagging. *Multimedia Tools and Applications*, 74(4):1443–1468, Feb 2015.
- [5] Vladimir Batagelj, Natasa Kejzar, Simona Korenjak-Cerne, and Matjaz Zaveršnik. Analyzing the structure of U.S. patents network. In *Data Science and Classification*, pages 141–148. 2006.
- [6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.
- [7] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated Tag Clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW'06*, 2006.
- [8] Richard Bellman. *Dynamic programming*. Courier Corporation, 2013.
- [9] Frank Benford. The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78(4):551–572, 1938.
- [10] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.
- [11] D. N. Bhat and S. K. Nayar. Ordinal measures for image correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):415–423, Apr 1998.
- [12] Henk M Blanken, Henk Ernst Blok, Ling Feng, and Arjen P Vries. *Multimedia retrieval*.

Springer, 2007.

- [13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [14] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008, 2008.
- [15] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: Model and applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08*, pages 609–618, New York, NY, USA, 2008. ACM.
- [16] Teresa Bracamonte and Barbara Poblete. Automatic image tagging through information propagation in a query log based graph structure. In *Proceedings of the 19th ACM international conference on Multimedia, MM '11*, pages 1201–1204. ACM, 2011.
- [17] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18):3825 – 3833, 2012. The {WEB} we live in.
- [18] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002.
- [19] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan Vranic. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.
- [20] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, January 2012.
- [21] Stefano Ceri, Alessandro Bozzon, Marco Brambilla, Emanuele Della Valle, Piero Fraternali, and Silvia Quarteroni. *Web information retrieval*. Springer, 2013.
- [22] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1 of *HLT '11*, pages 190–200, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [23] Xiangyu Chen, Yadong Mu, Shuicheng Yan, and Tat-Seng Chua. Efficient large-scale image annotation by probabilistic collaborative multi-label propagation. In *Proceedings of the international conference on Multimedia, MM '10*, pages 35–44, New York, NY, USA, 2010. ACM.
- [24] Yuxin Chen, Nenghai Yu, Bo Luo, and Xue-wen Chen. iLike: integrating visual and textual features for vertical search. In *Proc. of the 18th International Conference on Multimedia, MM '10*, pages 221–230, New York, NY, USA, 2010. ACM.
- [25] Jaeyoung Choi, Bart Thomee, Gerald Friedland, Liangliang Cao, Karl Ni, Damian

- Borth, Benjamin Elizalde, Luke Gottlieb, Carmen Carrano, Roger Pearce, and Doug Poland. The placing task: A large-scale geo-estimation challenge for social-media videos and images. In *Proc. of the 3rd ACM Multimedia Workshop on Geotagging and Its Applications in Multimedia*, GeoMM '14, pages 27–31, New York, NY, USA, 2014. ACM.
- [26] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall ii: Query expansion revisited. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 889–896, Washington, DC, USA, 2011. IEEE Computer Society.
- [27] W Bruce Croft, Donald Metzler, and Trevor Strohmann. *Search engines*. Pearson Education, 2010.
- [28] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval. *ACM Comput. Surv.*, 40(2):1–60, 2008.
- [29] Maaike de Boer, Klamer Schutte, and Wessel Kraaij. Knowledge based query expansion in complex multimedia event detection. *Multimedia Tools and Applications*, 75(15):9025–9043, 2016.
- [30] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
- [31] Carsten Eickhoff, Wen Li, and Arjen P. de Vries. Exploiting user comments for audio-visual content indexing and retrieval. In *Proc. of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, pages 38–49, Berlin, Heidelberg, 2013. Springer-Verlag.
- [32] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. AAAI Press, 1996.
- [33] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.
- [34] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.
- [35] R. Fournier-S'niehotta, P. Rigaux, and N. Travers. Querying music notation. In *2016 23rd International Symposium on Temporal Representation and Reasoning (TIME)*, pages 51–59, Oct 2016.
- [36] Bin Gao, Tie-Yan Liu, Tao Qin, Xin Zheng, Qian-Sheng Cheng, and Wei-Ying Ma. Web image clustering by consistent utilization of visual features and surrounding texts. In *Proc. 13th Annual ACM International Conference on Multimedia*, MM '05, pages 112–121, New York, NY, USA, 2005. ACM.

- [37] Shenghua Gao, Zhengxiang Wang, Liang-Tien Chia, and Ivor Wai-Hung Tsang. Automatic image tagging via category label and web data. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1115–1118, New York, NY, USA, 2010. ACM.
- [38] Jonathan Gemmell, Andriy Shepitsen, Bamshad Mobasher, and Robin D. Burke. Personalizing Navigation in Folksonomies Using Hierarchical Tag Clustering. In *Data Warehousing and Knowledge Discovery (DaWaK)*, pages 196–205, 2008.
- [39] Andrew Gilbert, Luca Piras, Josiah Wang, Fei Yan, Emmanuel Dellandrea, Robert Gaizauskas, Mauricio Villegas, and Krystian Mikolajczyk. Overview of the imageclef 2015 scalable image annotation, localization and sentence generation task. In *CLEF (Online Working Notes/Labs/Workshop)*, 2015.
- [40] Alexandru Lucian Ginsca, Adrian Popescu, Bogdan Ionescu, Anil Armagan, and Ioannis Kanellos. Toward an estimation of user tagging credibility for social image retrieval. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 1021–1024, New York, NY, USA, 2014. ACM.
- [41] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821, 2002.
- [42] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision*, 106(2):210–233, 2014.
- [43] C. Gui, J. Liu, C. Xu, and H. Lu. Web image retrieval via learning semantics of query image. In *Proc. of the IEEE International Conference on Multimedia and Expo*, ICME '09, pages 1476–1479. IEEE, 2009.
- [44] Alan Hanjalic, Christoph Kofler, and Martha Larson. Intent and its discontents: The user at the wheel of the online video search engine. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 1239–1248, New York, NY, USA, 2012. ACM.
- [45] Bernhard Haslhofer, Robert Sanderson, Rainer Simon, and Herbert van de Sompel. Open annotations on multimedia web resources. *Multimedia Tools and Applications*, 70(2):847–867, 2014.
- [46] Christoph Hölscher and Gerhard Strube. Web search behavior of internet experts and newbies. *Computer Networks*, 33(1–6):337 – 346, 2000.
- [47] Jeff Howe. Crowdsourcing: A definition. <http://www.crowdsourcing.com/>.
- [48] Dong-Hyuk Im and Geun-Duk Park. Linked tag: image annotation using semantic relationships between image tags. *Multimedia Tools and Applications*, 74(7):2273–2287, 2015.
- [49] Bogdan Ionescu, Adrian Popescu, Mihai Lupu, Alexandru L Ginsca, and Henning Müller. Retrieving diverse social images at mediaeval 2014: Challenge, dataset and

- evaluation. In *MediaEval 2014 Workshop*, 2014.
- [50] Vidit Jain and Manik Varma. Learning to re-rank: query-dependent image re-ranking using click data. In *Proc. of the 20th International Conference on World Wide Web, WWW '11*, pages 277–286, New York, NY, USA, 2011. ACM.
- [51] Bin Jiang, Jiachen Yang, Zhihan Lv, Kun Tian, Qinggang Meng, and Yan Yan. Internet cross-media retrieval based on deep learning. *Journal of Visual Communication and Image Representation*, pages –, 2017.
- [52] Di Jiang, Kenneth Wai-Ting Leung, and Wilfred Ng. Query intent mining with multiple dimensions of web search data. *World Wide Web*, 19(3):475–497, 2016.
- [53] Lu Jiang, Shoou-I Yu, Deyu Meng, Teruko Mitamura, and Alexander G. Hauptmann. Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ICMR '15*, pages 27–34, New York, NY, USA, 2015. ACM.
- [54] J. Johnson, L. Ballan, and L. Fei-Fei. Love thy neighbors: Image annotation by exploiting image metadata. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4624–4632, Dec 2015.
- [55] Anitha Kannan, Simon Baker, Krishnan Ramnath, Juliet Fiss, Dahua Lin, Lucy Vanderwende, Rizwan Ansary, Ashish Kapoor, Qifa Ke, Matt Uyttendaele, Xin-Jing Wang, and Lei Zhang. Mining text snippets for images on the web. In *Proc. of the 20th International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 1534–1543, New York, NY, USA, 2014. ACM.
- [56] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [57] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE, 2004.
- [58] D. Kelly, S. Dumais, and J.O. Pedersen. Evaluation challenges and directions for information-seeking support systems. *Computer*, 42(3):60–66, 2009.
- [59] Christoph Kofler, Martha Larson, and Alan Hanjalic. User intent in multimedia search: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 49(2):36:1–36:37, August 2016.
- [60] Beate Krause, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Logsonomy - social information retrieval with logdata. In *Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia, HT '08*, pages 157–166, New York, NY, USA, 2008. ACM.

- [61] Klaus Krippendorff. Reliability in content analysis. *Human Communication Research*, 30(3):411–433, 2004.
- [62] Clement H. C. Leung, Alice W. S. Chan, Alfredo Milani, Jiming Liu, and Yuanxi Li. Intelligent social media indexing and sharing using an adaptive indexing search engine. *ACM Trans. Intell. Syst. Technol.*, 3(3):47:1–47:27, May 2012.
- [63] S. Li, S. Purushotham, C. Chen, Y. Ren, and C. C. J. Kuo. Measuring and predicting tag importance for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.
- [64] X. Li, C. G. M. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322, Nov 2009.
- [65] Xirong Li, Cees G. M. Snoek, and Marcel Worring. Unsupervised multi-feature tag relevance learning for social image retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '10*, pages 10–17, New York, NY, USA, 2010. ACM.
- [66] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees G. M. Snoek, and Alberto Del Bimbo. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Comput. Surv.*, 49(1):14:1–14:39, June 2016.
- [67] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees G. M. Snoek, and Alberto Del Bimbo. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Comput. Surv.*, 49(1):14:1–14:39, June 2016.
- [68] Dong Liu, Xian-Sheng Hua, Meng Wang, and Hong-Jiang Zhang. Image retagging. In *Proceedings of the international conference on Multimedia, MM '10*, pages 491–500, New York, NY, USA, 2010. ACM.
- [69] Dong Liu, Shuicheng Yan, Yong Rui, and Hong-Jiang Zhang. Unified tag analysis with multi-edge graph. In *Proceedings of the international conference on Multimedia, MM '10*, pages 25–34, New York, NY, USA, 2010. ACM.
- [70] D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999*, pages 1150–1157, 1999.
- [71] Mathias Lux, Christoph Kofler, and Oge Marques. A classification scheme for user intentions in image search. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10*, pages 3913–3918, New York, NY, USA, 2010. ACM.
- [72] Lianze Ma, Lin Lin, and Mitsuo Gen. A pso-svm approach for image retrieval and clustering. In *41st International Conference on Computers and Industrial Engineering*, pages 629–634, 2012.

- [73] Konstantinos Makantasis, Anastasios Doulamis, and Nikolaos Doulamis. A non-parametric unsupervised approach for content based image retrieval and clustering. In *Proceedings of the 4th ACM/IEEE International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream, ARTEMIS '13*, pages 33–40, New York, NY, USA, 2013. ACM.
- [74] Anupama Mallik, Hiranmay Ghosh, Santanu Chaudhury, and Gaurav Harit. Mowl: An ontology representation language for web-based multimedia applications. *ACM Trans. Multimedia Comput. Commun. Appl.*, 10(1):8:1–8:21, December 2013.
- [75] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, Jun 2001.
- [76] Pierre-Alain Moëllic, Jean-Emmanuel Haugeard, and Guillaume Pitel. Image clustering based on a shared nearest neighbors approach for tagged collections. In *ACM International Conference on Image and Video Retrieval (CIVR)*, pages 269–278, 2008.
- [77] André Mourão and Flávio Martins. Novamedsearch: A multimodal search engine for medical case-based retrieval. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 223–224, 2013.
- [78] Apostol (Paul) Natsev, Alexander Haubold, Jelena Tešić, Lexing Xie, and Rong Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proceedings of the 15th ACM International Conference on Multimedia, MM '07*, pages 991–1000, New York, NY, USA, 2007. ACM.
- [79] Mark EJ Newman. Power laws, pareto distributions and zipf’s law. *Contemporary physics*, 46(5):323–351, 2005.
- [80] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.
- [81] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On Spectral Clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.*, pages 849–856, 2001.
- [82] S. Papadopoulos, C. Zigkolis, G. Tolia, Y. Kalantidis, P. Mylonas, Y. Kompatsiaris, and A. Vakali. Image clustering through community detection on hybrid image similarity graphs. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2353–2356, Sep 2010.
- [83] Symeon Papadopoulos, Yiannis Kompatsiaris, and Athena Vakali. A graph-based clustering scheme for identifying related tags in folksonomies. In *Data Warehousing and Knowledge Discovery (DAWAK)*, pages 65–76, 2010.
- [84] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

- [85] Georgios Petkos, Symeon Papadopoulos, Vasileios Mezaris, and Yiannis Kompatsiaris. Social event detection at mediaeval 2014: Challenges, datasets, and evaluation. In *MediaEval 2014 Workshop*, 2014.
- [86] Barbara Poblete, Benjamin Bustos, Marcelo Mendoza, and Juan Manuel Barrios. Visual-semantic graphs: using queries to reduce the semantic gap in web image retrieval. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, pages 1553–1556, New York, NY, USA, 2010. ACM.
- [87] Mohan Ponnada and Nalin Sharda. Model of a semantic web search engine for multimedia content retrieval. In *Proc. 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS'07)*, pages 818–823. IEEE Computer Society, July 2007.
- [88] Duangmanee Putthividhya, Hagai Thomas Attias, and Srikantan S. Nagarajan. Topic regression multi-modal Latent Dirichlet Allocation for image annotation. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 3408–3415, 2010.
- [89] X. Qian, X. S. Hua, Y. Y. Tang, and T. Mei. Social image tagging with diverse semantics. *IEEE Transactions on Cybernetics*, 44(12):2493–2508, Dec 2014.
- [90] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.
- [91] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proc. of the International Conference on Multimedia, MM '10*, pages 251–260, New York, NY, USA, 2010. ACM.
- [92] M. Rosvall and C.T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118, 2008.
- [93] Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Halevy. Clustering query refinements by user intent. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 841–850, New York, NY, USA, 2010. ACM.
- [94] Tefko Saracevic. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for information science*, 26(6):321–343, 1975.
- [95] Tefko Saracevic. Relevance: A review of the literature and a framework for thinking on the notion in information science. part ii: nature and manifestations of relevance. *Journal of the American Society for Information Science and Technology*, 58(13):1915–1933, 2007.
- [96] Tefko Saracevic. Relevance: A review of the literature and a framework for thinking on the notion in information science. part iii: Behavior and effects of relevance. *Journal of*

the American Society for Information Science and Technology, 58(13):2126–2144, 2007.

- [97] Neela Sawant, Ritendra Datta, Jia Li, and James Z. Wang. Quest for relevant tags using local interaction networks and visual content. In *Proceedings of the International Conference on Multimedia Information Retrieval*, MIR '10, pages 231–240, New York, NY, USA, 2010. ACM.
- [98] Markus Schedl, Nicola Orio, Cynthia C. S. Liem, and Geoffroy Peeters. A professionally annotated and enriched multimodal data set on popular music. In *Proc. of the 4th Multimedia Systems Conference*, MMSys '13, pages 78–83, New York, NY, USA, 2013. ACM, ACM.
- [99] Sebastian Schmiedeke, Peng Xu, Isabelle Ferrané, Maria Eskevich, Christoph Kofler, Martha A. Larson, Yannick Estève, Lori Lamel, Gareth J. F. Jones, and Thomas Sikora. Blip10000: A social video dataset containing spug content for tagging and retrieval. In *Proc. of the 4th ACM Multimedia Systems Conference*, MMSys '13, pages 96–101, New York, NY, USA, 2013. ACM.
- [100] Heng Tao Shen, Beng Chin Ooi, and Kian-Lee Tan. Giving meanings to www images. In *Proc. of the 8th International Conference on Multimedia*, MM '00, pages 39–47, New York, NY, USA, 2000. ACM.
- [101] Yi Shen and Jianping Fan. Leveraging loosely-tagged images and inter-object correlations for tag recommendation. In *Proceedings of the international conference on Multimedia*, MM '10, pages 5–14, New York, NY, USA, 2010. ACM.
- [102] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 327–336, New York, NY, USA, 2008. ACM.
- [103] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, December 2000.
- [104] Yale Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '15, pages 5179–5187. IEEE, June 2015.
- [105] Umberto Straccia. An ontology mediated multimedia information retrieval system. In *Proc. of the 40th IEEE International Symposium on Multiple-Valued Logic*, pages 319–324. IEEE, May 2010.
- [106] James Surowiecki. *The wisdom of crowds*. Anchor Books, 2005.
- [107] Johan W. Tangelder and Remco C. Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl.*, 39(3):441–471, September 2008.
- [108] Jing Tian, Tinglei Huang, Yu Huang, Zi Zhang, Zhi Guo, and Kun Fu. A new method for image understanding and retrieval using text-mined knowledge. In *Advanced Data*

Mining and Applications, pages 684–694. Springer, 2014.

- [109] Theodora Tsirikla, Christos Diou, Arjen de Vries, and Anastasios Delopoulos. Reliability and effectiveness of clickthrough data for automatic image annotation. *Multimedia Tools and Applications*, 55:27–52, 2011.
- [110] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):467–476, 2008.
- [111] Arash Vahdat, Guang-Tong Zhou, and Greg Mori. *Computer Vision – ECCV 2014: 13th European Conference*, chapter Discovering Video Clusters from Visual Features and Noisy Tags, pages 526–539. Springer International Publishing, Sep 2014.
- [112] Reinier H. van Leuken, Lluís Garcia, Ximena Olivares, and Roelof van Zwol. Visual diversification of image search results. In *Proc. of the 18th International Conference on World Wide Web, WWW '09*, pages 341–350, New York, NY, USA, 2009. ACM.
- [113] Thomas Vander Wal. Folksonomy, 2007.
- [114] Mauricio Villegas and Roberto Paredes. Overview of the imageclef 2012 scalable web image annotation task. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.
- [115] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14*, pages 157–166, New York, NY, USA, 2014. ACM.
- [116] Avery Wang. The shazam music recognition service. *Commun. ACM*, 49(8):44–48, August 2006.
- [117] Dingding Wang, Tao Li, and Mitsunori Ogihara. Are tags better than audio features? the effect of joint use of tags and audio content features for artistic style clustering. In *11th International Society on Music Information Retrieval Conference, ISMIR*, pages 57–62, 2010.
- [118] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [119] Wei Wang, Xiaoyan Yang, Beng Chin Ooi, Dongxiang Zhang, and Yueting Zhuang. Effective deep learning-based multi-modal retrieval. *The VLDB Journal*, 25(1):79–101, 2016.
- [120] Xinggang Wang, Xiong Duan, and Xiang Bai. Deep sketch feature for cross-domain image retrieval. *Neurocomputing*, 207:387 – 397, 2016.
- [121] F Wiering. Can humans benefit from music information retrieval? In *Proc. 4th Int. Conf. Adapt. Multimed. Retr. user, Context. Feed.*, pages 82–94, Berlin, Heidelberg,

2006. Springer-Verlag.

- [122] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):716–727, March 2013.
- [123] Lei Wu, S.C.H. Hoi, and Nenghai Yu. Semantics-preserving bag-of-words models and applications. *Image Processing, IEEE Transactions on*, 19(7):1908–1920, July 2010.
- [124] Lei Wu, Steven Hoi, and Nenghai Yu. Semantics-preserving bag-of-words models for efficient image annotation. In *Proc. 1st ACM Workshop on Large-Scale Multimedia Retrieval and Mining, LS-MMRM '09*, pages 19–26, New York, NY, USA, 2009. ACM.
- [125] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics, ACL '94*, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.
- [126] X. Xu, A. Shimada, and R. i. Taniguchi. Tag completion with defective tag assignments via image-tag re-weighting. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2014.
- [127] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07*, pages 824–833, New York, NY, USA, 2007. ACM.
- [128] Rong Yan and Alexander Hauprmann. Query expansion using probabilistic local feedback with application to multimedia retrieval. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07*, pages 361–370, New York, NY, USA, 2007. ACM.
- [129] Christopher C. Yang and K. Y. Chan. Retrieving multimedia web objects based on pagerank algorithm. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, WWW '05*, pages 906–907, New York, NY, USA, 2005. ACM.
- [130] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [131] Mark Yatskar, Lucy Vanderwende, and Luke Zettlemoyer. See no evil, say no evil: Description generation from densely labeled images. *Lexical and Computational Semantics (*SEM 2014)*, page 110, 2014.
- [132] Jun Yu, Richang Hong, Meng Wang, and Jane You. Image clustering based on sparse patch alignment framework. *Pattern Recognition*, 47(11):3512–3519, 2014.
- [133] M. Zaveršnik and V. Batagelj. Islands. *International Sunbelt Social Network Conference*, 2004.

- [134] Zheng-Jun Zha, Linjun Yang, Tao Mei, Meng Wang, Zengfu Wang, Tat-Seng Chua, and Xian-Sheng Hua. Visual query suggestion: Towards capturing user intent in internet image search. *ACM Trans. Multimedia Comput. Commun. Appl.*, 6(3):13:1–13:19, August 2010.
- [135] Xin Zheng, Deng Cai, Xiaofei He, Wei-Ying Ma, and Xueyin Lin. Locality Preserving Clustering for Image Database. In *Proceedings of the 12th Annual ACM Int. Conf. Multimedia*, pages 885–891. ACM, 2004.
- [136] Guangyu Zhu, Shuicheng Yan, and Yi Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 461–470, New York, NY, USA, 2010. ACM.

Appendix A

Benford's Law and Zipf's Law

A.1 Benford's Law

Benford's law, also known as *law of anomalous numbers*, and *first-digit law*, is an observation about the frequency distribution of leading digits in many real-life sets of numerical data. Benford's law states that in listings, tables of statistics, etc., the digit 1 tends to occur with probability about 30%, much greater than the expected 11.1% (i.e., one digit out of nine). Benford's law also makes predictions about the distribution of second digits, third digits, digit combinations, and so on. The implications of the digit rule are significant as not only is the distribution not uniform, implying that digit frequencies are not independent, but to be true it must also hold irrespective of the units of the data as well as their source. Hence a universal property of real world measurements is implied.

The law applies to budget, electricity bills, street addresses, stock prices, house prices, population numbers, death rates, lengths of rivers and processes described by power laws. In the face of such universality of the law, it's quite astonishing that there exists a more general framework, Zipf's Law, which falls under a more general rubric of scaling phenomena. It is named after physicist Frank Benford, who stated it in 1938 in a paper titled "*The Law of Anomalous Numbers*" [9], although it had been previously stated by Simon Newcomb in 1881. A set of numbers is said to satisfy Benford's law if the leading digit d ($d \in 1, \dots, 9$) occurs with probability:

$$P(d) = \log_{10}(d + 1) - \log_{10}(d) = \log_{10} \left(\frac{d + 1}{d} \right) = \log_{10} \left(1 + \frac{1}{d} \right)$$

An extension of Benford's law predicts the distribution of first digits in other bases besides decimal; in fact, any base $b \geq 1$. The general form is:

$$P(d) = \log_b(d + 1) - \log_b(d) = \log_b \left(1 + \frac{1}{d} \right)$$

For $b = 2$ (the binary number system), Benford's law is true: All binary numbers (except for 0) start with the digit 1. Nevertheless, the generalization of Benford's law to second and later digits is not trivial, even for binary numbers.

A.2 Zipf's Law

Zipf's law [79] is an empirical law formulated using mathematical statistics that refers to the fact that many types of data studied in the physical and social sciences can be approximated with a discrete power law probability distributions. Zipf's law states that given a large sample of words used, the frequency of any word is inversely proportional to its rank in the frequency table. So word number N has a frequency proportional to $1/N$. For example, Zipf claimed that the largest city in a country is about twice the size of the next largest, three times the size of the third largest, and so forth. While the fit is not perfect for languages, populations, or any other data, the basic idea of Zipf's law is useful in schemes for data compression and in allocation of resources by urban planners.

The law is named after the linguist George Kingsley Zipf, who popularized it and sought to explain it, though he did not claim to have originated it. Zipf's law is most easily observed by plotting the data on a log-log graph, with the axes being \log (rank order) and \log (frequency). The data conform to Zipf's law to the extent that the plot is linear. Zipf's law predicts that out of a population of N elements, the frequency of elements of rank k , $f(k, s, N)$, is:

$$f(k, s, N) = \frac{1/k^s}{\sum_{n=1}^N (1/n^s)}$$

where: N is the number of elements, k is their rank, and s is the value of the exponent characterizing the distribution.

Zipf's law holds if the number of elements with a given frequency is a random variable with power law distribution $p(f) = \alpha f^{-1-1/s}$. It has been claimed that this representation of Zipf's law is more suitable for statistical testing. If we use the classic version of Zipf's law to analyze more than 30,000 English texts, the exponent $s = 1$. $f(k; s, N)$ will then be the fraction of the time the k^{th} most common word occurs.