INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS MONTERREY

SCHOOL OF ENGINEERING AND INFORMATION TECHNOLOGIES
DIVISION OF MECHATRONICS AND INFORMATION TECHNOLOGIES
GRADUATE PROGRAMS

DOCTOR OF PHILOSOPHY
in
INFORMATION TECHNOLOGIES AND COMMUNICATIONS
MAJOR IN INTELLIGENT SYSTEMS

A Process for Extracting Groups of Thematically Related Documents in Encyclopedic Knowledge Web Collections by Means of a Pure Hyperlink-based Clustering Approach

By

Sara Elena Garza Villarreal

MAY 2010

A Process for Extracting Groups of Thematically Related Documents in Encyclopedic Knowledge Web Collections by Means of a Pure Hyperlink-based Clustering Approach

A dissertation presented by

Sara Elena Garza Villarreal

Submitted to the
Graduate Programs in Mechatronics and Information Technologies
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Information Technologies and Communications
Major in Intelligent Systems

Thesis Committee:

Dr. Ramón F. Brena Pinero	-	Tec de Monterrey, Campus Monterrey
Dra. Alma Delia Cuevas Rasgado	-	Instituto Politécnico Nacional
Dr. Juan Carlos Lavariega Jarquín	-	Tec de Monterrey, Campus Monterrey
Dra. Satu Elisa Schaeffer	-	Universidad Autónoma de Nuevo León
Dr. Hugo Terashima Marín	-	Tec de Monterrey, Campus Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey, May 2010

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Division of Mechatronics and Information Technologies
Graduate Program

The committee members, hereby, certify that have read the dissertation presented by Sara Elena Garza Villarreal and that it is fully adequate in scope and quality as a partial requirement for the degree of **Doctor of Philosophy in Information Technologies and Communications**, with a major in **Intelligent Systems**.

Dr. Ramón F. Brena Pinero
Tec de Monterrey, Campus Monterrey
Principal Advisor

Dra. Alma Delia Cuevas Rasgado
Instituto Politécnico Nacional
Committee Member

Dr. Juan Carlos Lavariega Jarquín
Tec de Monterrey, Campus Monterrey
Committee Member

Dra. Elisa Schaeffer
Universidad Autónoma de Nuevo León
Committee Member

Dr. Hugo Terashima Marín
Tec de Monterrey, Campus Monterrey
Committee Member

Dr. José Luis Gordillo Moscoso
Director of the Doctoral Program in Information Technologies and Communications
Division of Mechatronics and Information Technologies
Graduate Program

Copyright Declaration

I, hereby, declare that I wrote this dissertation entirely by myself and, that, it exclusively describes my own research.

Sara Elena Garza Villarreal
Monterrey, N.L., México
May 2010

Dedication

EXCEPT THE LORD BUILD THE HOUSE,
THEY LABOR IN VAIN THAT BUILD IT:
EXCEPT THE LORD KEEP THE CITY,
THE WATCHMAN WAKETH BUT IN VAIN.
(Psalm 127, 1)

This is for God and for my family, both in flesh and in spirit. You are my greatest joy.

Acknowledgements

I would like to express my deepest gratitude to all of those who have been side by side with me:

First, to my Lord, for the gift of life and his providence along the way. Without Him, none of this would have been possible; my work actually belongs to him, as He was the one who placed in me the desire to fulfill this dream and the means to achieve such goal…I am also very grateful for the intercession of my Blessed Mother, Mary, and my heavenly friends, the saints.

To my family, who have totally supported me on *every aspect*. To my father, Félix, specially for motivating me to study this PhD; to my mother, Laura, specially for joining me to the stay at Microsoft, and to my sister, Ivonne, for always being there. Needless to say, without your help, this dissertation would not even exist.

To the Tec de Monterrey, the DTC program, the Context Intelligence research chair, and the National Council of Science and Technology (CONACYT), for their financial support and the facilities provided.

To my advisor, Dr. Ramón Brena, and to the thesis committee: Dra. Alma Delia Cuevas, Dra. Satu Elisa Schaeffer, Dr. Juan Carlos Lavariega, and Dr. Hugo Terashima. I really appreciate your time, patience, and advice.

To my friends and colleagues, specially Lorena, Eduardo, Adrianita, and Marcelo, who were closer to this project. Thanks for everything!

To the secretaries, Doris and Isabel, for helping me out with administrative issues.

To my beloved Vanclar and Inesian family, for their prayers.

Finally, to everyone who was involved—in any way—with the development of this project.

A Process for Extracting Groups of Thematically Related Documents in Encyclopedic Knowledge Web Collections by Means of a Pure Hyperlink-based Clustering Approach

by

Sara Elena Garza Villarreal

Abstract

The present dissertation is being submitted as a requirement for obtaining the degree of Doctor in Information Technologies and Communications with a major on Intelligent Systems. Regarding contents, an approach for extracting groups of topically related documents in an encyclopedic knowledge Web collection by means of a hyperlink-based clustering method is introduced.

With the advent of Web 2.0, applications that exploit the "wisdom of crowds" have turned the Web into a dynamic, heterogeneous, massive—and almost chaotic—information spot. In this increasingly complex environment, information organization becomes not only convenient but actually necessary. Because manual solutions require a great deal of time and effort (while covering only a small fraction of the Web's resources), alternatives for *automatically* discovering the underlying semantics of the Web have been encouraged.

A more specific problem regarding automatic information organization concerns extracting the latent *topics* of a collection; this *topic extraction* task has been usually considered as a *secondary* endeavor, and a concrete definition for a topic still remains an open issue (specially on the Web), situation that motivates us for focusing on this discipline. Furthermore, since topics can be conceived as *document groups*, clustering can be employed as our main engine for detection. On the other hand, from all the available document source information, *hyperlinks*—common to find on our domain—seem to be suitable to use, not only because they are language-independent, but also since they are more clear and objective than text. Moreover, we can concentrate specially on *Web encyclopedias*, as it seems more natural to visualize knowledge pages as fitting into diverse subject matters. Finally, the best representative for such type of collections is *Wikipedia*, which acts as our case study.

To treat the topic extraction process, we can consider it as complying with four main axes or *topic sub-tasks*: definition, construction, description, and validation. These tasks, on their own, can be aligned to an overall *topic extraction conceptual model* (TCM) that views topics as document clusters whose semantics reveal a thematic bondage and depicts the general solution in terms of distinct abstraction layers.

Regarding topic definition, a graph-theoretic extraction *formal conceptual framework* that comprises the basic elements and operations of the process is introduced.

With respect to topic construction, this sub-task is regarded as the most important one of the extraction process. A first aspect consists of *link information extraction*, as it prepares data for the clustering procedure. However, the core of construction falls upon hyperlink-based document clustering. As for our clustering approach, it is heavily related to *community* *detection*, because it searches for highly inter-linked overlapping groups. Consequently, the method is *structure-based* (graph theoretic) and assumes that topics will tend to concentrate into *maximum-cohesion subgroups* (i.e., dense groups), which can be visualized as the local optima (peaks) on a multidimensional surface. The corner stone of the construction approach consists of *Graph Local Clustering* (GLC), which builds clusters on a bottom-up fashion by iteratively adding elements that increase the current group's cohesion.

For topic description, two main properties are being considered: a representative document subset and a tag. *Degree centrality* and a *weighting scheme* based on text frequency are used for obtaining such properties, respectively.

With respect to topic validation, our construction approach was carried out over the Wikipedia corpus to produce a clustering, which was evaluated internally and externally (alignment with Wikipedia categories, user tests). *Our results support the main hypothesis: the discovered groups are topics.*

In an overall sense, our approach is able to: detect groups with a common thematic based solely on structure, find the cohesive topics of a hyperlinked knowledge collection, and take into account the whole corpus. Our main contributions, besides the described topic extraction process, consist of a conceptual framework for topic detection, a specific method for topic extraction via document clustering, and a considerable quantity of topical clusters belonging to Wikipedia.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

With the advent of Web 2.0, new paradigms regarding content creation, design, and use of the WWW have been introduced since the past several years; some of these changes include exploiting *collective intelligence* and the so-called "wisdom of crowds" [105]. The former encourages users not only to read or watch, but to get actively involved by publishing contributions as well; this has resulted in the creation and expansion of content and structure-rich social platforms such as `Youtube`, `myspace`, `Facebook`, and `Wikipedia`. In the midst of this Web 2.0 application explosion, we clearly see how the Web has turned (since some time now) into a *public, dynamic, heterogeneous, massive—and almost chaotic—information spot*. For instance, consider the current size of the indexed Web, which is at least of 29.6 billion pages[1]; while this amount of webpages keeps increasing on a regular basis, users wish to be able to browse and search this huge on-line repository in an efficient manner. For this reason, solutions that facilitate access, understanding, and retrieval have been fostered—the most popular and important one being an initiative known as Web 3.0 or the "Semantic Web" [11].

1.1 Problem and motivation

While being introduced to the *context* where a number of *issues* takes place, we are able to notice several opportunity areas that it might be favorable to tackle. Of course, it becomes necessary to set apart and *delimit* one of these areas to study its *background* with more detail, *state* the specific problem it poses, and be able to provide a concrete solution.

1.1.1 Context

Web 3.0 aims to provide tools for webpages to be "machine readable" (thus, making it easier to search, for example). The traditional approach has been to structure data and give it a meaning by using ontologies (with OWL, RDF, DAM + OIL, etc). However, the migration from non-structured resources to semantic pages has not been fulfilled yet, and—consequently—there is a considerable gap between the desired Semantic Web and the current WWW. Some critics

[1] http://www.worldwidewebsize.com/ (retrieved June 2009)

have even stated that the "Semantic Web keeps unrealizable" [130]; even when this could be true, the need for such a Web is still present.

The former demand motivates us to look for other types of alternatives, perhaps more in the fashion of "soft computing"[2]. That way, we could take advantage of the existing Web—which is presumably self-organized [43]—to uncover or infer semantic properties, instead of "hard-coding" them. Actually, soft methods could even act as a bridge for the transition between Web 2.0 and Web 3.0 to finally take place.

So, up to now, we can state the following as the current situation of the issue under study:

1. The Web's size and complexity keeps increasing at every moment.
2. There is no point on having such an extensive and rich Web if an adequate structure for its exploration is not provided.
3. Manual alternatives, such as human-made directories or the ontologies proposed by the "hard" Semantic Web are totally valid, but require a great deal of effort and are still on the way of becoming widely used.
4. The former leads us to consider automatic options, inspired in soft computing.
5. Because the Web—even when it may seem chaotic on its surface—is actually "self organized", we can use these methods for uncovering its latent semantic structure.

Discovery of the Web's underlying semantic structure enables *automatic document organization*. This includes distinct aspects (not necessarily mutually exclusive), for which several approaches have been implemented:

- Document ranking: Involves measuring the value or worthiness of documents to sort them in order of importance; this is successfully accomplished by the PageRank algorithm [21].
- Document indexing: Involves creating structures to facilitate access and search. One of the main techniques here concerns Latent Semantic Analysis [34], which maps terms to concepts by means of Singular Value Decomposition (SVD).
- Webpage snippet creation: Consists of summarizing webpages for users to grasp their contents when presented, e.g. as results of a search engine. See Amitay's work [6] for a representative approach.
- Resource discovery: Consists of popular page and reference list spotting for a given query; this task is performed by the HITS algorithm [73].
- *Document cataloging into one or more topics.*

The last point concerns our particular problem of interest; therefore, it might be convenient to separate it from the rest in order to state our motivation for solving it and how it can be delimited with the intent of providing a specific solution approach.

[2] The soft computing concept was actually used by Zadeh in the context of Fuzzy Logic to refer to those methodologies that "aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost" [150]. This does not mean that we will use Fuzzy Logic, but rather implies that our method will work with the *actual* Web.

1.1.2 Delimitation

Document cataloging into one or more topics is given a considerable number of names. The two most common ones that we will use throughout the rest of the present work are *topic extraction* and *topic mining*, but *detection*, *discovery*, and *identification* are equivalent as well.

Topic extraction in the Web is still a broad discipline and comprises several aspects that can be specified for delimiting the problem; for that reason, it is important to explain these aspects and provide justification and motivation for solving our particular problem.

Topic extraction with document clustering based on hyperlinks on encyclopedic knowledge corpora using Wikipedia as a case study

In abstract terms, topic extraction involves *discovering the themes* that are present within a document collection; also, it allows to find related items more easily, aids visualization, and helps to conceptualize the composition of the collection, among other things. Even when these simple lines might succeed to create an understandable general notion of this discipline, there is “more than meets the eye” with topic extraction.

On one hand, while a topic is broadly defined as a theme or subject matter, there is currently not a clear, *concrete* definition for this entity—much less for its extraction. In fact, it is rare to find specific, formal definitions for topics. In that sense, this concept is fairly loose, implicit, and obscure to some extent: it comprises a considerable number of notions, it is usually expected for its meaning to be grasped from the exposed idea in relevant works, and a universally accepted standard definition still remains an open issue.

Moreover, *particularly on the Web*, topic extraction is generally treated as a derived endeavor that feeds itself from the “rebounced” results obtained by similar—but not equivalent—disciplines. For example, the topicality of a set of webpages found by techniques of community discovery (a discipline with an inherent social context) is, for the usual, taken as granted. Even when using techniques from other contexts is completely valid (more when they solve the same problem in spirit), a proper extrapolation is not being done, since an explicit bridge between one context and the other is missing. As a consequence, topic extraction remains partly vague.

With a loose definition on one end and a mining (extraction) process relegated in favor of different disciplines on the other, it becomes highly motivating to exclusively address the particular problem of topic extraction.

Now, because topics can be considered as natural groupings that represent the classes of a collection [7], one way to concretely view them is as *document clusters*. Therefore, topic extraction can be based on or “wrap” a clustering process by employing it as an *engine* for finding document related groups; then, the rest of the extraction process could consist of *contextualizing* the detected groups in the topic domain, e.g. by validating that the aspect that keeps the documents together is a common theme and not another concept, like ambiguity. Furthermore, viewing the core of our problem as a clustering issue allows us to take advantage of this mature field, since it offers a comprehensive amount of techniques and other features that we might get insight of; another benefit is that, unlike other methods, clustering is flexible for coping to different kinds of input sources, such as text, hyperlinks, and others (e.g. Meneses proposes the use of *symbolic objects*, which combine various data types [94]).

Another implication with clustering is the type of learning involved; in this case, we refer to *unsupervised learning*. Although both supervised and unsupervised methods are effective in a range of environments, we believe that the most appropriate one for the Web domain is the latter. On one hand, a set of pre-defined learning examples (needed for a supervised case) is not always available; on the other hand, unsupervised learning enables genuine *discovery*, in the sense that it allows detecting groups that are not necessarily represented by the reference classes that have been created so far. This is advantageous, because we might actually find groups whose existence we were unaware of.

Although clustering is a way to ground topic extraction, it is still very wide; therefore, it is convenient to continue delimiting our problem. A way of doing this is precisely by selecting the information source type that is to be managed; the three candidates essentially would be content (*text*), structure (*hyperlinks*), and a *hybrid source*.

Because it is better to first observe and quantify the effects of a single information type on the corpus of interest (specially with the case study collection that we are about to discuss), it is apparently more favorable to work initially with a *pure* source, either text or links. Let us briefly compare both.

Regarding text-based methods, their greatest strength is that they can be used for any type of collections—either plain text or hypertext; furthermore, content is the most usual information source, and plenty of works can be found about its use. However, the main drawback of textual information is the well-known *Word Sense Disambiguation* (WSD) problem [62, 101], in which the two main affairs are *synonymy* (different words that have the same meaning) and *polisemy* (different definitions for the same word). Also, text is more susceptible to *subjectivity*, specially in the midst of collections that gather a considerable number of assorted writing styles.

Link-based methods, on the other hand, let us work around text ambiguity issues by exploiting relations among data (in fact, these methods can find a substantial aid on graph-theoretic approaches). As the reader can infer, link-based methods cannot be applied over plain text collections, but can be rather *attractive for Web environments*; furthermore, a great advantage of structure is that it is *language-independent*. It is true that hyperlinks have other disadvantages, but these are in part mitigated with the type of corpora we intend to use, and this is our next discussion point.

A directly relevant environment for topic extraction is given by *on-line encyclopedic knowledge collections*, where articles tend to share common thematics and can be encompassed into global categories in a natural way.

As mentioned earlier, hyperlink use has both advantages and disadvantages. Regarding the latter, a reported drawback concerns "rival site linking", i.e., websites of competing companies or other entities that avoid linking each other (e.g., `Apple` and `Microsoft`); this, logically, impacts structure. However, this problem is more inherent with collections composed of commercial webpages, and this is not the case of didactic corpora, since these are neutral. Similar to the previous problem is "link spam", which—among other things—presents hyperlinks with a deceiving anchor text; such links lead to malicious or unrelated sites. Although this problem is persistent on many collections (not always because of deception, but rather by *irrelevance*), the effect of link spam is minimized on knowledge corpora. For instance, on Wikipedia, if the document denoted by an internal link is not found on the repository, it is highlighted with a different color (red). As a result, the structure of encyclopedic knowledge collections is more reliable.

On the other hand, there is already important prior work done on structure analysis for other types of networks, such as social networks and citation networks, where links consist of references among scientific papers. In fact, trends such as the "Web as a graph" have popularized the use of structure (links) and graph-theoretic methods for solving Web-related problems. Consequently, this prior work could be useful for providing foundations to our solution, and, at the same time, the execution of this kind of analysis in knowledge networks could give new insights. Seeing all of this, focusing our attention on this class of collections appears to be worthy.

Finally, it seems valuable to concentrate on a specific corpus for developing hyperlink-based topic extraction. With regard to this last delimitation point, let us describe a collection of rising interest that is rich in content and structure: Wikipedia.

Being accounted as the 7th most visited site in the world[3], this online repository has been credited as the fastest-growing, most current, and largest encyclopedia, and has gained a considerable acceptance for admitting contributions from *anywhere* and *anyone* at *any time* (which actually also causes the collection to be dynamic and highly heterogeneous). Its English version contains approximately 3 million articles, and these have been written by at least 35,000 persons, but read by more than other 65 million[4]. Furthermore, it has created such an impact on the Web community, that it is becoming a frequent choice to *wikify* regular webpages by introducing links to Wikipedia entries into their contents with the purpose of explaining certain concepts (see [96] and [97] for efforts to automate this process).

While Wikipedia inherits some of its main features from the Web—like being a popular, rapidly-evolving publication platform—, it is also true that this collection admits several particular aspects of its own. One such aspect is the availability of valuable document-cataloging manual resources (topic lists, portals, a category hierarchy, . . .). Unlike general Web directories (e.g., ODP[5]), these resources indeed cover a significant portion of the total documents—in fact, it has been reported that nearly 90% of the articles have been categorized in the Dutch and Catalan Wikipedia versions[6]. Nevertheless, the former, far from being discouraging for the introduction of automatic topic identification techniques, concerns instead a motivation because of two primary aspects: 1) an exhaustive categorization implies that a considerable amount of time and effort is being spent with the intent of creating and updating categories and 2) such manual resources are still *subjective* (this second point being quite sensitive when it comes to Wikipedia). Moreover, another source for subjectivity (but in the sense of a lack of knowledge) is that a complete corpus awareness is practically impossible—even for small topics—, not only because of the size, but also for the constant changes it suffers. Contributors are therefore not expected (not even a community of them) to provide universal, omniscient resources.

It is for these reasons (relevance and complexity) that automatic methods for the task we have described seem appropriate. On one hand, they can let contributors—at least by making

[3] http://www.alexa.com/topsites (retrieved June 2009)

[4] http://siteanalytics.compete.com/wikipedia.org/?metric=uv (retrieved June 2009)

[5] Open Directory Project. Currently references 4.6 million sites. Site: http://dmoz.org (retrieved June 2009)

[6] http://en.wikipedia.org/wiki/Wikipedia_talk:Special:UncategorizedPages (retrieved June 2009)

suggestions—to concentrate on more critical aspects, like information credibility and accuracy. On the other hand, they enforce the document collection to become *semantic-oriented* as well. Consequently, it seems worthwhile to explore automatic topic extraction here.

A sensible aspect regarding the selection of Wikipedia as a case study is whether the approach to be proposed is exclusive for this collection (in other words, what happens if we have a repository that is not as structured as Wikipedia); with respect to this issue, we might as well clarify that our approach is not married to Wikipedia, although we use it for our exploration on the domain. In that sense, the approach can be applied to other Web corpora (even different from encyclopedic collections); it really does not matter if the repository is very dense or not, the only mandatory aspect is the presence of hyperlinks. In fact, as we will see later, the general method our specific approach sits on has been used on different collections of websites. However, our special interest on Wikipedia is founded on the fact that it is a challenging case study and, on the other hand, results on the Wikipedia domain have a *high impact*, as this repository is of interest for the research community (e.g., refer to the Wikipedia Labs[7], which are totally devoted to studying this corpus, and a specific research area is given precisely by Wikipedia mining). Over all of this, Wikipedia also seems the indicated corpus to work on if we want to get involved with topics, because it covers knowledge areas of different types.
The delimitation for our problem is shown in Figure 1.1.

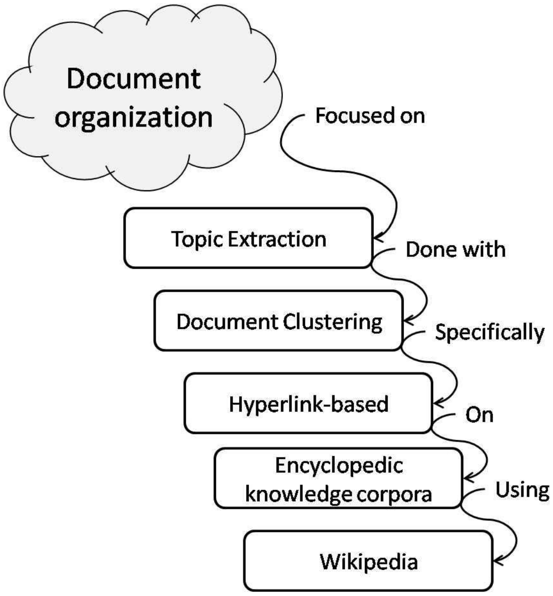


Figure 1.1: Problem delimitation

1.1.3 Background and issues

The problem that has just been delimited comprises a specific background and a set of issues that exist on the area. We concisely present this background and bring up some discussion as well to provide a more comprehensive view of the problem. Note that this section is merely

[7] http://wikipedia-lab.org/

introductory; to see a broader background and a more thorough explanation of approaches (what has been done before), please refer to Chapters 2 and 6.

Web structure mining arises from and is inspired by several disciplines. Two of these concern *data mining* and *link mining*, the first referring to the discovery of useful information from data sources, and the second one to mining relational data (e.g., extracting patterns from databases). If we follow this same line, *Web mining* in general attempts, then, to discover useful information from the WWW; the three typical sources are page content (text, images, etc.), structure (hyperlinks), and/or usage data. Seeing all of this, Web structure mining specifically focuses on the *analysis of links* of webpages.

At the same time, link analysis has its origins on works of the *Web as a graph*, area of study that is simultaneously rooted on the analysis of *complex networks*. Not surprisingly, the study of alternate complex networks, such as social and citation nets (which treat people and scientific papers, respectively), has had an impact on the development of structure mining.

With respect to *topic extraction*, by analyzing approaches from this discipline, we can recover topic representations and tasks. In that sense, topics are generally presented as lists of documents, labels, probabilistic models, or a combination of these. The tasks, aligned usually with the different representations, are several, as to know: modeling (characterizing a topic), enumeration (listing the topic's elements), labeling (naming the topic), and distillation (detecting authorities on the topic). Other tasks concern topic segmentation (detecting in a document which segments cover different themes) and Topic Detection and Tracking (TDT), that is specially employed on the domain of broadcast news. From the previously mentioned tasks, probably the most studied one in general has been topic modeling; however, it has been questioned whether these methods are adaptable to corpora of a considerable scale, such as the Web [115]. On the other hand, topic models are usually restricted to text, since they work with frequencies and distributions of words; moreover, even when models are very useful, in other contexts it might be necessary to use other representations, such as document lists.

Within the Web domain, the two topic tasks that have more prominence are *distillation* (mentioned before as "resource discovery") and enumeration. Regarding the distillation endeavor, it has practically been taken by the HITS (Hypertext Induced Topic Search) algorithm, which in fact can be considered as one of the most representative works for Web structure mining (note also the interrelation between topic and structure mining). Nevertheless, HITS is not an approach free from issues; for example, although the "topic" concept is part of the acronym for this approach, an explicit definition for such concept is not present—even the concept of broad query is only explained intuitively. Now, with respect to topic *enumeration*, it has been managed more as community discovery: a discipline that is strongly founded on Social Network Analysis (SNA) and consists of finding groups that have more links to the inside than to the outside. Two representative approaches are the ones of Flake [41] and Kumar [78]; however, these related works are not specialized properly in topics, since they have most of the interest on the social aspect (webpage authors) and the particular structure of communities by themselves, i.e., without taking thematic into account.

It is also important to highlight relevant background concerning Wikipedia. Because it is such a broad subject of study, we might rather focus on one aspect: semantic information extraction (we could also call this "mining"). A recurrent trend is to only utilize Wikipedia as an information source from which knowledge can be extracted, but not so much as a destination that could benefit from such knowledge discovery; in that sense, the target corpora of these works is other than this on-line encyclopedia. Another inclination is to use only a part (selected portion) of the corpus; therefore, not all works venture on taking all the articles available. Finally, it is not common to see the sole link structure used; works are more bent towards using content (either pure text or in combination with other resources, such as information boxes).

1.1.4 Problem statement

Within our delimitation frame, topic extraction can be defined as the task of *discovering topically-related document groups*. We can decompose further this task into four main components or "sub-tasks" for its treatment (Figure 1.2):

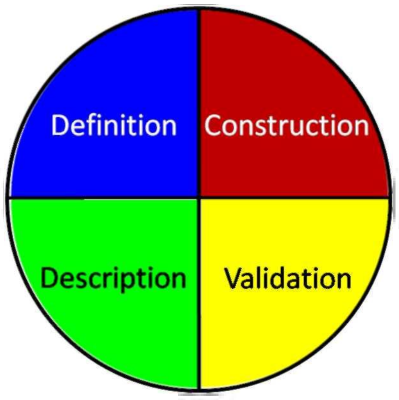


Figure 1.2: Topic sub-tasks

Topic definition.- Establishment of an explicit, formal definition of the topic detection task and its elements.

Topic construction.- Development of a base mechanism for enumerating topic members.

Topic description.- Stipulation of topic properties.

Topic validation.- Confirmation of the topical bondage among group members.

The central aspects of each one are highlighted by defining them as answers to key questions and are presented in Tables 1.1, 1.2, 1.3, and 1.4, respectively.

Other variables and aspects (more technical, perhaps) this problem involves are the following:

Nature of the corpus The corpus has several traits that we must take into account.

Size and complexity Here the main consideration is to enforce (as much as possible) the solution approach to cope with the size of the corpus, which not exclusive of Wikipedia, but is also found on other Web collections and the Web itself. This implies the approach being prepared for working with a large number of documents and links, and a possibly entangled structure.

Table 1.1: Main aspects of the definition sub-task.

Definition.- *Consists of establishing an explicit, formal definition for the topic extraction task in the context environment and the elements involved.*	
Key question	**Related aspect**
What is the intuitive definition of our task?	Define clearly what topic detection in a Web collection is.
Which elements make up the task?	List the task's elements.
In terms of the acknowledged elements, how do we formally define the task?	Provide a formal Web topic detection framework.

Table 1.2: Main aspects of the construction sub-task.

Construction.- *Regards the development of a base mechanism for enumerating topic members; in other words, it consists of providing a process for mapping each document into one or more topics.*	
Key question	**Related aspect**
How is the mapping going to take place?	Define how the mapping process is going to be carried out.
How is information going to be managed?	Define inputs, outputs, and data operations of the process.
What are the parameters?	Define parameter management.
How is it going to be validated?	Establish evaluation criteria according to context.

Wideness and purpose of the platform On one hand, besides providing facilities for sharing knowledge content, Wiki's also enable discussions and record modification histories; allowing such options not only duplicates the amount of material available for a Wiki site, but also gives rise to a series of phenomena, e.g., controversies and social aspects. It seems important to be aware of this, but also to realize that these phenomena are not of our interest. Therefore, we are only committed to pure content (structure, more correctly), and this implies discarding irrelevant material. On the other hand, the main purpose of the platform is to *inform*; in that sense, our case study corpus is *natural* (was not artificially created for experiments) and does not intend to make research easier—at least not directly. As a consequence, we must look for ways to handle contents (e.g., pre-processing and validation options).

Dynamic structure Wikipedia is not a static corpus, but evolves constantly as users are enabled to introduce modifications at any time. Consequently, without failing to realize that this trait should be addressed at some point, we might choose to relax this property and work with only a *snapshot*.

Other traits In Wikipedia, structure is impacted by several special features, such as

Table 1.3: Main aspects of the description sub-task.

Description.- *Concerns stipulation of topic properties and their calculation.*	
Key question	**Related aspect**
What properties are we interested in?	Select, on a topic-driven basis, relevant properties.
How are they going to be obtained?	Provide a series of methods for discovering these properties.

Table 1.4: Main aspects of the validation sub-task.

Validation.- *Consists of proving that the extracted topics are coherent.*	
Key question	**Related aspect**
What type of evaluation is going to be carried out?	Enumerate validation techniques that are to be applied according to the context.
How is the topical coherence going to be proved with the chosen evaluation type?	State how the topical bondage among elements is specifically shown by each validation technique.
What metrics are going to be used?	For each evaluation type, list the metrics to be employed.

redirected pages. Also, the corpus is heterogeneous in many senses: there are broad and specific articles, there is a wide variety of subjects covered (academic disciplines, people, events, concepts, etc.), it has been written by a considerable number of editors who have different backgrounds and writing styles, etc.

Nature of topics A document may belong to one or more topics; therefore, it is important to keep in mind that an exclusive partition (one topic per document only) could fall short for this domain. At the same time, it also seems natural to place topics into a hierarchy or directory. While trying not to get overwhelmed by these two highly desirable traits, we can tackle the overlapping aspect first.

1.2 Objectives, hypotheses and research questions

1.2.1 General objective

Our general objective is to *develop a topic extraction process for detecting groups of topically related documents in Web encyclopedic knowledge collections by means of a pure hyperlink-based clustering approach. The extraction process is driven by four main sub-tasks: definition, construction, description, and validation.*

1.2.2 Particular objectives

These objectives flow directly from the general aim:

- ∗ Define a topic detection conceptual framework for a hyperlinked environment.
- ∗ Propose a specific hyperlink-based clustering method for grouping thematically-related documents.
- ∗ Propose properties that describe the resulting document groups and provide strategies for calculating those properties.
- ∗ Apply the approach (construction and description) to our corpus case study.
- ∗ Validate that our document groups are actually topics.

1.2.3 Hypotheses

Along the current work, we are committed towards testing the next hypotheses:

- *In WikiWikiWeb encyclopedic knowledge corpora, densely connected (community-like) groups of documents are held together by a common theme, i.e., they are topically related.*
  - Community structure indicates topicness.
  - There is a positive correlation between the cohesion (density) of these groups and their topicness.
  - Our solution approach is capable of detecting these groups based purely on structure while taking the whole collection into account, not only a small fraction (e.g. 1000 documents).

1.2.4 Research questions

Furthermore, we plan to answer several research questions:

1. What is the relation between structure and thematic in Web knowledge corpora (our study domain)?
2. Is our approach able to detect (construct) topics solely based on structure? How?
3. Is our approach able to extract topics while considering the whole collection? How?

1.3 Solution overview

In an overall sense, our solution consists of finding ways to comply with our established topic sub-tasks. For this reason, part of our approach involves conceptualizing topic extraction as a layered model whose core is given by a clustering process (see Figure 1.3); the other important layer of this model corresponds to *semantics*, as we consider that it is necessary to make clear that our clusters belong to topics. Continuing with this model, at the very bottom lies a (subdivided) level, whose relevance is given by the fact that it facilitates dealing with the actual collection by logically (i.e., conceptually) defining the parts that are needed for clustering and physically obtaining such parts. Finally, at the very top, a layer of applications can make use of the extracted topics; however, such layer is beyond our scope. Of course, each one of our sub-tasks are carried out at one or several of the layers.

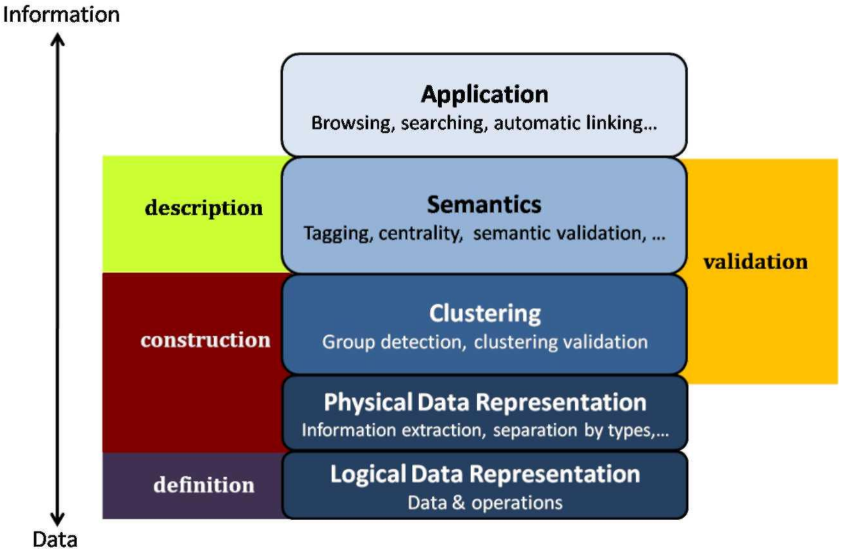

Information
Data
Application
Browsing, searching, automatic linking...
description
Semantics
Tagging, centrality, semantic validation, ...
validation
construction
Clustering
Group detection, clustering validation
Physical Data Representation
Information extraction, separation by types,...
definition
Logical Data Representation
Data & operations

Figure 1.3: Topic extraction layered model

Now, allow us to present an overview of how each individual sub-task is to be tackled:

(Definition).- The main intention is to provide a *graph-theoretic conceptual framework*, where the most impotant elements and operations are explicitly defined and formalized. We basically consider three informational elements: the whole corpus, a single document, and a topic. This last element is the most important entity of our framework, and is represented as a *cluster with (semantic) properties*. With regard to operations (a.k.a. extraction functions), two fundamental ones are described: a construction (basic) function and a clustering (extended) function.

(Construction).- Our intention is to provide a specific *graph-theoretic clustering algorithm* that explores local neighborhoods to create clusters by iteratively adding elements in the vicinity of a starting point. This approach receives the name of *Graph Local Clustering*.

(Description).- We initially consider two properties that serve for making the (topical) semantics of discovered groups more explicit: a name (tag composed of keyword terms) and a set of representative documents. These can be obtained by methods such as word frequency weighting schemes (tf-idf) and SNA-inherited centrality measures, respectively.

(Validation).- We consider that it can be done at two rough levels: one to confirm the cohesion of the groups (clustering level) and the other one to explicitly show that what holds them together is a common thematic (semantic level). To carry out these tests, we consider employing intra-cluster vs. inter-cluster similarity metrics, a set of reference categories extracted from Wikipedia, and human judgment.

Because the construction sub-task is the most critical one, we intend to discuss it with more detail, specially regarding the clustering algorithm.

1.3.1 Clustering approach

There is a comprehensive list of clustering methods available for hyperlinked-document clustering, which range from conventional techniques (such as k-means) adapted for this domain to more specialized alternatives, such as spectral clustering. Even when all these methods exhibit considerable advantages, many of them rely on feature vectors and/or (squared) matrices that encode link information for the whole collection altogether and attempt to produce a clustering for all data at once. This makes them inherently *global*; so, unless an additional strategy is used for dealing with computational demands, these methods can become quite expensive—in time and memory—for the scale of the Web. On the other hand, *local* clustering methods [125], tend to naturally group data based on *partial views*, and compute clusters one at a time (in an agglomerative fashion) by iteratively adding elements in the vicinity of a given starting point. To accomplish this, a graph local clustering approach (which we will call *GLC*, for short) makes use of a local search method that maximizes a cohesion-related fitness function; furthermore, the algorithm is applied over an initial subset of vertices (known as "seed") and neighbors of this subset are added (or removed, in some instances) with each iteration into the resulting cluster. As expected, the algorithm finishes when it does not find a vertex capable of increasing the cohesion value of a given group.

The GLC approach was initially proposed by Virtanen for clustering the Chilean Web with the intent of studying network properties and validating graph generation models [140]. Similarly, it has been further studied as a clustering alternative for massive graphs [124] and to compare with similarity-based techniques [93].

Furthermore, GLC is an approach that: a) is purely structure-based, b) is specialized on finding dense groups (which we assume are topics that represent the "peaks" of a search space where height is given by density), and c) is able to detect overlapping groups, since clusters are created independently from each other.

1.4 Contributions and scope

Our main contribution consists of *a topic extraction process for detecting groups of related documents in encyclopedic knowledge Web collections by means of a pure hyperlink-based clustering approach.*

From this general contribution, other specific ones can be drawn:

Concepts.- A first contribution regarding this aspect consists of the statement and characterization of *four topic extraction sub-tasks* (definition, construction, description, and validation). Such conceptualization of the problem could be used as a guideline for other works on the area. In addition, our *layered conceptual model* for the extraction process (explained with more detail in Chapter 3) could serve for this same purpose, and our *formal framework* already leaves a precedent on topic-related concrete definitions. Actually, this framework could be used as a basis for other conceptions (more specific, more general, different).

Methods.- On one hand, we propose a *specific* method for constructing topics, which is made up of different components—some of these being presented, in fact, as original resarch (e.g., the employed removal strategy). We offer, as well, a set of suggestions for describing and validating topics. All of this opens the possibility for a variety of derived or future works, such as hybrid-source approaches (e.g., contents and structure) and "add-ons".

Products.- Finally, we also contribute with a set of reusable products, from which the set of *Wikipedia-extracted topics* outstands. Along with the proposed methods, these topical clusters can be employed for various applications, such as semantic information retrieval, visualization, and automatic linking. Other related products (perhaps more useful for scientific research) include a pair of automatically generated Wikipedia subcollections and an expanded category tree also from this corpus.

Because we have focused on the above, it is important to mention that our scope comprises the following aspects:

(a) Working with the sole process of *topic extraction* (applications being left as future direction)

(b) Using only *links* for building topics (leave out mixed information sources)

(c) Providing *flat* clusters for representing topics (exclude hierarchical approaches)

(d) Testing the approach exclusively with *Wikipedia*, which is our case study (leave out other corpora)

1.5 Chapter summary

Realizing that the inception of Web 2.0 (the social Web) has turned the WWW into a larger and more complex information spot, the need for making the transition to Web 3.0 (semantically-oriented by nature) becomes substantially more tangible. Because manual approaches require time and effort, several automatic methods have so far been proposed to leverage information organization, as to know: webpage ranking, indexing, snippet creation, resource discovery, and topic extraction. This last task is our subject of interest, because there are still things to be done on this area (currently seen more as a secondary endeavor). Now, topic extraction could be handled by employing *document clustering* as a main engine for finding groups of related documents, and *hyperlink* information—common to find on our domain—not only is language-independent, but also more clear and objective than text. Moreover, we can concentrate specially on *Web encyclopedias*, since it seems more natural to visualize knowledge pages as belonging to diverse subject matters. The best representative for such type of corpora is *Wikipedia*, which acts as our case study.

To begin treating the topic extraction process and solve our specific problem, we can start by considering it as comprising four main axes or *topic sub-tasks*: definition, construction, description, and validation. These tasks, on their own, can be aligned to an overall model that views topics as document clusters whose semantics reveal a thematic bondage; this model is inspired by layered architectures. The main layer, then, is given by clustering; to carry out this endeavor, and because our main hypothesis is that topics will tend to concentrate on groups of highly inter-linked documents (community-like structures), we propose to use graph local clustering (GLC). This method detects groups in a bottom-up fashion by iteratively adding documents in the vicinity of a starting point in order to maximize cohesion among the cluster's members.

Our approach is framed within a specific scope, and several contributions, such as concepts, methods, and products, may be drawn as result from the proposed solution.

1.6 Organization

The current document has the next organization:

Chapter 2: Background and State of the Art This second chapter has two primary purposes: to explain the foundations of our approach (notation, vocabulary, prior work, and concepts that need to be described for a better understanding) and to review (as well as classify) relevant state-of-the-art works. Chapter contents revolve around three main axes: Web structure mining, topic extraction, and Wikipedia semantic information extraction.

Chapter 3: Conceptual Framework We start this chapter by providing a conceptual model for the topic extraction process, which is followed by our graph-theoretic formal framework for topic extraction. This last aspect covers the topic definition sub-task.

Chapter 4: Approach This is one of the main chapters. It covers topic construction and description. With respect to the former, data representation is first addressed; subsequently, the construction and clustering algorithms are explained, both in a rough (basic) and detailed (fine-tuned) form. Finally, topic properties and their calculation (this corresponds to description) are discussed.

Chapter 5: Experiments and Results This chapter is also crucial, as it presents (in summarized form) the topic clusters obtained from Wikipedia, shows how they were validated with respect to their topicality and cohesion, states relevant findings, and discusses results.

Chapter 6: Related Work On this chapter, our work is first situated within the state of the art; afterwards, similar works (with varying degrees of resemblance to our own) are brought up to discussion. In that sense, we describe these approaches with slightly more detail than on Chapter 2 and highlight similarities and differences with regard to our proposed method.

Chapter 7: Conclusions and Future Work This final chapter intends to "wrap up" our work by summarizing it, offering conclusive remarks, listing contributions, and suggesting future work.

Chapter 2

Background and State of the Art

Hyperlink-based topic extraction with Wikipedia as a case study suggests a review in three dimensions: 1) Web structure mining for group detection, 2) topic mining, and 3) semantic information extraction in Wikipedia (which we can refer to also as "Wikipedia mining"). For the first dimension, our main focus is to discuss the principal methods for structure-based group discovery; therefore, a description of these shall be provided, along with their advantages and disadvantages, relevant metrics and computations, specific performed task, type of detected group, way of overcoming complexity, and used evaluation techniques. Of course, such discussion implies the introduction of several basic concepts, notation, and prior work (mostly given by the study of complex networks). Now, for the topic mining dimension, the review is centered on the different topic notions that are currently found in literature, the subtasks related to the discipline in general (not only on the Web), and the representatives of the main approaches. Finally, for the Wikipedia dimension, we deepen into this Web collection by introducing the kind of repository it is and discussing relevant works that aim to automatically organize its information.

Even though the three areas are important, a special emphasis shall be placed upon the first one, since it concerns a field of great interest for the community, comprises some of the most important related work, and constitutes our basic foundations. Furthermore, as we will see later, meaningful group detection describes the core of our topic extraction process, and leads the discussion of the topic mining area to be more focused on describing conceptual aspects. Finally, the aim of analyzing the third area is to present an overview of the efforts devoted to Wikipedia information organization, especially in terms of the Semantic Web. It becomes relevant to mention that the current state of the art review is not—and does not intend to be—exhaustive.

2.0.1 Basic concepts and notation

In order to have a better understanding of the fore coming sections, we will define some basic concepts and introduce the necessary mathematical notation. Regarding the essential vocabulary, a list of the most common terms, along with a succinct description and their respective synonyms is to be provided. As for synonyms, although they will usually express the same concept, it is important to clarify that they might occasionally exhibit slightly different meanings; whenever this is the case, it will be opportunely noted. Now, concerning notation, the most part will be used to denote graph theory concepts.

Taking into account that our field of study is multi-disciplinary, there is a considerable number of knowledge areas involved, as to know: network analysis, information science, data mining, and our specific domain (the WWW and Wikipedia). In some instances, the terms that emerge from these disciplines refer to the same concept and/or may cause confusion if not explained early; these key issues compose our motivation for listing our essential vocabulary:

- **Element**.- Atomic entity.

  Synonyms: node, vertex, member, document, article, actor, page, webpage, object.

- **Link**.- Relationship between entities.

  Synonyms: arc/edge, hyperlink, tie.

  - In-link.- An arc incident *upon* a vertex v_i ($v_j \to v_i$).
  - Out-link.- An arc incident*from* a vertex v_i ($v_i \to v_j$).

- **Corpus**.- Document collection (universe).

  Synonyms: collection, document graph.

- **Group**.- Entity conglomeration.

  Synonyms: cluster, community, sub-group, sub-graph, document set.

- **Grouping**.- Set of groups.

  Synonyms: clustering, partition.

- **Group detection**.- Act of gathering elements into groups.

  Synonyms: clustering, group discovery.

- **Topic**.- Subject matter.

  Synonyms: theme, thematic.

Some of the previous descriptions are still abstract, but suffice to get an initial idea of the implied concept. On the next chapter, the aforementioned terms are to be explained with more detail and formalized as well.

The second part of the current section consists of introducing graph-theoretic notation. While certain definitions (graph, vertex degree, adjacency matrix) are assumed to be known, others shall be provided during the rest of the section.

Let us start by defining a cut. A *cut* is a graph partition that creates two disjoint and non-empty sets of vertices, namely S and $\bar{S}$, ($\bar{S} = V - S$). The *edge-cut* is the set of edges that "cross" the partition by having one endpoint in S and the other one in $\bar{S}$; the cardinality of this set is called the *cut size* and is denoted by $c(S, \bar{S})$.

With the previous foundations, it is possible to introduce a cluster's *internal degree* and *external degree*. First, let us assume that we have a cluster $\mathcal{C}$ (which is equivalent, in practical terms, to a cut S) on an unweighted, undirected graph, and that $\mathcal{C}$ has two kinds of edges: internal and external, the former being edges that have both endpoints on cluster members (these do not belong to the edge-cut) and the latter being edges that have one endpoint on a cluster member and the other on a cluster non-member (and thus belong to the edge-cut set). The number of external edges is considered to be the external degree ($deg_{\text{ext}}(\mathcal{C})$), and analogously, the number of internal edges is the internal degree ($deg_{\text{ int}}(\mathcal{C})$). More formally:

$$deg_{\text{int}}(\mathcal{C}) = |\{(u,v) : (u,v) \in E, u,v \in \mathcal{C}\}| \tag{2.1}$$

$$\begin{aligned} deg_{\text{ ext}}(\mathcal{C}) &= |\{(u,v) : (u,v) \in E, u \in \mathcal{C}, v \notin \mathcal{C}\}| \\ &= c(\mathcal{C}, V - \mathcal{C}) \end{aligned} \tag{2.2}$$

Also, note that this edge classification can be extended to vertices, resulting that, for a particular vertex of $\mathcal{C}$ we have: $deg(v) = deg_{\text{int}}(v,\mathcal{C}) + deg_{\text{ext}}(v,\mathcal{C})$.

Another important concept is the *neighborhood* of a vertex; for a vertex v, it consists of the set of vertices that share an edge with v.
The discussed notation is summarized in Table 2.1.

Table 2.1: Notation.

Graph	$G = (V,E), n = \|V\|, m = \|E\|$
Sub-graph	$G_s = (V_s,E_s), n_s = \|V_s\|, e_s = \|E_s\|$
Edge (undirected graph)	$\{i,j\}, i,j \in E$
Arc (digraph)	$(i,j), i,j \in E$
Adjacency matrix element	a_{ij}
Vertex degree	$deg(i), i \in V$
Vertex neighborhood (undirected)	$\Gamma(i) = \{j : \{i,j\} \in E\}$
Vertex neighborhood (digraph)	$\Gamma(i) = \{j : (i,j) \wedge (j,i) \in E\}$
Cut	$(S, V - S)$ or simply S
Cluster or Community	$\mathcal{C}, \mathcal{C} \in V$
Internal degree	$deg_{\text{int}}(\mathcal{C})$
External deree	$deg_{\text{ext}}(\mathcal{C})$
Vertex internal degree	$deg_{\text{int}}(v,\mathcal{C})$
Vertex external degree	$deg_{\text{ext}}(v,\mathcal{C})$

Having explained some of the basics, let us describe group detection via Web structure mining.

2.1 Web structure mining for meaningful group detection

Web mining, the use of data mining techniques to automatically discover and extract information from Web documents and services [75], can be decomposed into three central axes:

Web content mining.- Analyzes content media (mainly text); this information may be unstructured (free text), semi-structured (HTML pages), or structured (XML, automatically generated HTML). The main purpose is to assist information finding and filtering and data modeling.

Web structure mining.- Uses hyperlinks (which depict structure) in order to discover knowledge; usually, this is done to rank webpages and find "meaningful" groups (i.e. communities of users).

Web usage mining.- Makes use of secondary information or metadata regarding user behavior: session logs, server logs, clicks, scrolls, cookies, transactions, etc. The aim is to extract meaningful usage patterns, which may be exploited in a variety of ways (for instance, to improve usability). Some examples of this task are given by [134] and [107].

Even though each axis is fundamental and is worth to be discussed separately, let us recall that our focus is on Web structure mining; for this reason, usage is only briefly mentioned above and our content mining section concentrates solely on clarifying those aspects that will be retaken on posterior chapters.

Let us describe at this moment the organization of the current discussion. The central point concerns reviewing the different *methods* for meaningful group detection; to understand the most general notions of these, it is necessary to go through *prior work*, mostly represented by complex networks (e.g. citation and social nets). Afterwards, to have a clearer picture of the environment our discussed methods work in, it is convenient to enumerate the specific *tasks* that they carry out for discovering distinct *group types*. Furthermore, this implies analyzing the assorted *metrics and information matrices* they use and then describe the methods properly, by classifying them and enumerating their advantages and disadvantages; now, considering that the Web environment exhibits several special features, we might also mention how methods *overcome complexity* issues. Our final point of discussion covers the *evaluation* schemes that have been employed to validate results. This organization is portrayed in Figure 2.1.

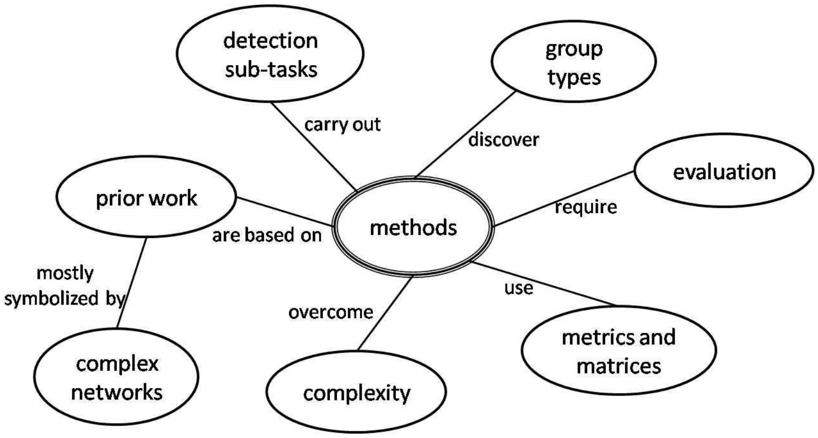


Figure 2.1: Meaningful group detection concept map

2.1.1 Prior studies and related areas

Web structure mining (which is also referred to as *link analysis*) has its roots on the disciplines of *Social Network Analysis* and *bibliometrics* (citation analysis). Furthermore, it parts from the visualization of the *Web as a graph*. Let us describe each of these and highlight the features used for link analysis.

COMPLEX NETWORKS

Complex networks consist of non-conventional networked systems that exhibit irregular properties but, at the same time, share several topological features. These intricate, non-trivial, dynamic structures that stand as the middle point between regular and purely random graphs usually depict real-world networks. For our purposes, we shall describe these structures at a high level; however, the analysis of these structures intrinsically adds several terms to our graph theoretic vocabulary. These definitions are shown in Table 2.2.

Table 2.2: Basic glossary of complex networks

Reachability. If there exists a path between a pair of nodes, then these are said to be *reachable*.

Complete graph. A graph whose edges are all adjacent to each other.

Connected graph. A graph where there is a path between any pair of nodes.

Connected component. Sub-graph in which any belonging pair of nodes is reachable, and in which this property cannot be attained if 1) more nodes are added and 2) we consider a graph that contains the subgraph in question. In other words, we talk about a *maximal connected subgraph*. For digraphs, a *strongly connected component* (SCC) depicts a connected component where paths all contain arcs in the same direction. If the previous condition is not met, but the paths exist, we talk about a *weakly connected component* (WCC).

Path length. Number of edges contained in a path.

Geodesic. Shortest path between a pair of nodes.

Diameter. Size of the largest geodesic (in a graph).

According to Newman's classification [102], complex networks can be divided into four main categories:

Social networks.- Exhibit sets of persons or groups of people who interact among themselves. Examples: collaboration networks, friendship networks (like Facebook), film actor graphs.

Information networks.- Used for sharing data and knowledge. Examples: citation networks, the World Wide Web.

Technological networks Represent physical networks designed for distributing some type of commodity. Examples: power grids, telephone networks, railways.

Biological networks.- Stand for biological systems. Examples: metabolic pathways, protein interaction networks, food webs.

The study of complex networked systems has revealed the existence of special properties that seems worthwhile to take into account and quantify. Some of the most relevant are the following:

Small-world effect.- Reachability of any destination target within the network in only a few steps; the so-called "six degrees of separation"[1] conception displays such property. This trait can be seen as a factor for quick information spreading (disease contagions, packet hops between a source and a destination, rumor dissemination, etc.).

Community structure.- Presence of dense groups, where the members from these groups have many edges among themselves and only a few edges with respect to other elements in the network. This trait indicates, among other things, what factors divide up the net (for instance, race or nationality in a social network, or subject matter in a citation system).

Clustering (transitivity).- Presence of an edge between a node A and a node C when it is known that A is connected to a node B and B is connected to C. The degree of transitivity can be measured by obtaining the fraction of triangles (transitive relations) in the network. This property can be reflected, for example, in acquaintance networks, where persons with a common friend are likely to know each other.

Resilience.- Extent up to which a network preserves connectivity when its elements are progressively removed; variations in this feature can be found by the amount of eliminated vertices and their type (high or low degree, for example). Study of this property is valuable for assessing, for instance, the impact a vaccine causes to the transmission of a disease (where it is assumed that the appliance of antibodies not only affects the vaccinated person, but also "cuts" or eliminates contagions to more people).

Degree distributions.- Type of distribution followed by examining the number of edges incident to and/or from the network's vertices. A "long tail", which indicates a power-law distribution, tends to characterize a considerable amount of complex networks; systems showing this kind of structure are known as *scale-free networks*. Knowing about distributions helps, among other things, to identify special node types, such as *hubs* (that represent comprehensive link lists about a certain subject).

Additional compilations about complex networks include the reviews by Boccaletti et al. [15] and Albert and Barabási [3].

Once having an overview of complex networks, it is possible to go deeper into those types of our interest, such as citation and social networks.

[1] A theory stating that each person is connected to every other one by a distance not greater than six hops (where we understand a "hop" as a pair of persons who know each other).

Citation analysis

Citation analysis studies aspects related to article, journal, and book citation—such as patterns, impact, and other statistics. As a logical consequence, citation networks are systems that depict relations (cites) between academic papers; a special characteristic about these networks is that they are acyclic, since papers can only reference sources that have already been created.

The two main concepts from bibliometrics that regard link analysis are *co-citation* [132] and *bibliographic coupling* [71]. While co-citation states that two documents are similar if they are simultaneously cited by the same sources, bibliographic coupling regards a pair of documents as similar if they both cite the same documents; therefore, these two types of analysis can be considered as complementary. Later, we will see exactly how these quantities are calculated.

Other fundamental facets for this discipline (analyses, statistical studies, open issues, etc.) have been studied mainly by Garfield [45].

Social Network Analysis (SNA)

With respect to Social Network Analysis (SNA), this discipline studies in a formal manner the implications and patterns that emerge from interactions among social entities (persons, organizations, political parties, etc.). For instance, it might be used to study kinship relationships, affiliation to political parties, friendship detection, and the like. Some of the concepts that have been used for link analysis confer *prestige*, *centrality*, *density*, and *cohesive sub-groups*. For the rest of the section, we will employ the term “actor” to refer to a network node and the term “tie” for edges and arcs, in order to respect the domain.

An important issue for SNA is the detection of subsets of social entities among whom there are relatively strong, direct, intense, frequent, or positive ties [145]. These subsets are known as *cohesive sub-groups*, and are considered of key relevance because they uncover certain social aspects, such as consensus, affection, homogeneity, influence, and communication among the actors of the network.

Cohesive sub-groups might be characterized by one or more of the following traits:

1. Mutuality of ties
2. Closeness
3. Frequency of ties
    - Absolute (based on nodal degree)
    - Relative

Mutuality of ties describes pairs of actors that are adjacent to each other; this trait indicates friendship, common choices, and affection. The traditional structure that denotes subgroups based on complete mutuality is a *clique* (a complete maximal sub-graph of 3 or more nodes), since it indicates that every corresponding actor is related to all of its partners, and there is no other actor on the rest of the network that is chosen by all clique members. However, even when cliques are the basic structure, they tend to be strict and not so useful in practice.

Closeness indicates that all members are reachable among themselves (although not necessarily adjacent to each other); this trait indicates influence and communication. Can be considered as a relaxation for the mutuality of ties property. The typical structures are:

n-clique.- a maximal sub-graph in which the largest geodesic is not greater than n

n-clan.- an n-clique in which the diameter (for paths inside the clique) is smaller than n

n-club.- a maximal sub-graph of diameter n

Absolute frequency states that actors have a certain number of ties with other sub-group members; that is, there is a minimum number of adjacent actors to each member of the sub-group. This trait helps to understand processes related to contacts, redundant communication channels, and robustness. Some of the structures used for this trait are:

k-plex.- a sub-graph in which each member lacks at least k ties to other sub-group members. Mostly driven by the absence of ties. $k = 1$ is equivalent to a clique.

k-core.- a sub-graph in which each member is adjacent to at least other k members. Driven by the presence of ties.

Relative frequency suggests that actors have more ties to sub-group members than to non-members. This property indicates strength and the presence of communities. It may also imply density and sparseness in the network. To find this kind of cohesive groups, several approaches have been proposed, i.e., searching for graph structures that reveal relative cohesiveness, such as LS and Lambda sets. Let us briefly describe each structure:

LS set.- each of its proper subsets contains more ties to its complement inside the sub-graph than outside of it

λ **set.-** almost like LS, but uses line connectivity (minimum quantity of links that has to be removed in order to leave no path between a pair of nodes) instead of link count.

Another alternative is to employ measures for sub-group relative cohesion; particularly, Bock and Husain (as mentioned by Wasserman) have developed an iterative way of constructing sub-groups by successively adding members to an existing sub-group as long as the *relative strength ratio* (Eq. 2.17) is preserved.

Apart from cohesive sub-groups, a key concept that link analysis inherits from SNA is the use of *centrality* and *prestige*. Both metrics are related to the sociometric concept of a "star"—that is, the most popular person of a group, who is usually the center of attention [129]. In that sense, centrality measures the *relative importance* of an actor either at a local (network sub-group) or global scale (the whole network); regarding prestige, this term is utilized when dealing specifically with digraphs, but is equivalent to centrality.

There are various types of centrality, which vary according their degree of sophistication and the scope they work within. These are:

Degree centrality.- Also known as "point centrality", this is the simplest form of centrality. It is calculated by obtaining the ratio of an actor's ties with respect to the total amount of actors in the network. The scope of this metric is local, since it only considers *direct* connections (therefore not taking into account paths between actors).

Betweenness centrality.- The idea behind this kind of centrality consists of letting intermediary actors to achieve high scores, as they are fundamental channels of communication and influence. Betweenness calculation involves obtaining the geodesics (shortest paths) between all pairs of actors in the network.

Closeness centrality.- This measure is based on the calculation of the geodesic between a certain actor with respect to the rest in the network.

Eigenvector centrality.- The main idea behind eigenvector centrality is that actors being tied by other well-connected entities are assumed to be more valuable sources of information and hence be considered central.

Technical details about SNA and bibliometrics calculations employed for Web structure mining shall be explained along the section devoted to metrics and computations.

The Web as a graph

Another seminal contribution from which link analysis arises is the one given by works that view the *Web as a graph*. One of the main representatives of this area is the work by Kleinberg et al. [74], which presents a "guided tour" of the most relevant algorithms that have been applied over the Web for discovering information resources and essential measurements that have been carried out over this complex structure. Moreover, according to observations flowing from these measurements, new random graph models and methods to fit the behavior of the "Web graph" are proposed.

Related to the previous work is the review by Broder et al. [22]; this second Web graph analysis provides a thorough study of several important properties of the Web structure, such as degree distributions and reachability. It also introduces an overview of its "bow tie" structure, and this is perhaps its greatest contribution; such structure depicts a strong connected component (SCC) that covers the most part of the nodes, along with tendril and tube substructures (see Figure 2.2). The former consist of portions that either go in or go out without entering the SCC, and the latter stand for passages that do not go through the SCC either.

Another job that follows a similar line is presented by Kumar et al. [77]; it reviews Web graph findings and algorithms for its analysis. Moreover, it introduces methods used for *topic search* and *enumeration* (we will later discuss this pair of tasks).

A more recent analysis done by Bonato [16] summarizes the observed properties of the Web graph:

1. It is a dynamic structure (*on-line* property)
2. Follows a power law distribution
3. It is a small world
4. Contains a considerable amount of bipartite sub-graphs and cores

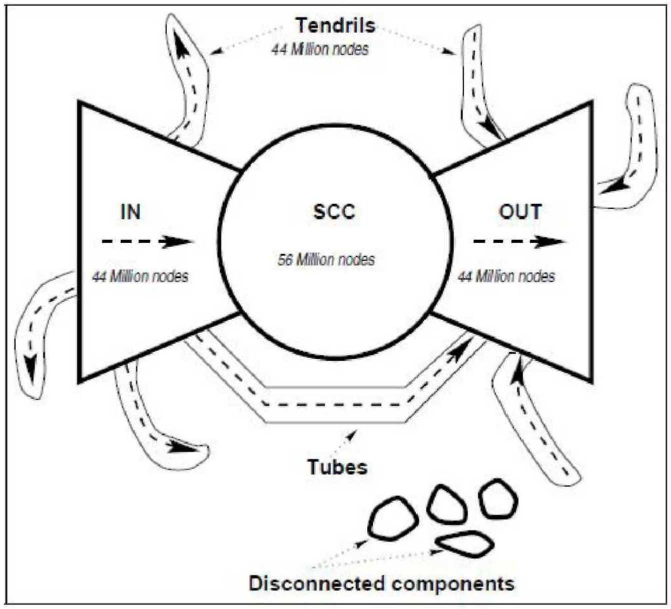


Figure 2.2: Structure of the Web (2000). Source: “Graph structure in the Web” [22].

2.1.2 Group types

Web structure mining for meaningful group identification encompasses methods that intend to find conglomerations whose members either:

(1) Contain distinctive features

(2) Conform a cohesive mass

(3) Share commonalties

The distinctive-feature type represents a *proper subset of distinguished members* (which is usually quite small), according to a certain criterion; because the elements of such group are not necessarily related to each other, this kind of conglomeration may be visualized more precisely as a “bag”. Furthermore, since members outstand from the rest, we can draw two additional aspects about this meaningful group type. First, the primary operations to carry out here are *ranking and selection*; second, the attained coverage of the universe is partial, as there are only a few distinctive-feature sets given an object collection and a specific criterion, and only certain elements are chosen per group. An example of this type of gathering is given by a set of *authorities*, whose main characteristic is their high *popularity* with respect to a certain *subject*.

Common-trait groups, in contrast, *hold similar elements together*. In that sense, every element tends to have a higher resemblance with its fellow group members than with the rest of the objects. Furthermore, the main operation to conform this kind of groups consists of *mapping* all universe members to one or more groups (the groups being initially represented by randomly-taken elements or middle points in the space, which are called “centroids”); a direct consequence of this mapping regards achieving a total collection coverage. To close this description, let us illustrate this second type of conglomeration with a document *cluster* gathered by co-citation *affinity*.

Cohesive groups, finally, *represent densely-linked structures*. The former implies a communitarian gathering, where members tend to be individually and/or collectively more related to each other than with respect to other elements of the universe. The two basic operations for finding these groups are *extraction* (bottom-up detection) and *partitioning* (top-down detection). If we consider that not every element belongs to a structure that complies with the communitarian aspect, then the achieved coverage of the universe is partial (i.e., not every group is cohesive or nearly sectarian); however, if every generated partition is finally taken into account–regardless of its cohesion—then the coverage is total. In that sense, cohesive groups can be seen as lying between distinctive-feature (completely elitist) and common-trait groups (completely inclusive). As a closing remark, let us introduce Web communities as an example of this third group type.

2.1.3 Meaningful Group Detection Sub-Tasks

As it was stated earlier, the general—and, in a certain way, abstract—task of discovering meaningful groups consists of either finding lists of elements with distinctive features, groups whose members share certain traits, or dense structures. The first case is covered by *topic distillation*, the second one by *pattern-based clustering* (data clustering), and the third one by *community identification* (network clustering). We now briefly describe each of these tasks (depicted in Figure 2.3); moreover, let us note that a more concrete explanation will be given when reviewing their respective works.

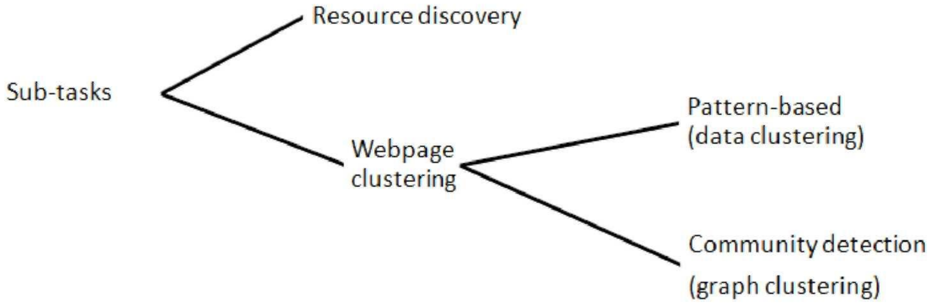


Figure 2.3: Meaningful group detection sub-task taxonomy

Topic distillation (resource discovery)

Topic distillation, on one hand, consists of finding quality documents on a query topic [13]; this typically includes identifying popular pages (*authorities*) and comprehensive lists of links to these pages (*hubs*). For example, for the query “Harvard University”, an authority could be its homepage `http://www.harvard.edu` and a hub could be a page that lists sites of universities. Topic distillation is also known as *resource discovery*.

Webpage clustering

In order to describe, on the other hand, community detection and pattern-based clustering, it results convenient to highlight several aspects of their common root: clustering.

Clustering, the most important *unsupervised learning task*, concerns the conformation of distinct groups (clusters) over an object collection—in very abstract terms. If the collection is viewed as a graph, this task is actually equivalent to the link mining object-related endeavor of *group detection* [48]. Unlike with classification and other supervised methods, the number and composition of the clusters is initially unknown.

To better understand cluster analysis (another name for the clustering discipline), a description of its most important types (in a broad sense) has been summarized in Table 2.3. From this table, the most prominent classification for our purposes is given by the *information management* criterion; in that aspect, grouping methods can be divided into pattern-based (usually known as *data clustering*) and network-based (commonly known as *graph clustering*).

Table 2.3: Clustering classification

Criterion	**Types**
Structure	**Flat**: Clusters are all at the same level. **Hierarchical**: Clusters are nested into more general ones.
Group membership	**Exclusive**: Partitional. Every element belongs to exactly one cluster. **Overlapping**: Non-exclusive. Elements can belong to one or more clusters. **Fuzzy**: Every element belongs to all clusters with a membership value ranging from 0 (does not belong) to 1 (completely belongs).
Coverage	**Total**: All elements are assigned to a cluster. **Partial**: Not every element ends up in a cluster.
Conformation approach	**Agglomerative**: Bottom-up. Starts with one cluster per element. **Divisive**: Top-down. Starts with all elements in a single cluster.
Randomness	**Deterministic**: No randomness introduced. The same result is always achieved given a data set. **Stochastic**: Randomness introduced. Implies running the clustering process several times to get the best results.
Information management	**Network-based**: Graph-theoretic methods are used for clustering. **Pattern-based**: Similarity between pairs of information patterns is used.

a) Pattern-based clustering (data clustering)

Pattern-based clustering (widely reviewed by Jain et al. [65]) consists of grouping objects according to their similarity. A pattern is usually understood as a *feature vector*. Similarity implies a certain kind of affinity, and can be represented, for instance, with Euclidean-space proximity (if taken as a distance, we more concretely talk about a *dissimilarity* between objects) or conceptual alikeness [95], just to mention a couple of ways. Two straightforward forms of grouping based on similarity concern either iteratively merging the most similar pair of objects (where objects could even be clusters) or defining cluster centroids and letting elements to be grouped "around" their nearest centroid. These methods represent the corner stone of data clustering (as simple as it may sound) and are best represented by the hierarchical and k-means approaches, respectively.

Admitting that the current section should aim to give just an overview of group detection sub-tasks, it is also important to realize that data clustering is indeed a very broad discipline. As a result, it seems convenient to describe as well, in a general way, the most outstanding methods that have been developed within this area. By doing the former, it should be easier to understand the specific methods of the next section, as they tend to be more elaborate and exhibit method combinations.

To start with, it seems relevant to note that all of the classical approaches share a common structure, which can be decomposed into four basic steps:

INITIALIZATION.- Make a first "guess" by providing random parameters , such as number of clusters, initial position of cluster centroids, membership of elements to centroids, etc., that allow the algorithm to start processing. This also implies constructing the necessary initial structures.

PROCESSING.- Approach to the solution by calculating actual distances, centroids, memberships, probabilities, etc.

UPDATE.- By taking the output of the previous step, re-calculate parameters that were initially assumed or that have changed.

REPETITION.- Repeat the processing and update steps until the algorithm converges or a termination criterion is satisfied.

As for the traditional approaches, there are basically four representative methods:

K-means.- It is probably the simplest and most popular data clustering algorithm; fixes k clusters *a priori*, assigns each data point to the nearest one of these clusters, and afterwards re-calculates new k centroids by taking the actual centers of the conformed groups [89]. The main goal is to minimize the squared error (SSE) with respect to cluster centers. Even though it has a good general performance, demands modest requirements, and is conceptually uncomplicated, results may vary depending on the selected number of clusters, the chosen distance metric, and the initial centroid positions. For this reason, the algorithm may have to be executed several times with distinct parameters.

Hierarchical.- Represents a popular alternative as well; it consists of creating a distance matrix and successively building clusters by merging the pair of closest objects at each step (this pair may contain two atomic elements, two clusters, or a cluster and an atomic element). The former can be done via different operations, such as single-linkage (takes shortest distance between elements), complete-linkage (takes greatest distance), or average-linkage (takes average distance). In contrast to other algorithms, which may operate in a more "obscure" fashion, the hierarchical technique allows to end up with a tree structure that explains how the clustering was done; this structure is called *dendogram* (Figure 2.4). A major drawback, however, is that hierarchical clustering is not scalable.

Fuzzy c-means.- The main contribution of this algorithm is its ability to generate non-exclusive clusters, since it designates a single data point to all clusters, but with a *membership value* that indicates *how much* the point belongs to a particular group. Can be seen as an extension of the traditional k-means algorithm; therefore, it is important to remark that it can vary according to initial values.

Gaussian mixtures.- This technique, which is inherently stochastic, reformulates the clustering process as the problem of recovering a probabilistic mixture distribution model that generated the given data points [118]. For achieving this purpose, it uses the Expectation-Maximization (EM) algorithm [99], which works in two steps (expectation and maximization, respectively); the expectation step calculates the probability of a data point being originated given a certain component i and the maximization step then refits the model's components to the data. By doing this, the *log likelihood* of the data increases at every iteration. This kind of clustering can be considered as "generative".

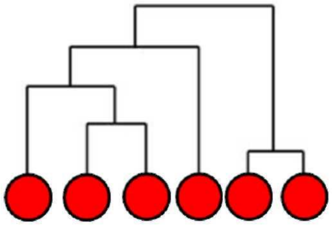


Figure 2.4: Example dendogram

Other representative clustering methods include (but are not limited to) the following:

Bisecting k-means.- Divisive variant of k-means that splits up the cluster with the least element similarity at each iteration (see description by Rossell [117]).

Kernel k-means.- Makes a non-linear mapping (kernel function) of the data to a higher-dimensional space, where points are more likely to be linearly separated [35].

Ant-based clustering.- Multi-agent solution that creates an analogy between finding the shortest path to a food source (ants individually mark with pheromones different ways and the most popular ones are followed by other ants, who also release pheromones, and this is repeated until the best path is discovered) and finding the shortest distance between documents [85].

Sub-space clustering Approach specially suited for high dimensional feature vectors; to make the space more manageable, it might transform or select a subset of features.

Density-based clustering (DBSCAN).- Detects regions of high density (concentration of data points within a certain radius) that lie among regions of low density.

A fundamental issue that has not been discussed yet is how structure mining can be carried out with pattern-based clustering. This issue takes place because this type of mining is usually thought of as related to form and organization; in that sense, network clustering seems the most intuitive task to carry out. However, pattern-based clustering is applicable as well to the Web (hyperlinked) domain if we are able to *transform link-related data* into a collection of features. Probably the most simple example of such conversion is given by an adjacency matrix, which actually represents a graph, but at the same time can be seen as an ordered set of feature vectors, where each vector corresponds to a row of the matrix. Similarly, common connections between nodes could be translated into patterns of co-citation and bibliographic coupling that can be clustered by any traditional algorithm. Obviously, these are not the only possible conversions; on the next section, other link-related similarities shall be discussed with more explicit detail.

Another important issue is that, except for a limited number of instances, data clustering is not very adequate for meaningful group detection in high-dimensional environments, such as the Web. The main issue of these approaches is that they heavily rely on element-to-element comparisons; this not only increases spatial complexity by the use of large affinity matrices, but also intensifies the time required for execution. For instance, while DBSCAN has shown to yield good results in dense environments, it has also proved to scale poorly as it performs a considerable amount of computations when having a lot of dimensions. Now, although graph clustering methods have been preferred for clustering Web collections nowadays, pattern-based approaches are far from falling into disuse. On the contrary, scalability issues have motivated a series of complexity management techniques (which we will see later). Furthermore, let us not lose from sight the fact that data clustering inherently represents a task different from network clustering, in the sense that it is more committed to finding common trait groupings—which finally can be considered as less elitist. Consequently, the data clustering task is more appropriate for certain applications, such as visualization.

b) Community identification (network/graph clustering)

Community identification involves finding sets of webpages that have more links to elements *inside* the set than *outside* of it [41] (which is actually equivalent to dense sub-graph extraction in graph mining [33]). These sets are called communities, and this kind of congregation usually indicates persons sharing common interests or webpages talking about the same theme. The previous definition is the one commonly accepted, but works such as the one presented by Radicchi have extended and specified it by providing formal statements and introducing communities with varying degrees of cohesion [114]; the two presented notions are the following:

Strong community.- Each member node has more connections within the community than outside it. Formally:

- $deg_{\text{int}}(v, \mathcal{C}) > deg_{\text{ext}}(v, \mathcal{C}), \forall v \in \mathcal{C}$

Weak community.- The community has, in general, more internal than external connections. Formally:

- $deg_{\text{int}}(\mathcal{C}) > deg_{\text{ext}}(\mathcal{C})$

Another more general definition, which comprises not only group members but also the *theme* that they revolve around, is given by Liu [83]. This definition consists of a tuple $C = (T, \mathcal{C})$, where C stands for the community, T represents the community's gathering theme (an event, concept, etc.), and $\mathcal{C}$ depicts the member set.

Yet another definition is the one given by Kumar et al. [77]; such definition is based on bipartite structures and was oriented towards explaining the discovery of emerging cyber-communities ("trawling").

A point of discussion regarding this task concerns the scope of its most common name: community identification. For social networks, (which this discipline partly inherits its foundations from) the phrase is completely coherent, as communities intuitively refer to people gatherings. For certain Web contexts (e.g. on-line collaboration networks), the name still makes sense, but if the communitarian group does not imply a social conglomeration, it could be misleading. Then, referring to this task as network or graph clustering when not dealing with a social context, seems more appropriate and neutral (of course, this merely concerns an opinion). Part of this discussion is to be retaken for topic mining.

For further reading on community identification in general (not only for the Web), see Puig's survey [113].

A final remark about data and graph clustering concerns elucidating how they can cope to contexts where they do not seem, apparently, to be the most intuitive option. For example, since the Web is by nature a graph, it seems more logical to apply graph clustering for its analysis; however, by turning the structure into a set of feature vectors, it is also possible to take advantage of pattern-based techniques, as we already said before. With regard to the opposite case (network clustering for data spaces), similarities among elements can be coded as weighted edges and, thus, graph-theoretic techniques can be used. The former (translating data to one format or the other) by itself concerns a discipline, and several techniques and algorithms have emerged as a result; some instances are the METIS [70], Chameleon [69], and Jarvis-Patrick [67] algorithms, which build similarity-based graphs.

2.1.4 Affinity measures, similarity matrices, and other computations

To discover groups, an affinity measure is usually needed, because it provides a way of relating the elements (even by simply indicating if they are connected or not), is able to state the degree of similarity that they have, and/or may be capable of evidencing how strong or cohesive the group as a whole is. In that sense, we may talk about two general types of affinity measures: *pairwise similarity* and *group quality*; in fact, this classification is shown in Figure 2.5. We now describe each type and discuss several of its metrics.

Pairwise similarity.- Affinity between pairs of elements. Usually used for data clustering.

Co-citation ($co(i, j)$).- Counts the number of documents that *cite* given a pair of other documents; introduced first in the bibliographic domain by Small [132]. See Eq. 2.10.

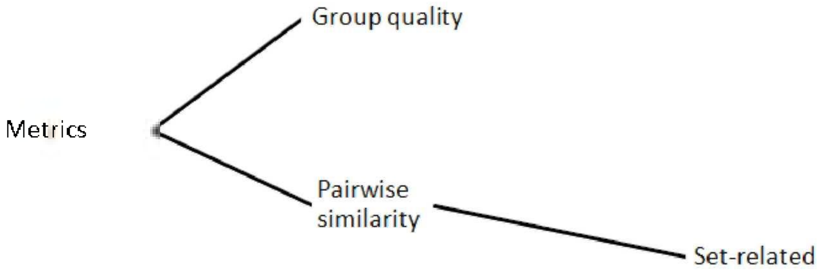


Figure 2.5: Meaningful group detection metric taxonomy

Bibliographic coupling ($b(i, j)$).- Counts the number of documents that *are cited* given a pair of other documents; introduced by Kessler [71]. See Eq. 2.11.

Structural similarity ($\sigma(i, j)$).- Uses neighborhood comparison to state how similar a couple of elements is (like the Jaccard coefficient). Introduced by Xu for creating cluster "cores" based on structure [149]. See Eq. 2.3.

Set-related metrics.- Generic measurements that state the similarity between a pair of sets I and J.

Jaccard index (set similarity).- Simple metric that compares the intersection size of two sets against their union size. Its complement $(1 - \text{Jaccard}(I, J))$ may be used as a set distance. See Eq. 2.4.

Dice coefficient.- Can be considered as a variation of the Jaccard index. See Eq. 2.5

Group quality .- Cohesion of elements gathered in a collection. Usually used for network clustering.

Conductance ($\phi(\mathcal{C})$).- Measures how "well knit" a community is by comparing the number of external links (cut size) against the minimum between the links from the cut to the complete graph and the links from the rest of the graph (cut complement) to the complete graph. See Eq. 2.6.

Local density ($\delta(\mathcal{C})$).- Graph theoretic measure that obtains the ratio of the *existing* edges versus the maximum number of *possible* edges [66]. See Eq. 2.9.

Relative density ($\rho(\mathcal{C})$).- Defines the proportion of internal links in a cluster $\mathcal{C}$ with respect to all of its links. See Eq. 2.8. Now, if we consider $A(S)$ as the denominator in the formula for conductance, we can in fact state that relative density is the complement of this quantity: $\phi(\mathcal{C}) + \rho(\mathcal{C}) = 1$ and $1 - \rho(\mathcal{C}) = \phi(\mathcal{C})$

Modularity (Q).- Can be seen as a network property; tries to measure the quality of the partition that has been made. As relative density, compares the number of internal and external links, but also incorporates the link expected value of the grouping in order to provide a fairer measurement. Was originally proposed by Girvan and Newman [104, 103]. See Eq. 2.7.

$$\sigma(i,j) = \frac{|\Gamma(i) \cap \Gamma(j)|}{\sqrt{|\Gamma(i)| \cdot |\Gamma(j)|}} \tag{2.3}$$

$$\text{Jaccard}(I,J) = \frac{|I \cap J|}{|I \cup J|} \tag{2.4}$$

$$\text{Dice}(I,J) = \frac{2|I \cap J|}{|I| + |J|} \tag{2.5}$$

$$\begin{aligned} \phi(S) &= \frac{\sum\limits_{i \in S, j \notin S} a_{ij}}{\min\left\{A(S), A(\bar{S})\right\}}, \\ A(S) &= \sum_{i \in S, j \in V} a_{ij} \\ A(\bar{S}) &= \sum_{i \in \bar{S}, j \in V} a_{ij} \end{aligned} \tag{2.6}$$

$$Q = \tfrac{1}{2m} \textstyle\sum_{ij} \left[a_{ij} - \frac{deg(i) \cdot deg(j)}{2m} \right], i,j \in \mathcal{C} \tag{2.7}$$

$$\rho(\mathcal{C}) = \frac{deg_{\text{int}}(\mathcal{C})}{deg_{\text{int}}(\mathcal{C}) + deg_{\text{ext}}(\mathcal{C})} \tag{2.8}$$

$$\delta(\mathcal{C}) = \frac{2deg_{int}(\mathcal{C})}{|\mathcal{C}|\,(|\mathcal{C}|-1)} \tag{2.9}$$

$$co(i,j) = |k : (k,i) \in E, (k,j) \in E| \tag{2.10}$$

$$b(i,j) = |k : (i,k) \in E, (j,k) \in E| \tag{2.11}$$

Since link information is naturally represented by graphs, it is common to use squared, $n \times n$ (or $n_s \times n_s$ if talking about a sub-graph) matrices for including adjacency or similarity information. Therefore, we will list some of the most common matrices used for link-based methods; it is important to mention that, because our aim here is to provide only a general idea, the variants (by "variant" we mean the cases for digraphs, DAG's, or weighted graphs) of these structures shall not be discussed in a thorough manner. Moreover, for each matrix, the contents will be illustrated formally.

Adjacency matrix.- Basic and well-known structure for representing graph edges (Eq. 2.12). Can vary according to the type of graph (i.e., the undirected case yields a symmetrical matrix, while the directed one does not). For weighted graphs, this matrix is conceived as the also well-known *weight matrix*:

Weight matrix.- Represents the costs associated to the connection of any node pair (Eq. 2.13).

Co-citation matrix.- Contains the co-citation indices for every pair of nodes in the graph; it is important to note that co-citation inherently works with directed graphs, which are given as input, but the computed matrix is symmetrical ($co(i, j) = co(j, i)$).

Laplacian matrix.- Is derived from the identity and adjacency matrices and is recommended, instead of the simple adjacency matrix, for calculating eigenvalues[125]. Consequently, it serves as a basis for spectral computations. See Eq. 2.15 and Eq. 2.16 for the normalized version.

$$a_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

$$w_{ij} = \begin{cases} 0 & \text{if } i = j \\ w \in \mathbb{R} & \text{if } \{i, j\} \in E \\ \infty & \text{otherwise} \end{cases} \tag{2.13}$$

$$\begin{aligned} c_{ij} &= co(i, j) \\ &= |k : (k, i) \in E, (k, j) \in E| \end{aligned} \tag{2.14}$$

$$L_{ij} = \begin{cases} deg(i) & \text{if } i = j \\ -1 & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

$$\mathcal{L}_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } deg(j) \neq 0 \\ \dfrac{1}{\sqrt{deg(i) \centerdot deg(j)}} & \text{if } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

Other useful computations involve recent trends that have been successful for finding communities in other types of networks and some resources from SNA. One such computation is *edge betweenness* (proposed by Girvan and Newman as well), which was designed for detecting community "peripheries"; this is accomplished by calculating the shortest path between all pairs of nodes in the graph and determining, for each edge, how many of those paths run through it [51, 104]. The philosophy behind this metric is that the edges with highest betweenness correspond to the boundaries of distinct communities, and these can be identified by removing those edges. Because this method focuses on the least central elements, instead of the most central ones, it has been considered as pretty innovative. Nonetheless, because it demands significant operations to be executed, this method is not suitable for every environment (has been used mainly in social and biological networks).

Regarding Social Network Analysis, a metric for defining cohesion in sub-groups is the previously mentioned "relative strength ratio" (see Eq. 2.17), which compares the absolute density of internal links against the density of external links and intends to evaluate the relative strength of a sub-graph.

$$\begin{aligned} rsr(\mathcal{C}) &= \frac{rs_i(\mathcal{C})}{rs_e(\mathcal{C})}\text{, where:} \\ rs_i &= \frac{deg_{\text{int}}(\mathcal{C})}{|\mathcal{C}|-1} \\ rs_e &= \frac{deg_{\text{ext}}(\mathcal{C})}{n-|\mathcal{C}|} \end{aligned} \tag{2.17}$$

Another relevant computation from SNA is *degree centrality* (Eq. 2.18).

$$C_D = \frac{deg(v)}{n-1} \tag{2.18}$$

2.1.5 Methods (concrete works)

If we were to classify the different methods for finding groups by their *modus operandi*, four different classes could be distinguished: pattern similarity-based, cut-based, spectral, and structure search-based. It results quite relevant to mention that works have been labeled according to the class that yields the best representation, or that best captures the overall approach; however, the elements used by each particular approach are not mutually exclusive. Also, it is important to keep in mind that all methods are *structural*; consequently, they all use related concepts. Moreover, while the current section is only committed to methods for the *Web*, we might occasionally include outstanding works from other network types.

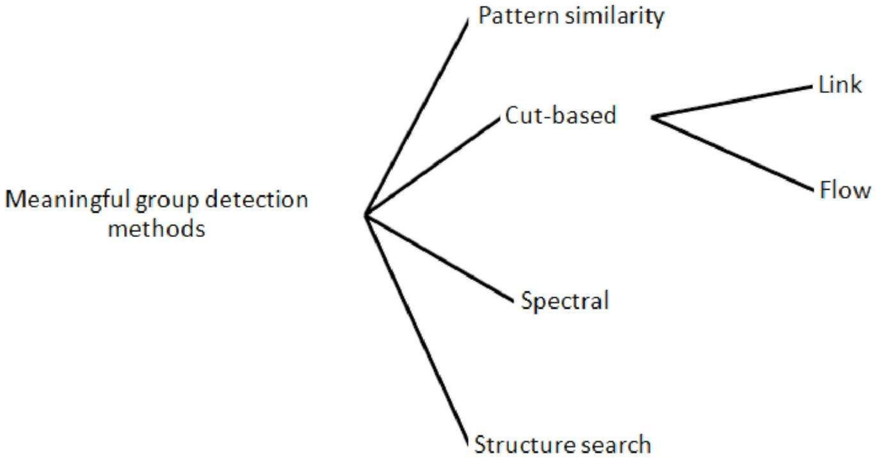


Figure 2.6: Meaningful group detection method taxonomy

Beginning with pattern similarity methods, these are completely aligned with pattern-based clustering and, thus, fix structural information into feature vectors; these vectors are usually grouped by using a classical technique or one of its variations. A clear representative of this method class is given by the approach of Modha and Spangler, that aims to cluster hypertexts based on an affinity measure that mixes textual and structural (in-links and out-links) alikeness using a weighted sum [98]; documents are grouped using *toric* k-means. Lying on this same line is Meneses's site clustering, which uses simple k-means with hyperlink cosine and Euclidean proximities [93]. Moreover, an original approach by Gibson et al. discovers cohesive groups in an efficient manner by combining the data stream paradigm with *shingling* [50]; the latter technique recursively finds sets of similar structures by creating "shingles", that is, fingerprints based on overlapping windows of information.

By analyzing the traits of these specific approaches, we can see that the pattern similarity class is more prone to the use of a *hybrid information source*. A pioneer effort within this area is given by Pirolli et al. [112], since their method mixes text, hyperlinks, and even weblogs to carry out categorization and prediction tasks.

Regarding *cut-based* approaches (a.k.a. *modularity-based*), they use graph partition cuts and algorithms that work with these structures. Methods concerning this type (according to works found in the state of the art) can be further divided into two sub-classes: link-count based and flow approaches. While the former are highly based on vertex and cluster degrees, the latter address the problem of finding groups as a *max-flow/min-cut theorem* issue.

The most important representatives of the link-count class are the approaches presented by Schaeffer, which propose local graph clustering [125] for coping with large scale graphs, such as the Web. In that sense, [140] and [124] discuss a fitness measure composed by local and relative density, which is to be maximized by local search methods in order to cluster webpages. Simulated annealing is the one used on these works, but the method itself is not limited to only this search strategy (this was studied by Meneses as well, where this type of clustering is done with genetic algorithms); in fact, the approach has been shown to work not only in the context of the Web graph, but also in routing problems [126]. A similar tactic is followed by [61] for graph visualization.

Regarding flow approaches, the most relevant work was developed by Flake et al.–whose basic claim is that the Web is a self-organizing structure. This work consists of a framework for finding communities with the aid of the maximum-flow/minimum-cut theorem [41, 42]. Here, artificial sink and source vertices are added to the seed graph (which is undirected) and the minimum cut is calculated by doing approximations with a focused crawler and augmenting the graph by using the Expectation-Maximization algorithm to reseed the crawler. Furthermore, a substantial contribution besides this algorithm consists of the basic notion of a Web community, which is defined as a "collection of webpages in which each member page has more hyperlinks within the community than outside the community". Respecting this idea, link count approaches such as the ones proposed by Schaeffer can be considered as a "relaxed" solution for community search.

The central advantages of cut-based approaches are, on one side, that they have graph-theoretic foundations and, on the other hand, that they embody purely link-based alternatives. Nevertheless, because they depend on certain techniques and algorithms they also become prone to the disadvantages of such strategies; this point is better illustrated by looking at each sub-class separately. For instance, regarding the discussed link count approaches, they lean on local search, which not always returns optimal results and is bounded by input parameters (cooling rate, precision thresholds, randomness, etc.). This last weakness is also shared by flow methods, since the algorithms that solve the maximum flow problem require networks to hold certain properties.

Despite the afore stated shortcomings, it is important to mention that cut-based approaches have other benefits. Concerning link count methods, they are intuitive, clear, and simple; contrary to spectral methods, they can be traced at any step of the process and need no interpretation to be understood. What is more, link count approaches are naturally adapted for the Web size.

The third type of link methods has its foundations on *spectral graph theory* (for more information on this matter, see [30, 29, 142]), which includes elements like matrix decomposition, eigenvalues, random walks, laplacians, etc. It seems convenient to state that, despite they can be considered as a sub-class of pattern similarity methods, spectral approaches have been placed as a separate category because of their great prominence in the Web meaningful group detection domain.

Probably the most important work regarding this type of methods is the HITS (Hypertext Induced Topic Search) algorithm [26, 73, 25], which was designed to fulfill the topic distillation task by providing resources that will serve as a guide for a *broad topic search* (a search that uses general terms like “leukemia”, “jaguar”, etc.). This algorithm is composed of three basic steps [86]: 1) assemble a target subset of webpages, 2) compute the principal eigenvectors to form the vector hub scores and authority scores, and 3) output the top-scoring hubs and authorities.

Kleinberg’s algorithm and research have been a seminal contribution, which has constituted the basis for other works; one example of such works is the one done by Gibson [49], where HITS is applied with the purpose of identifying community “cores” (which consist, not surprisingly, of authoritative pages). The former approach claims to be able to find not only a single community given a user query, but a small set of “non-principal” communities, that arise due to word ambiguity and because certain sub-topics for the main theme may show to have their own communities as well.

Other works included in this category are [60], which uses random walks to find communities and [57], which identifies topics by combining a series of methods, such as recursively using a spectral graph partitioning method and constructing an affinity metric from different sources (text, hyperlinks, co-citations). Later, HITS is used as a post-processing algorithm. An additional attempt is the one given by [106], which approximates Fiedler vectors to find clusters in large graphs.

An obvious advantage of using spectral methods is that these constitute a solid, clean, mathematic approach; moreover, as we can observe from the discussed works, the utilization of such methods has proven to yield important results in Web structure mining. The main drawback of approaches that rely on spectral theory is that they are not scalable *per sé*, because they usually involve matrix computations, and the size of these matrices for the Web simply cannot be handled; as we will see on the next section, possible solutions that have been adopted for this issue include building a matrix only for pages that respond to a certain query and using approximations. Also, another disadvantage (which is minor, since it does not necessarily affect in all contexts) is that spectral operations are difficult to trace, because they tend to be “obscure” and require interpretation [113].

Structure search methods look for certain patterns in graphs, and basically involve detecting various special kinds of sub-graphs, such as bipartite sub-graphs and components with particular features. Some early attempts concerning this kind of methods are the ones developed by Botafogo [19, 18, 17](not precisely tested for the Web, but rather for stand-alone hypertexts). In these works, a divisive clustering strategy is followed by iteratively removing edges from special nodes in the graph (namely, index and reference nodes, which can be seen as analogous to hubs and authorities) for finding biconnected components (sub-graphs for which every two nodes have two paths between them); these, in turn, are furtherly decomposed into strongly connected components.

On the other hand, Kumar presents the "trawling" process for identifying emerging communities [78]; the basic assumption is that communities are characterized by dense directed bipartite graphs. Therefore, the "cores" of these structures must be detected and this is done by an *elimination-generation* algorithm, which iteratively looks for cores of a fixed size and eliminates candidates that either do not belong to the core or whose degree lies below the threshold. An important feature of this algorithm is that it is able to process the *entire Web graph*.

Probably the main positive aspect of structure search methods is that they use conceptually known kinds of graphs and there exist established methods for finding these structures in an efficient manner. Also, since the general task is to look for certain patterns, it is possible to incorporate different methods (including some of the ones discussed before) to accomplish this purpose. However, a drawback to this type of approaches is that it can tend to be strict—thus, if the structure is not found, no results are returned. This can be solved by relaxing structures, but in turn can exhibit additional complications.

A complementary classification of link-based methods can be made by taking into account the view of the search space; this results in *global* and *local* approaches—the first exploring all the search space and the second only local neighborhoods. Even when this type of classification seems logical, actually the limits of one approach versus the other are not so neat. This is because certain methods claim to be global, but operate using local views of the Web. In that sense, only the works by Schaeffer and Kumar are clearly stated as local and global, respectively.

2.1.6 Overcoming complexity

An absolutely relevant issue respecting the WWW is that its processing is computationally expensive; therefore, algorithms are expected to be able to deal with high time and memory requirements. We will now discuss some strategies (which are not necessarily mutually exclusive) used by the aforementioned link-based methods to overcome complexity; an attempt of structuring these strategies into a taxonomy is given by Figure 2.7.

"On-demand" or focused crawling.- Several methods tend to start with a very small *root set* or *seed*, which is later expanded into a larger set (sometimes called "base set"). The root set is obtained by querying a search engine and crawling only the resulting pages or simply by manually inputting seed sites to the algorithm; the expansion is usually done by crawling as well the root set's linking pages or crawling for a fixed depth (departing from the root). By doing this, the methods are allowed to focus on a specific portion of the Web and are able to reduce the problem dramatically (for instance, HITS considers a base set to be of maximum 5,000 pages when the whole WWW contains billions of webpages). Recurring to this kind of strategy is convenient for applying matrix methods (such as spectral clustering) and other algorithms that,

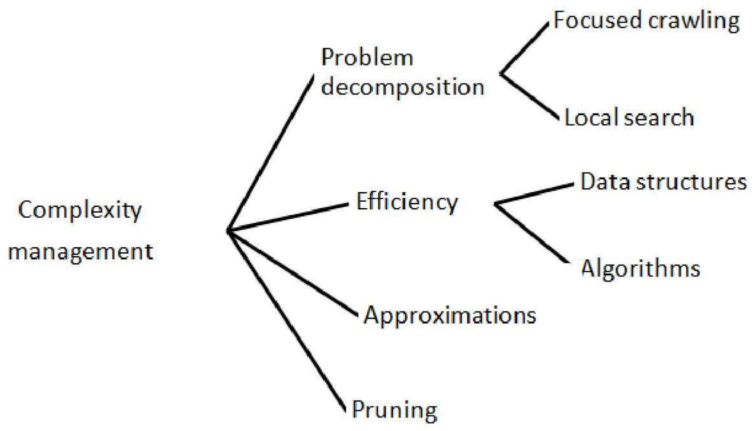


Figure 2.7: Complexity management taxonomy.

even when not scalable, are well established and have been shown to yield high quality results.

Local search.- This technique is related to the previous strategy, since it deals only with *partial views* of the graph [125]; also, since the search is done over neighborhoods, it is not necessary to store the adjacency information of the whole Web graph in a data structure (it is sufficient to retrieve adjacency lists for the required vertices). Furthermore, since there exist algorithms that calculate connected components in linear time, it is possible to apply local methods only to selected graph components.

Efficient data structures and algorithms.- Structures like heaps and balanced binary trees can be used for faster data access; also, constructing special structures (such as sparse matrices) for reducing memory consumption is useful. Moreover, avoiding needless calculations is fundamental for dealing with data sets of a considerable magnitude.

Approximations.- Heuristics and methods that yield approximate answers are also valuable for tackling the problem of complexity; the Web is not an exception. For example, [106] uses this type of calculations to apply spectral methods.

Pruning.- Aggressive elimination of unnecessary or less important material (i.e., duplicate pages, media that will not be used, HTML tags, etc.) and data that no longer remains vital has also been crucial, specially for global methods.

Parallelization.- Designing algorithms that can be distributed over several computers concerns also a feasible alternative. For instance, the graph shingling algorithm [50] is specially suited for the "data stream paradigm".

It could be stated that these different strategies encompass three general ideas: decompose the problem (and optionally solve only some parts), approximate the answers, and eliminate dispensable material. In fact, decomposing the problem and eliminating dispensable material is also addressed by subspace clustering [109] (this type of clustering, as we had seen, encompasses a series of techniques but can be summarized as focusing only on a subset of attributes) and dimensionality reduction (explained in several texts, such as [79] and described in [24] for the Web context).

2.1.7 Evaluation techniques

When it comes to evaluating results, techniques for group detection usually fall into one of three categories: internal, external, and relative [137] (this basic classification scheme is portrayed in Figure 2.8). Each type is to be described and some examples are being provided as well.

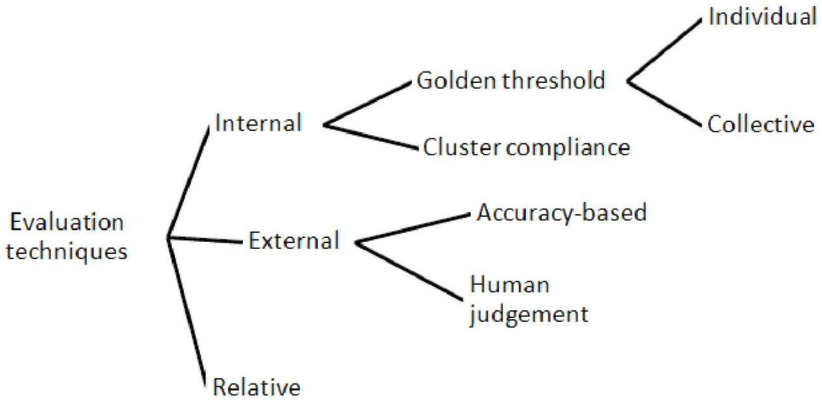


Figure 2.8: Cluster evaluation taxonomy

Internal evaluation is based on the groups' *intrinsic properties*; in that sense, an external source is not needed for assessing result quality. Two principal methods can be distinguished for this first class: the "golden threshold" and cluster compliance.

As for the golden threshold, a group (or the whole grouping, in some cases) is evaluated according to a *collective quality measure* (such as the ones we observed earlier); if the achieved quality is superior to a given threshold (usually determined by an external authority or taken as convention by the research community), the evaluation is positive. Otherwise, the group is considered as having a low quality. For instance, Clauset et al. claim that achieving a modularity above 0.3 indicates a fair graph partition [32]; similarly, Wasserman and Faust present 1.0 as a relative strength ratio threshold value for proving a considerable cohesion among members of the same group [145]. Another possibility (although it has not been used for evaluation purposes yet) is given by Raddicci's community classification; consequently, groups can be validated based on whether they comply or not with the definition of a community (strong or weak, depending on the case). Now, the overall quality of the detected groups can be taken as the average attained quality; if the chosen metric evaluates the grouping, it can logically be taken as is.

Like we can observe, evaluation with group quality thresholds is relatively simple. Moreover, it allows to either evaluate *individual clusters* or the *whole clustering*. However, it is also subjective to some extent (and not everyone might agree about the cut-off values or the metrics themselves), sensitive to the characteristics of the corpus (e.g. a certain metric might behave more strictly or more indulgently depending on the size of the graph or its sparseness), and there is not always a golden threshold to use.

If golden thresholds show to be unfit for evaluation, another internal option is *cluster compliance*—that is, ratifying that the members of each group are more similar among themselves than with respect to the other groups. Such validation scheme is inherited from data clustering and is known as *intra-cluster compactness versus inter-cluster separation*. This concept is visually displayed in Figure 2.9 and presented in general terms by Equation 2.19. As we can see from the equation, compactness is evaluated by measuring the proximity between pairs of elements within the same cluster; analogously, separation results from measuring the proximity between pairs of elements belonging to distinct clusters. It is important to mention that *proximity* stands for any similarity or dissimilarity function (or a combination of them). In that aspect, we can align this type of evaluation with pairwise affinity metrics.

$$\begin{aligned} \text{compactness}(\mathcal{C}_i) &= \textstyle\sum_{x,y\in\mathcal{C}_i} \text{proximity}(x,y) \\ \text{separation}(\mathcal{C}_i,\mathcal{C}_j) &= \textstyle\sum_{x\in\mathcal{C}_i,Y\in\mathcal{C}_j} \text{proximity}(x,y) \end{aligned} \tag{2.19}$$

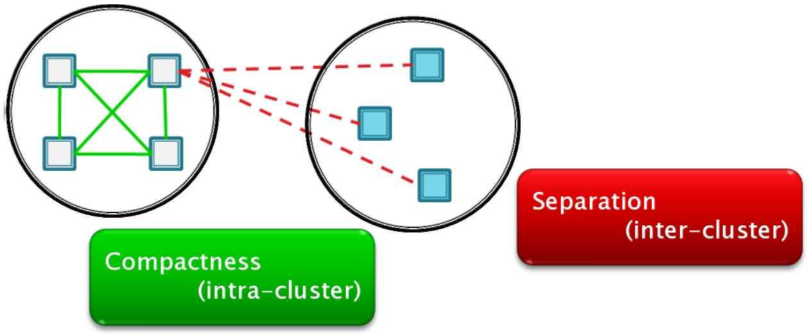
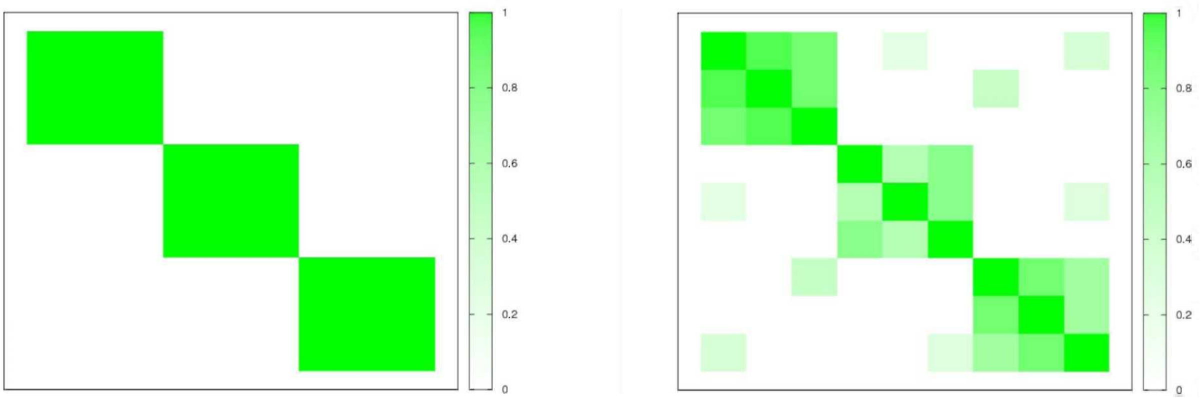


Figure 2.9: Compactness and Separation

Regarding overall quality, it can be obtained by, once again, taking the average compactness and separation (a variant is given by additionally weighting results with respect to cluster sizes) of the clustering; even when both metrics can be obtained individually and then contrasted, there exist validity indices that measure compactness and separation together (e.g. the Dunn Index, which is explained in the survey by Kim and Ramakrishna [72]). Another commonly accepted way of displaying cluster compliance results is by means of *proximity matrices*. To build these matrices, elements are first ordered in such a way that the ones belonging to the same cluster are contiguous to each other; similarity (or dissimilarity) is then depicted by the tonality of each individual cell, where a darker tone generally indicates a higher resemblance. Because in an *ideal* matrix of this type, elements should have a similarity of 1 with respect to members of their cluster and of 0 with respect to elements of other clusters, it is expected for "good" clusterings to exhibit a *block-diagonal pattern* (the stronger, the better). Note that in the case of a dissimilarity matrix, tones are inverted; as a result, the block-diagonal is noted by a lighter pattern. Examples of proximity matrices are shown in Figure 2.10.



(a) Ideal matrix

(b) Actual matrix example

Figure 2.10: Proximity matrices.

Even when cluster compliance seems the most straightforward type of internal evaluation, it is also true that pairwise comparisons can be very expensive if there is a massive amount of data. On the side of proximity matrices, they can be seen as a powerful visualization tool for results, particularly when a general, quick overview is desired. However, they also present issues to overcome; for example, when having clusters of varying sizes, it might be necessary to take samples to make the clusters more uniform. At the same time, these matrices are also fond to subjectiveness, as an acceptable standard diagonal pattern (i.e., that surpasses a certain proximity value on average) still remains undefined. With regard to this last aspect, it seems that the sole presence of such pattern suffices for a positive evaluation; consequently, the intensity of such pattern is left as domain-dependent.

While internal evaluation is indeed helpful for gaining insight on results, it is usually the least preferred type, specially when having information available for an external or relative assessment.

On the other hand, external validation techniques count with a pre-established *result model*. Conventionally, this model is conformed by a *set of reference classes*, and quality is assessed by the use of *accuracy-based* metrics. In order to explain these metrics, let us provide some notation and definitions as well.

To start with, consider a cluster $\mathcal{C}$ that belongs to a clustering $\mathbb{C}$ ($\mathcal{C} \in \mathbb{C}$) and a reference class $\mathcal{R}$ that belongs to the aforementioned reference class set $\mathbb{R}$ ($\mathcal{R} \in \mathbb{R}$). From an information retrieval point of view, $\mathcal{C}$ is the *retrieved set* (that is, the set obtained by means of a given detection method) and $\mathcal{R}$ is the *relevant set* (the "real" group). Then, we have three basic external evaluation metrics:

Precision (p)**.-** Fraction of the cluster that actually belongs to the reference class (see Eq. 2.20). Stands for correctness.

Recall (r)**.-** Fraction of the reference class that was actually placed in the cluster (see Eq. 2.21) Stands for completeness.

F-score (F)**.-** Aims to balance precision and recall by combining them into a single metric. Traditional F-score (a.k.a. F_1) gives both precision and recall the same importance;

however, it is not uncommon to find variations that award one or the other a greater prominence (F_2, $F_{0.5}$).

Another important issue regarding precision and recall is that these two quantities are usually not seen in isolation; therefore, it is common to graph both scores on a *Precision vs. recall curve*, in which the precision for a cumulative number of "observed" (recalled) documents is recorded. For instance, let us consider that ten documents have been retrieved; if the first one of the documents is relevant, then our precision is 1.0 at 10% recall. But, if the next two documents fail to be relevant, our precision will be lower at 30% of recall, on such luck that the curve will fall on the average achieved precision; consequently, these graphs for the usual look like decreasing (e.g., see Figure 2.11). Therefore, the highest the last precision score is at the last level of recall (11 standard recall levels are commonly used), the better a result can be considered. A similar curve can be obtained by placing F-scores in a decreasing order and plotting them against a cumulative percentage of groups (0-100%).

$$p = \frac{|\mathcal{R} \cap \mathcal{C}|}{|\mathcal{C}|} \tag{2.20}$$

$$r = \frac{|\mathcal{R} \cap \mathcal{C}|}{|\mathcal{R}|} \tag{2.21}$$

$$F = \frac{2}{\frac{1}{p} \blacksquare \frac{1}{r}} \tag{2.22}$$

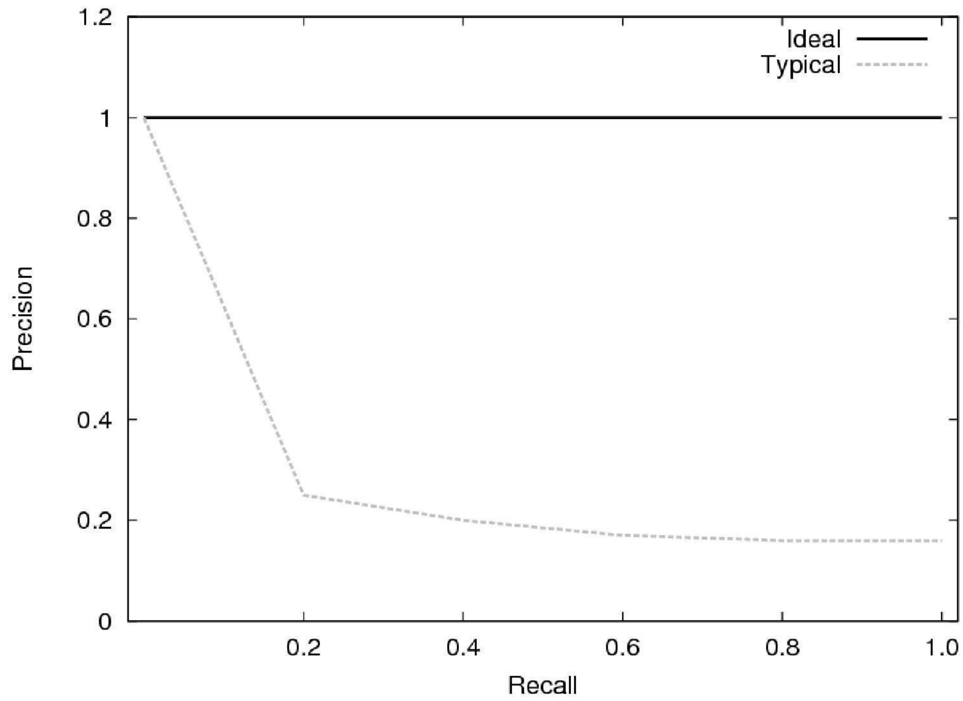


Figure 2.11: Precision vs. recall curve

It is also possible to express the previous metrics in classification terms; in that sense, we can distinguish four quantities, namely $f_{00}, f_{01}, f_{10}, f_{11}$:

□ f_{11} is the number of pairs of objects that are placed in the same class both in $\mathbb{R}$ and $\mathbb{C}$ (*true positives*)

□ f_{01} is the number of pairs of objects in the same cluster in $\mathbb{C}$, but not in the same class in $\mathbb{R}$ (*false positives*)

□ f_{10} is the number of pairs of objects in the same class in $\mathbb{R}$ but not in $\mathbb{C}$ (*false negatives*)

□ f_{11} is the number of pairs of objects in different classes and different clusters (*true negatives*)

As a result, precision and recall can be seen as in Equations 2.23 and 2.24, respectively. In fact, recall represents the *sensitivity* statistical measure (true positive rate) within this context.

$$p = \frac{f_{11}}{f_{11} + f_{01}} \tag{2.23}$$

$$r = \frac{f_{11}}{f_{11} + f_{10}} \tag{2.24}$$

This alternative conception for precision and recall also allows us to introduce another common external metric, the *Rand Index* [116]; in its simplest form, it measures the *accuracy* of a partition—that is, disjoint clusters—and provides a succint overview of how "good" it was. The conventional Rand Index is shown in Equation 2.25.

$$R = \frac{f_{11} + f_{00}}{f_{00} + f_{01} + f_{10} + f_{11}} \tag{2.25}$$

Other, probably less used, external evaluation metrics include*purity* and *entropy* (information-theoretic quantities).

Because group detection commonly implies not knowing classes *a priori*, obtaining a reference set may, in some instances, be a complicated endeavor. To address this situation, several testbeds have been designed for method evaluation; the TREC collection and the Reuter's corpus concern two examples (although these are for content-based approaches).

Another form of external validation concerns *human judgment*. One form of human evaluation implies comparing the obtained groups against known *actual* groups; for instance, in Kubica's work [76], communities found for a test set consisting of webpages regarding researchers' interests were shown to correspond to the institutes's research groups. The same validation approach was used in [81]. Related to the former is manual inspection, which can be used to examine the groups and assess qualitative properties; this kind of validation is suitable for very small collections or subsets of Web documents, but when dealing with big groups, this alternative clearly becomes unfeasible. However, it can still be possible to manually validate numerous groups by choosing representative samples.

External evaluation, in broad terms, results extremely convenient for comparing results against "ground truth". Moreover, it not only serves for cohesive and common-trait clusters, but also for distinctive-feature groups, as it is independent from their properties. Nevertheless, as already stated, a result model (users, reference classes) is required; considering that the model tends to be static, it can undermine the *discovery* of non-trivial groups. Finally, external metrics could additionally exhibit some degree of subjectiveness if we take into account that there is not a standard precision or recall "satisfactory" level; the former can be particularly sensitive when approaching complex contexts where, probably, very low scores could be acceptable. In that sense, a comparison between approaches could be more adequate.

Relative evaluation and other schemes

Relative evaluation consists of comparing different approaches, according to certain criteria. It is important to state that this evaluation is not by itself a different kind of evaluation, since it can use either internal or external metrics as a base for comparison [137]. For example, quality can be assessed by asking users to rate competing methods; also, the Rand Index can be used for stating the similarity between a pair of partitions obtained by different methods.

Yet another form of evaluating results is by means of an *indirect evaluation*, which consists of assessing a method's effectiveness by observing how well it leverages a certain *primary task*. In that sense, clustering or community detection might not be the ultimate target, but perhaps a more user-oriented activity, such as browsing. This is specially useful for comparing methods more "neutrally".

So far, the evaluation techniques have been explained in terms of the most conventional clustering scheme: a partition. However, there exist evaluation methods that are exclusive for hierarchical or fuzzy groupings; several of these metrics are explained in the survey by Halkidi et al. [55].

2.1.8 Summary

Because an extensive discussion has been carried out for the current section, it seems appropriate to provide a summary as a means for closure. First, Table 2.4, presents a review of the covered group types and a possible alignment among the different concepts that were explained along the section; please note that, while the most common combination of such concepts is portrayed on the table, we do not discard the existence of different combinations.

Table 2.4: Alignment of meaningful group detection concepts.

Group type	Relationships	Operation	Associated sub-task	Metrics	Coverage	Methods
Distinctive feature	Not necessarily related	Ranking, selection	Resource discovery	Centrality	Partial	Spectral
Cohesive	By cohesion	Extraction, partition	Community detection	Group quality	Total / partial	Structure search, cut-based, spectral
Common trait	By similarity	Mapping	Data clustering	Pairwise	Total	Spectral, pattern-based

Finally, a summary with the most relevant methods and their features according to the established taxonomy is illustrated in Table 2.5.

Table 2.5: Meaningful group detection methods

Author	Contribution / Method	Method type	Group type	Task	Complexity management	Evaluation
[18] Botafogo (1992)	Aggregates and bi-connected components	Structure search	CH	NC	-	-
[32] Clauset et al. (2004)	Communities in very large networks	Cut-based: Link count	CH	NC	Efficient data structures (EF)	-
[41] Flake et al. (2000)	Maximum-flow communities	Cut-based: Flow	CH	NC	Focused crawling (PD), approximations	Manual (E)
[42] Flake et al. (2002)	Maximum-flow communities	Cut-based: Flow	CH	NC	Focused crawling (PD), approximations	External
[49] Gibson et al. (1998)	Communities with HITS	Spectral	CH	NC	Focused crawling (PD)	-
[50] Gibson et al. (2005)	Graph shingling	Pattern-based	CH	DC	Parallelization (PD)	Manual (E)
[60] Huang et al. (2006)	Communities with random walks	Spectral	CH	NC	Focused crawling (PD), efficient methods (EF)	Intuitive comparisons (R)
[76] Kubica et al. (2002)	Stochastic blockmodeling	Pattern-based	CH	DC	Focused crawling (PD)	Manual (E)
[78] Kumar et al. (1999)	Communities by trawling	Structure search	CH	NC	Pruning	Manual (E)
[73] Kleinberg (1999)	HITS	Spectral	DF	RD	Focused crawling (PD)	-
[93] Meneses (2006)	Clustering of Central American sites	Cut-based: Link count, pattern-based	CT, CH	NC, DC	Focused crawling (PD)	Rand Index (R)
[98] Modha & Spangler (2003)	Hypertext clustering for Web search	Pattern-based	CT	DC	Focused crawling (PD)	Intuitive
[124] Schaeffer (2005)	Clustering of massive graphs	Cut-based: Link count	CH	NC	Local search (PD)	Similarity matrices (I), comparisons (R)
[140] Virtanen (Schaeffer) (2003)	Clustering of the Chilean Web	Cut-based: Link count	CH	NC	Local search, focused crawling (PD)	Comparisons (R)
[148] Wu et al. (2004)	Automatic Topic Discovery (ATD)	Spectral, cut-based	DF	RD	Focused crawling (PD)	User ratings (R)
[151] Zhu et al. (1999)	PageCluster	Pattern-based	CT	DC	Focused crawling (PD)	User tasks (IN) and relative

DC=Data clustering, NC=Network clustering, RD=Resource discovery
CH=Cohesive, CT=Common trait, DF=Distinctive feature
PD=Problem decomposition, EF=Efficiency, E=external, I=Internal, R=Relative, IN=Indirect

2.1.9 A brief overview of Web content mining

For the sake of completeness and an understanding of the upcoming chapters, we will briefly describe some content-related aspects (specifically regarding text). These aspects correspond to: text normalization, general text-based methods, and the Vector Space Model. Since the first two require a less broad explanation, they will be discussed next.

With respect to text normalization, it normally involves a number of preprocessing steps, like word tokenization, stemming (taking words with the same lexical stem as equal), and common term (stopword) removal.

Even when conventional data clustering techniques can be used for grouping based on text, there are methods that have been created more specifically for managing this type of information, e.g. NLP-based clustering (suffix trees) and query grouping. A comprehensive description of these and other methods is given in the review by Andrews and Fox [7].

Let us recall that our emphasis is on unsupervised methods (clustering); however, Web content mining in general can be accomplished by using supervised and/or semi-supervised techniques as well.

The Vector Space Model

A practical form to represent documents for clustering and retrieval is by considering them as "bags of words" (that is, term containers where word ordering is irrelevant), and thus as vectors composed of weighted terms. This vector has all vocabulary words as components and weights are calculated according to term importance measures, usually *term frequency* and *inverse document frequency* (a.k.a. "tf-idf" [121]); this document conception is known as the Vector Space Model [120, 122]. The notation for this model is given by Table 2.6.

Table 2.6: Vector Space Model notation.

$\|K\|$	Number of (unique) terms in a corpus C.
$K = \{k_1, \ldots k_t\}$	Vocabulary terms.
k_i	A generic term.
d_j	A document inside C.
$w_{i,j}$	Weight of term k_i in document d_j.
$\vec{d_j} = (w_{1,j}, \ldots w_{t,j})$	A weight vector associated to d_j.
N	Number of documents in C.
n_i	Number of documents where term k_i appears.
$g_i(\vec{d_j})$	Function that returns the weight $w_{i,j}$ for term k_i in doc. d_j.
$f_{i,j}$	Number of times term k_i is mentioned in the text of doc. d_j.
$tf_{i,j}$	Text frequency for k_i in d_j.
idf_i	Inverse document frequency for k_i.

In principle, since every document can be represented by a term vector, a document-term matrix may be constructed to visualize this information (in fact, techniques such as Latent

Semantic Indexing [44] and spectral clustering heavily rely on this matrix). Equation 2.26 shows this structure.

$$\mathbf{D} = \begin{bmatrix} w_{0,0} & \cdots & w_{|K|,0} \\ \vdots & \ddots & \\ w_{0,N} & \cdots & w_{|K|,N} \end{bmatrix} \tag{2.26}$$

Regarding keyword weights, these can be calculated in different forms; however, the usual scheme is to assign weights according to *tf-idf* (term frequency-inverse document frequency). This scheme consists of balancing the importance of keywords inside specific documents against their overall popularity (that is, their frequency in the whole document collection). For example, consider a corpus that contains documents that deal with places, consider that we have a document that talks about a *beach*, and consider that we have three keywords, “tan”, “location”, and “sand”. Concerning the first keyword, let us assume that it occasionally appears on our beach document (has a low text frequency) and it is sporadically found along the rest of the corpus (has a high inverse document frequency). On the other hand, a word like “location” would have a low idf if we assume that it probably appears on almost every document of the corpus and a high tf, since it might be a popular word inside individual texts as well. Finally, we would expect for a keyword like “sand” to be awarded the highest weight, because it can be seen as substantial for the individual document and also has a low popularity (high idf) with respect to the corpus. Equations 2.27 and 2.28 show how to calculate tf and idf, respectively, and Eq. 2.29 depicts this formula as a whole.

$$tf_{i,j} = \frac{f_{i,j}}{\sum\limits_{i=1}^{i=|K|} f_{i,j}} \tag{2.27}$$

$$idf_i = \log \frac{N}{|n_i|} \tag{2.28}$$

$$\begin{aligned} w_{i,j} &= tf_{i,j} \times idf_i \\ &= \frac{f_{i,j}}{\sum\limits_{i=1}^{i=|K|} f_{i,j}} \times \log \frac{N}{n_i} \end{aligned} \tag{2.29}$$

A very important aspect of the vector space model is that it allows to apply similarity metrics for correlating pairs of documents. Probably the most common of these metrics is the *cosine similarity* (Eq. 2.30), which calculates the “cosine of the angle” between two documents [9]; a similarity of 1 indicates that the documents are identical, while a value of 0 indicates no relation. Other similarity measures can be found in [54].

$$\begin{aligned}\text{cosim}(d_a, d_b) &= \frac{\vec{d_a} \cdot \vec{d_b}}{\left|\vec{d_a}\right| \times \left|\vec{d_b}\right|} \\ &= \frac{\sum\limits_{i=1}^{i=t} w_{i,a} \times w_{i,b}}{\sqrt{\sum\limits_{i=1}^{i=t} w_{i,a}^2} \times \sqrt{\sum\limits_{i=1}^{i=t} w_{i,b}^2}}\end{aligned} \tag{2.30}$$

To illustrate the vector space model and cosine similarity, let us take a small corpus that only consists of four documents (d_1,d_2, d_3, and d_4) and a vocabulary of four terms ("cat", "coffee", "sugar", "dog"). We begin by describing each document in terms of its vocabulary and the number of occurrences of each keyword:

$$\begin{aligned}d_1 &= \{(\text{sugar}, 3), (\text{coffee}, 2)\} \\ d_2 &= \{(\text{cat}, 4), (\text{dog}, 8)\} \\ d_3 &= \{(\text{cat}, 5)\} \\ d_4 &= \{(\text{coffee}, 6)\}\end{aligned}$$

Let us recall that, even though there are documents that do not contain the whole vocabulary, in order to build the vectors, every corpus keyword is taken into account. Therefore, a vector document $\vec{d_j}$ is considered as $\vec{d_j} = (w_{1,j}, w_{2_j}, w_{3,j}, w_{4,j})$, where k_1 ="cat", k_2 ="coffee", k_3 ="dog", k_4 ="sugar", (terms are usually arranged in lexicographic order). So, for instance, let us calculate the weight for term k_4 in d_1 and then show the contents of $\vec{d_1}$:

$$\boxed{\begin{aligned}w_{4,1} &= \tfrac{3}{5}\ln\left(\tfrac{4}{1}\right) \\ &= 0.6\ln(4) \\ &= 0.6\,(1.386) \\ &= 0.832 \\ \\ \vec{d_1} &= (0, 0.277, 0, 0.832)\end{aligned}}$$

Similarly, we calculate vectors $\vec{d_2}$, $\vec{d_3}$, and $\vec{d_4}$ to build the document-term matrix:

	cat	coffee	dog	sugar
d_1	0	0.277	0	0.832
d_2	0.227	0	0.927	0
d_3	0.693	0	0	0
d_4	0	0.693	0	0

As we can see, by constructing the document-term matrix, the cosine calculation among documents is almost straightforward. For instance, returning to our example, consider obtaining cosine similarity for document vectors d_1 and d_3, and for vectors d_1 and d_2. As expected, the result between intuitively similar documents (d_2 and d_3) is higher:

$$\begin{aligned}
\text{cosim}(d_1, d_2) &= \frac{(0)(0.227) + (0.277)(0) + (0)(0.927) + (0.832)(0)}{\sqrt{0^2 + 0.277^2 + 0^2 + 0.832^2} \times \sqrt{0.227^2 + 0^2 + 0.927^2 + 0^2}} \\
&= \frac{0}{0.877(0.955)} \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\text{cosim}(d_2, d_3) &= \frac{(0.227)(0.693) + (0)(0) + (0.927)(0) + (0)(0)}{\sqrt{0.227^2 + 0^2 + 0.927^2 + 0^2} \times \sqrt{0.693^2 + 0^2 + 0^2 + 0^2}} \\
&= \frac{0}{0.955(0.693)} \\
&= 0.238
\end{aligned}$$

2.2 Topic mining - identification - detection - discovery - extraction

When dealing with collection topics and their automatic identification, we are confronted with a wide variety—a “soup”, colloquially speaking—of representations, approaches, tasks, and information sources that, even when highly intertwined among each other, seem also to fall into distinct categories. The former reveals a couple of aspects: on one hand, realizing that the *topic identification* concept is certainly vague and, as a matter of fact, acts more like an umbrella that covers a considerable number of other concepts and methods. On the other hand, building a crisp taxonomy to classify works related to this area results in a non-trivial endeavor. In that sense, let us disclose the aim of the upcoming review by exposing its conductive string.

First, topic mining[2] is fed by a variety of *information sources*, which in turn may lead to different topic *representations* or embodiments when treated according to a given mining *task*. Then these factors, together with a chosen *learning type*, make up specific *approaches* for detecting topics inside collections. However, as previously stated, not everything is crystal clear when talking about such domain, and this orients us towards wrapping up the subject with a topic mining issue *discussion*.

Let us clarify that, similar to the meaningful group section, specific works are to be discussed when arriving to the method type; the main reason for such decision is that we consider this division criterion as the most delicate one, and probably the most adequate classification for understanding works. Nevertheless, unlike the previous section, which was developed with more technical detail, we will treat the upcoming discussion in a more conceptual form. Like

[2] In the most abstract sense, topic mining could be defined as a discipline that consists of *figuring out* the existing thematics of a corpus. Broadly speaking, this includes *finding the names* of the topics or *conforming* them.

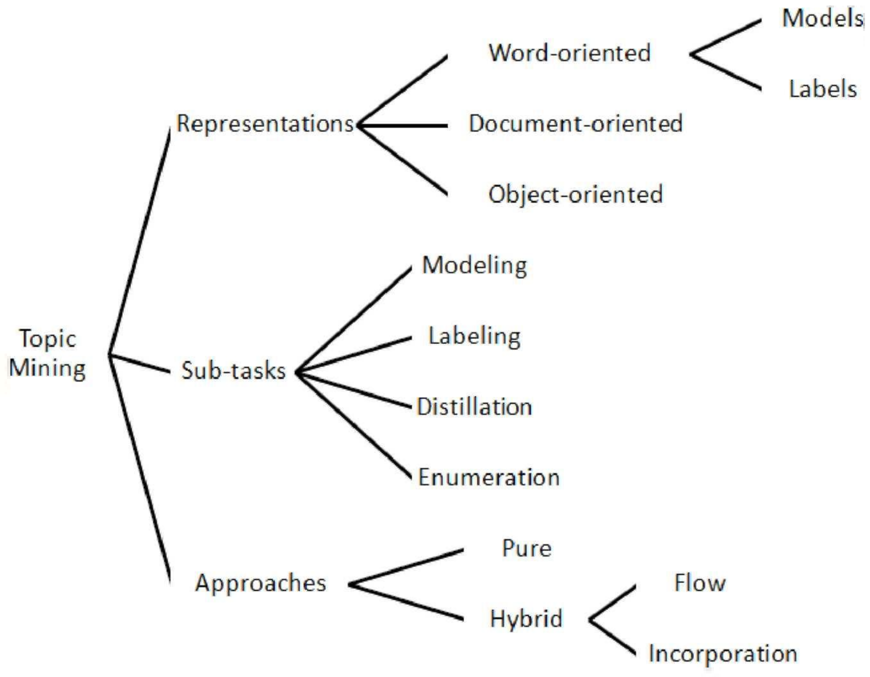


Figure 2.12: Topic mining taxonomy

we will explain later, topic mining concerns to us the *what*, while meaningful group detection, the *how*. Another important remark consists of *relevant work overlapping*; for instance, there are several approaches that use meaningful group detection in the Web to mine for topics and are worthy enough to be mentioned again. To avoid redundancy, we will try to describe a different aspect of such jobs, logically more focused on topic mining.

To start with, it is relevant to introduce different classes of topic mining according to the used information source. This division is quite simple: either text, structure, or a mixture of these and other features can be used for the endeavor. Representation of topics, as it can be inferred, is partly derived from this choice; for instance, it is more usual to find that topics are treated as word collections if they use text as their information source. Moreover, source mixing is generally utilized under the notion of "getting the best out of several worlds" or "combining evidence" in order to better exploit the capacities of a certain technique. For example, text and structure can be integrated to achieve a more comprehensive similarity metric between documents.

Some concrete information sources that have been adopted for topic mining include the following:

- Text
  - Bag of words
  - Term distributions
- Structure
  - Directed or undirected edges

  - Weighted or unweighted edges
  - Co-citation

- Combination
  - Terms, co-citation information, and direct links all in one feature

A more important classification is given by dividing methods according to their representation of topics (which is partly defined as well by the kind of information source that is being used). In that sense, let us list the types we have found in literature:

Word-oriented.- *Descriptive* view of topics that basically consists of a *word collection*. In that aspect, the main entities for representing the topic are *individual terms*. Then, the topic could either be seen as a *model* (probabilistic distribution embodiment) or as a label (ad-hoc embodiment). Labels, on their own, constitute a loose concept, and thus can be constituted as a title phrase, a query, a simple keyword set (concise or lengthy), or—unfortunately for our classification—something in between. Models, on the other hand, are properly described by Griffiths et al. in [53].

Document-oriented.- *Enumerative* view of topics that conceives them as *document lists*. Unlike the previous representation, the basis for the topic embodiment is given by this other entity (which may in turn be composed of words or other kind of information). In that sense, it is common to find lists with document identifiers: titles, URL's, or the most popular words, for example. This representation is specially suited for approaches that do not employ text (e.g. link-based methods).

Object-oriented.- *Comprehensive* view of topics that combines the aforementioned representations. This causes the topic to be embodied as a *compound object*: a probabilistic model and a document enumeration, a document enumeration and a label, a model and a label.

A third, simple classification criterion concerns the type of *learning* that is used (supervised vs. unsupervised). Supervised approaches, as we have seen, count with a set of training examples where the correct output is already known. Unsupervised approaches, on the contrary, do not count with this knowledge *a priori*. While there are methods that opt for either one or the other, we may also find a few that employ both learning types for topic mining, specially if they look for compound objects; for example, unsupervised topic discovery (UTD) uses this kind of learning to create topic models and supervised learning to label the resulting models.

Another key classification criterion is the kind of topic mining approach. We can distinguish two main types: pure and hybrid. The former only use one type of basic task and utilize one type of representation. The latter, on the other hand, principally mix basic tasks, information sources, and/or representations. According to literature, we may consider the following as basic tasks:

Labeling. Consists of *naming* the topic. Given a cluster or category, the topic labeling task assigns a coherent, descriptive (human-readable, for the usual) "title"; commonly, this is done by mapping the conformed topic to a concept given by a certain ontology. In a strict sense, this task considers topics as labels; nevertheless, if we take into account that a cluster is also being considered for the labeling to take place, then this task actually views topics as compound objects. Other more technical names for this task concern *cluster annotation* and *cluster labeling*.

Distillation. Consists of detecting the *authoritative* documents of the topic. Given a "broad" query, the most popular documents, along with those resources that link to these documents, are extracted. Concerning the topic representation, the same issue arises as with labeling. In the narrowest sense, distillation conceives a topic as a list of documents; however, realizing that a query is needed for the distillation process to be executed, we might as well point out that such task works with topics as compound objects. Distillation is equivalent to the previously discussed *resource discovery* task, and is also known as *broad topic search* (or just "topic search").

Modeling. Refers to *characterizing* the topic. Unlike the last couple of basic tasks, modeling has different flavors; the most common one involves probabilistic modeling, and it consists of conforming a theme by finding its term probability distribution (generative model). Needless to say, this task conceives topics as models. Some authors may refer to modeling as *topic analysis*.

Enumeration. Refers to *listing* the elements of the topic. The enumeration task is not less controversial than modeling; as defined by Kumar [77], it strictly consists of community discovery in the Web, where a community is defined as a *bipartite core* and the discovery process is given by *trawling*. Unluckily, this task definition seems quite reduced, as there exist other community definitions and other relevant community discovery techniques; what is more, we might even argue that enumeration does not have to be limited only to this kind of approach, but also to those that are able to provide a document list (data clustering, for example). Consequently, we can talk about having two different definitions for enumeration: a narrow one and a wide one. The former would be the definition provided by Kumar; the latter, the one we just gave. By taking this into account, topic enumeration can also be technically conceptualized as *dense sub-graph extraction* (graph mining task), *blockmodeling* (SNA task), and *group detection* (link mining task).

Two additional tasks that are worth to be mentioned, but lie out of our analysis are topic *segmentation* and Topic Detection and Tracking (TDT). The first one is dedicated to fragment texts (documents) in such a way that every "chunk" of information stands for a different topic. In other words, the goal is to divide a document according to the different themes it exhibits; to accomplish this, topic shifts in the text have to be detected. Segmentation is usually applied for fragmenting texts that result from speech recognition. In fact, sometimes it is applied in the context of a broader task. Related to this aspect is the second special task, TDT. This process can be seen, actually, as a *sui generis super-task*. On one hand, it is totally committed to a unique type of domain: event-based organization of broadcast news. On the other side, it comprises a series of activities, which include breaking down the text into individual news stories, monitoring the stories for events that have not been seen before, and gathering the stories into groups that each discuss a single news topic [5]. This last issue is solved by the *cluster detection* TDT sub-task, that, explicitly, aims to place news following the same line into disjoint clusters, called “bins”. In that sense, cluster detection resembles topic enumeration and can additionally be accounted as the most related to topic mining.

Once having the notions of these tasks, it is possible to describe *pure approaches*. With respect to topic labeling, a representative method for annotating groups in a hierarchical fashion according to ontologies and term frequencies is suggested by Stein and Eissen [135]. A significant contribution of this work, moreover, consists of a conceptual framework from which we can highlight the establishment of label properties, as to know: uniqueness, expressiveness, summarization, discrimination, contiguity, hierarchization, consistency, and irredundancy. Similarly, Schönhofen uses the Wikipedia category network as a source for labels; however, the aim is not precisely to name a group of documents, but instead to classify them by selecting the most dominant topic (category) they fall into; to test the approach, (annotated) documents were clustered by solely using their corresponding category and results were compared to the actual classification (external validation).

With regard to modeling, a characteristic approach that consists of producing generative models for scientific topics is supplied by Griffiths and Steyvers; such method views topics as probability distributions over words, and considers each document as a mixture (also probabilistically speaking) of these themes [52]. This work only employed article abstracts and was validated by means of comparing document topics with the classification provided by their authors. On the other hand, Wartena and Brussee are more fond to following the topic analysis philosophy of Li and Yamanishi [82]; consequently, the method consists of extracting a list of the most informative keywords (by means of the Kullback-Liebler divergence, mostly) and then clustering these keywords with induced bisecting k-means and a novel similarity measure based on the Jensen-Shannon divergence [144]. The mentioned approach was tested with the Dutch version of Wikipedia and concretely takes topics as cluster centroids given by the average distribution of co-occurrence distributions.

Pure enumeration approaches can be initially exemplified with Flake’s maximum-flow Web communities (previously discussed), which are topically related [41]; remarkably, this method only uses structure to gather the members of such topics, and every topic is shown as a collection of URL’s. A different approach is introduced by Ertöz et al., who attempt to find topics with shared nearest neighbors [38]; this work, although entirely enumerative, combines several features that it might be relevant to discuss. First, the overall method is based on the Jarvis-Patrick clustering algorithm, which calculates pairwise similarity between documents to construct a neighbor graph and applies a similarity threshold to break weak links, thus yielding a partition of the graph. However, the topic enumeration approach instead of partitioning uses different kinds thresholds to either discard a document from the clustering, make it a representative of its neighborhood, or consider it as “mergeable” with another document. The former results into a hierarchical structure that is shown to obtain purer clusters than k-means (taken, in turn, to be more effective than conventional hierarchical clustering), and, interestingly, the method is conceptually regarded as equivalent to a generative model. A secondary line of this work consisted of clustering individual words with the intent of finding concepts related to topics.

Distillation is best appreciated on the works by Kleinberg [73] and Chakrabarti et al. [25, 26] (Kleinberg and Chakrabarti belong to the same group, actually). Regarding the first, which has already been discussed in the meaningful group section, let us recall that the most representative pages of a query are obtained via spectral calculations; topics are depicted as URL lists. With respect to the second, the approach can be seen as an upgrade of the HITS algorithm (now referred to as the *Clever system*), since it addresses some of its issues, such as straying and multi-topic pages. The key for treating those issues is based on the use of anchor text[3] and link-weight assignment. Both approaches were validated with user studies; in some cases, these were asked to rate the outputs against competing results. It is important to remark that, even when these approaches should strictly be treated as hybrid because they use different information sources, we have decided to still classify them as pure; this decision was based on the fact that the method *per sé* is mostly link-based. However, such decision is arguable, as other authors do not hesitate on designing HITS and Clever as mixed approaches.

Regarding hybrid approaches, we can still subdivide works into several categories by assigning *paradigms*. On a first instance, we can talk about paradigms specific to *operation*, which usually create a flow composed of different tasks. For example, the continuation of Flake's method consists of an enumeration-modeling flow, as webpage communities are first listed and then characterized for validation [42]. Also, Modha and Spangler execute an enumeration-labeling flow by first clustering hypertexts and then annotating them [98] (this work was already reviewed). Furthermore, He et al. carry out an enumeration-distillation flow by first clustering webpages according to a mixed similarity metric and then selecting the most representative members of each topic by looking for hubs and authorities [57] (although in this case the result set given by the broad query is substituted by a webpage cluster). Related to the former is the work by Gibson et al. which first does distillation to find community cores and then expands results by adding non-principal site conglomerations (this can be accounted as a relaxed form of enumeration). An additional instance of a hybrid flow approach is provided by Schwartz et al. [128] and Sista et al. [131] (these being again the same group), who tackle the problem of *Unsupervised Topic Discovery* (UTD); the method (modeling-labeling) basically consists of characterizing topics via probabilistic tools, and then extracting human-readable names from an annotated corpus using classification (therefore, as had noted earlier, they also mix learning types). In that sense, the "unsupervised" term is awarded to the task because no training corpus is used to generate the topic models. On the other hand, Liu et al. seem to go further by proposing a framework that unifies community discovery with topic modeling [84]; such framework attempts to find a "joint" generative model for the text and hyperlinks of a topic. A method that uses the same philosophy is the one by McCallum et al. which portrays a *Role-Author-Recipient-Model* [90]. Yet another example is the Topic Modeling with Network Structure (TMN) task, proposed by Mei et al. [91]. Even when the three previously mentioned approaches can be classified as hybrid for their *modus operandi*, they could also fit into the next paradigm.

Other paradigm for hybrid works is due to assuming that *information incorporation* will yield better topics. An example of such paradigm is given by the work of Jo et al. which integrates co-citation information to obtain more accurate topic models [68]. This is accomplished creating a term citation graph that consists of an undirected network where only links of documents containing a certain term are taken into account; furthermore, the posterior probability of this term (word or phrase) being relevant to a topic given that its citation graph

[3] The text that appears on a link. For example, if we have `<a href=http://apple.com/store>Apple Store</a>`, the anchor text is "Apple Store".

is well connected is calculated (evidence combination). The aforementioned approach of He et al. follows also the incorporation paradigm by mixing textual similarity with direct link presence and co-citation indices.

Another interesting aspect with regard to topic mining approaches concerns the distinction between *important* and unimportant topics; concerning this, several authors have portrayed their special interest for discovering not all, but the most relevant topics. For example, Griffiths and Steyvers talk about *hot versus cold* topics, Ertö and et al. about *dominant vs. non-dominant* themes, and Wartena and Brussee imply this distinction as *prominent versus ordinary* topics.

A summary with the most relevant methods and their features according to the established taxonomy is illustrated in Table 2.7.

Table 2.7: Topic mining methods

Author	Contribution / Method	Representation	Task	Source
[38] Ertöz et al. (2003)	Shared searest neighbors	Document-oriented	E	Text
[41] Flake et al. (2000)	Maximum-flow communities	Document-oriented	E	Structure
[42] Flake et al. (2002)	Maximum-flow communities	Object-oriented	E, M	Structure
[49] Gibson et al. (1998)	Communities with HITS	Object-oriented	D, E	Structure
[52] Griffiths & Steyvers (2004)	Modeling of scientific topics	Word-oriented (model)	M	Text
[57] He et al. (2001)	Hybrid-source topic mining	Document-oriented	E, D	Mixed
[68] Jo et al. (2007)	Models with co-citation	Object-oriented (hybrid model)	M	Mixed
[73] Kleinberg et al. (1999)	HITS and Clever	Document-oriented	D	Structure
[84] Liu et al. (2009)	Topic-Link LDA	Object-oriented (hybrid model)	M	Mixed
[91] Mei et al. (2008)	TMN	Object-oriented (hybrid model)	M	Mixed
[98] Modha & Spangler (2003)	Hypertext clustering for Web search	Compund objects	E, L	Mixed
[127] Schönhofen (2006)	Text classification with Wikipedia categories	Word-oriented (label)	L	Text
[128] Schwartz et al. (2001)	UTD	Compound objects	M, L	Text
[135] Stein & Eissen (2004)	Topic labeling framework	Word-oriented (label)	L	Text
[144] Wartena & Brussee (2008)	Keyword clustering with co-occurrences	Word-oriented (model)	M	Text
D=Distillation, E=Enumeration, L=Labeling, M=Modeling				

2.2.1 Issues with the topic mining definition

Gathering and classifying topic mining works arises several interpretation issues and misleadings that it might as well be relevant to review. First, there does not exist a common, accepted, formal definition for topic mining; regarding this, it is usual to find that authors define their own conception of topic mining before describing their work. Furthermore, this discipline is labeled with a considerable number of different (although synonymous) names, as to know: identification, detection, discovery, extraction, and finding. In fact, an actual issue is that when using "topic detection", this broad task may be confused with TDT, a discipline that we have seen is very specific and focused towards a target, counts with particular components, and handles strictly the temporal dimension. Also, it is not uncommon to see that "identification" and "discovery" are more frequently used when dealing with supervised methods, and "extraction" has been related on some works as specifically related to modeling, while in others it implies enumeration. These previous issues may cause several other confusions.

An even more controversial matter is given by sub-tasks. As we saw on the last section, these disciplines by themselves can also act as umbrellas for covering a series of methods that seem to be similar, but at the same time have peculiar characteristics. Therefore, it remains unclear what a specific sub-task should carry out (e.g. should modeling include segmentation?) and if the equivalence between group detection techniques and topic sub-tasks is strict or loose (e.g. is community discovery exactly the same as topic enumeration?). Then, the expectations regarding tasks are *wide open*, to say the least.

To ground this brief discussion, let us state the main point for addressing these issues:

1. *Accept the openness of the discipline*
   (a) Works cannot be classified as correct or incorrect, as they all approach a field that has not been strictly defined.
   (b) The limits for topic mining are blurry; therefore, it is also quite difficult to state with precision what is topic mining and what isn't.
   (c) Sub-tasks of the field are vague as well.

2.3 Wikipedia revisited

Wikipedia is probably—and so far—the most representative instance of a *WikiWikiWeb* (usually named only “wiki” for short) [27]. The essence of this collaborative Web technology relies on enabling users to share knowledge in a fairly simple way by letting them to modify page contents with nothing more than a browser and some text markup[4]. Wikis additionally provide discussion and history pages.

Because of the links existing among articles and to other Web pages, Wikipedia can be studied as a complex network. Analyses of this type include [143], which is more focused on growth statistics and [152], that reports measurements such as degree distribution, topology, clustering coefficient, and path lengths.

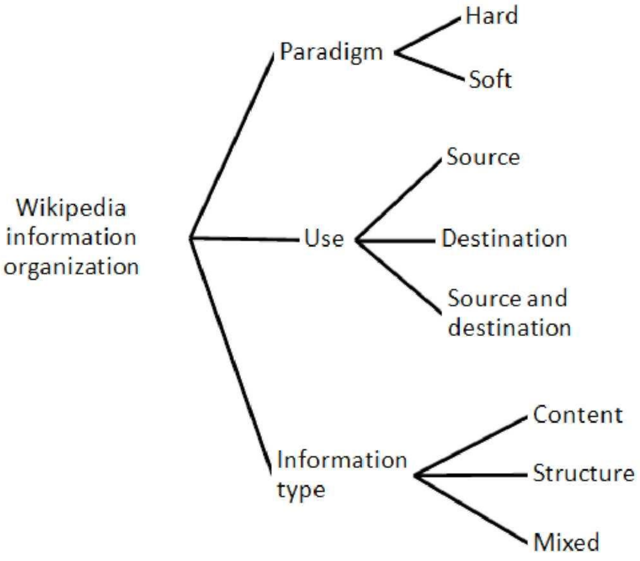


Figure 2.13: Wikipedia information organization taxonomy

Efforts concerning Wikipedia use for extracting semantics can be seen as having a threefold division (see Figure 2.13). First, we have hard vs. soft approaches, being the use of ontologies the main difference between these kinds of methods (we may consider this categorization as *Semantic Web paradigm*). Second, we can distinguish works that consider Wikipedia only as a rich information source from those that use Wikipedia both as a source for semantics and a destination over which the extracted knowledge can be applied (let us define this as the *type of use* given to the corpus). Finally, works can be differentiated according to the *type of information* used (content, structure, or both). Since we are more interested on works where Wikipedia is also a destination, we will enumerate these works with slightly more detail.

[4] Check Wikipedia’s cheatsheet (`http://en.wikipedia.org/wiki/Wikipedia:Cheatsheet`) for a clear example of such markup. For references that explain wiki server functionality and installation, see the book by Ebersbach et al. [37].

Concerning soft methods that consider Wikipedia as source and destination, Adafre and de Rijke propose an approach for discovering missing hyperlinks inside Wikipedia's articles [1], and this was accomplished by performing two tasks: first, clustering pages with related content (including titles and links) and co-citations, and second, obtaining from these clusters the most related articles, for finally determining which links should be inserted in the queried page. A similar work is [14] research was focused on clustering Wikipedia music-related articles by means of a Self-Organized Map (SOM). Fragments of text and the titles of each article were used, as well as text belonging to hyperlinks included in these articles. The final result consisted of a topological map where similar documents were grouped together, according to discovered categories. Furthermore, [147] uses supervised machine learning methods to automatically generate links and infoboxes.

Regarding text-based approaches, there are plenty of works that cluster related documents, and it is difficult to supply a comprehensive list. However, two methods that focus on the Wikipedia corpus are [127] and [144]. Concerning the former, it uses Wikipedia's category network and document titles for clustering and classification in several corpora. On the other hand, Wartena and Brussee experiment with various similarity metrics utilizing bisecting k-means as the clustering approach and encourage their proposed metric, that is based on term co-occurrence, for obtaining higher quality results.

Two works that use NLP techniques for automatically linking Web pages to Wikipedia articles (this action has been baptized as "Wikifying" pages) are the ones proposed by Mihalcea [96] and Milne [97]; the core of such works consists of disambiguating the terms where links are to be inserted. An important contribution of Milne's research is a measure that outputs the semantic distance between two Wikipedia documents.

A first (hard) approach for constituting a semantic Wikipedia is given by [141]; this work attempts to offer a framework for contributors to include "typed" links and attributes in RDF and OWL format. Furthremore, another very relevant effort in the hard area concerns DBPedia (http://dbpedia.org), which automatically extracts semantics from infoboxes (called "templates") and codes this information into RDF triples [8]; the ontology can actually be queried with SPARQL[5] and serves for satisfying general knowledge inquiries. Therefore, it basically uses Wikipedia as an information source. Both approaches can be considered as using mixed information, since typed link information is given by text and infoboxes are a combination also of content and structure. Another work that falls into this category is [100], since it intends to exploit the Wikipedia's text and links for creating a universal Web ontology; similarly, the Yago ontology [136] is built on top of Wordnet[6] relations and Wikipedia extracted knowledge.

A summary with the most relevant methods and their features according to the established taxonomy is illustrated in Table 2.8.

[5] A language for querying RDF. http://www.w3.org/TR/rdf-sparql-query/

[6] An automatic English lexicon. Can be accessed in http://wordnet.princeton.edu . For a comprehensive description of this language tool, see the book by Fellbaum [40].

Table 2.8: Wikipedia automatic organization methods

Author	**Contribution / Method**	**Paradigm**	**Use**	**Information**
[1] Adafre & Rijke (2005)	Wikipedia missing link detection	Soft	Both	Mixed
[144] Wartena & Brussee (2008)	Keyword clustering with co-occurrences	Soft	Source	Content
[136] Suchanek et al. (2008)	Yago (ontology)	Hard	Source	Content
[141] Völkel et al. (2006)	Wikipedia ontology framework	Hard	Both	Mixed
[8] Auer & Lehmann (2007)	DBPedia	Hard	Source	Mixed
[96] Mihalcea & Csomai (2007)	Wikification	Source	Soft	Content
[97] Milne & Witten (2008)	Wikification with WSD	Soft	Destination	Content
[147] Wu & Weld (2007)	Supervised automatic linking and infobox creation	Soft	Both	Content
[127] Schönhofen (2006)	Text mapping to Wikipedia categories	Soft	Source	Content
[14] Bloehdorn & Blohm (2006)	Article clustering with SOM's	Soft	Both	Content
[100] Nakayama et al. (2008)	Universal Web ontology	Hard	Source	Mixed

2.4 Chapter summary

Relevant background and state of the art methods revolve around three central areas: Web structure mining, topic mining, and Wikipedia mining. Web structure mining, on its own, is founded on the analysis of complex networks (e.g., social and citation networks) and the study of the Web as a graph, and comprises techniques that include group discovery methods. So far, we can distinguish three types of conglomerations: distinctive feature groups, common trait groups, and cohesive groups. In general, these groups are detected with three broad tasks, given by resource discovery, data clustering, and network clustering, respectively. Moreover, specific methods can belong to one of four classes: cut-based, pattern-based, spectral, and structure search-based. Such methods rely on affinity metrics and structures, like density, co-citation, and adjacency matrices. Also, they must overcome complexity either by employing problem decomposition, efficiency, approximations, and/or pruning. Finally, they can be evaluated internally, externally, and/or in a relative form.

With topic mining, we have different conceptions for a topic, which can be word-oriented, document-oriented, or hybrid. Furthermore, four mining sub-tasks can be found in literature: modeling, enumeration, labeling, and distillation. Specific approaches are pure by using one kind of representation and one kind of task or hybrid by combining representations and tasks.

Wikipedia regards a WikiWikiWeb collection. Its mining for semantic information extraction involves different Semantic Web paradigms (hard if ontologies are used, soft if the approach works with the information as is), uses for the repository (source, destination, or both), and information types (content, structure, or mixed).

By having more clear the present background and existing approaches for tackling the topic extraction problem, we are able to introduce our own method.

Chapter 3

Conceptual Framework: Model and Definition

The goal of this chapter is to *present our conceptual framework*; this framework can be seen as divided into two parts. The first part, on its own, involves the *conceptual model for the topic extraction process*. This model is introduced as a layered architecture, in relation to which the four topic sub-tasks can be placed. On the other hand, the second part of the chapter properly describes the first extraction sub-task: *topic definition*. Therefore, a graph-theoretic formal framework is presented in order to define the elements (universe) and operations (guidelines) that are being considered for detecting topics in a Web environment; such formal definition is to be exemplified as well for a better understanding.

3.1 Topic extraction as a layered model

From previous chapters, we have learned that topic extraction is concerned, in general terms, with the discovery of themes residing inside a document collection; it has also been stated that we have grounded this process as the one in which *topically-related document groups are discovered*. A form of elucidating such process for an easier tackling consists of breaking it down into *levels*; in that sense, a basic level could correspond to group discovery (*clustering*), and a more abstract one could correspond to handling these groups as topics (*semantics management*). These levels can be seen as the backbone of our *topic extraction conceptual model* (which we will refer to as "TCM" from now on).

Simplification is not the only reason (although the most obvious one) for splitting up the topic extraction process. This concrete separation into distinct abstract views also serves for delineating the relationship between clustering and topic extraction, which is conceptually "coarse", as it has not been explicitly stated what part does clustering play when attempting to find document groups with the same thematic (is it equivalent? is it a part of? are they complementary?. . .).

As we have seen from the state of the art, acknowledging individual document groups as sharing the same thematic is usually taken as granted. Therefore, at an initial glance, it seems that we could ultimately: 1) treat topics as being equal to clusters and 2) carry out their extraction merely as a clustering procedure. Nevertheless, a "rough" cluster can represent anything: a group of persons, particles, sensors, etc. So, if we wish to go a step further, this leads us to additionally consider *meaning* into topic extraction—therefore visualizing topics as *clusters whose semantics imply a common theme among the elements of the same group*.

By including semantics into the equation, clustering becomes *the basis* for topic extraction. It must remain clear that by defining the clustering-extraction relationship this way, we do not intend to state that a cluster *per sé* lacks semantics, but rather that these are latent and would have to be uncovered or made more explicit in order to say that our group is actually a topic.

Now, with regard to "uncovering" the semantics of a cluster, we imply the following:

- Validating that it is indeed a topic
- Naming the theme that gathers the documents together
- Presenting it as a usable piece of information for theme-acquainted applications (search utility, visual map, navigation tree, etc.)

As we can see, the first point of semantics management (ratification) can be directly related with our sub-task of topic validation, whereas the other two points are more concerned with topic description. Also, as the reader may infer, topic construction is intrinsically related to clustering. However, with the intent of providing a clearer picture of the discussed extraction levels and depicting as well how they are intertwined with our sub-tasks, perhaps it is easier to introduce an architecture for visualizing the whole topic extraction process.

An architectural style that seems appropriate to display our TCM is given by *layered systems*. This kind of software architecture groups system components according to their generality: less abstract components are placed on lower layers, while specific ones are built "on top" of these [63]. Each layer interacts directly just with the layer immediately beneath it and provides facilities for use to the layer above it. Probably the best representative of the layered style is the ISO-OSI model for networks.

By analyzing this architectural style, it becomes clear that the use of *levels that range from the concrete to the abstract* helps to decompose a complex problem into simpler parts; furthermore, because lower layers are less abstract, usually *information starts making sense at upper levels*. Another interesting trait is that information is *presented in an understandable form* for a layer by the preceding one. Even when the former may be considered as a tradeoff since it implies inter-layer communication, this actually enables a modular structure, where *the particular implementation of each module is independent from the others*, as long as it complies with its function. Finally, a relevant aspect to highlight is that functions at lower levels *represent the nucleus* of the process that solves the problem. Let us discuss these interrelated features with more extent in order to be able to fully describe the TCM.

Abstraction levels. In a layered model, data remain the same from level to level; variations, instead, occur with the interpretation of such data. This assortment of interpretations is given by increasing levels of abstraction, which is higher as we travel to upper layers (for instance, the OSI model has seven). The former reveals two aspects: data acquire meaning in a bottom-up fashion and information is facilitated for the immediate upper layer. Let us go deeper into this pair of aspects.

Bottom-up meaning acquisition. At the lowest level, *raw* data pieces are managed (basically no abstraction done here), but at each superior layer these pieces have a higher resemblance with information—that is, data shaped into a meaningful and useful form [80].

Facility provision. Layers get what they need from the ones below, and the ones below present information in such a way that it can be "usable" for the upper layers that will manage it (typically the one immediately above). This may imply to tag or summarize information.

Transparency. Implementations can be interchanged without redesigning the whole system again. By handling encapsulation (concealment of implementation details), a model becomes more general and adequate for component reuse.

Core process definition. Layers cannot only be represented as a pile structure (where each layer usually lies on top of another one), but also as a collection of concentric circles, where the innermost circle represents the lowest layer [10]. This enables us to visualize the concept of core functionality more clearly; in that sense, a layered architecture is best suited for describing models where we have a base process.

3.1.1 Topic Extraction Conceptual Model (TCM)

Our conceptual view of topic extraction (presented in Figure 3.1) can be portrayed as a four-layered architecture, where the levels—from lowest to highest—comprise the following:

Data representation layer.- Provides a coherent data depiction that can be used for clustering. Physically, this implies obtaining relevant information from raw data; logically, it demands to formalize information pieces. Therefore, data representation consists of two additional sub-layers: logical and physical.

Clustering layer.- Embodies topical clusters from the abstracted pieces of information.

Semantics layer.- Uncovers the meaning of the embodied clusters.

Application layer.- Makes use of the extracted topics for various purposes.

Within the lowest level, the data representation layer, information—raw data, more precisely—concerns a *bunch of physical documents* (mixture of text, hyperlinks, media, markup, and other metadata). Specific functions of this layer include extracting information of interest (therefore dismissing unimportant data) and separating information by types (e.g. conceptually, by defining different elements, and, physically, by keeping data stored in distinct structures); note that these operations can be implemented in a variety of ways, for example, by employing distinct extraction scripts and data storage media. The ultimate function consists of creating objects for, subsequently, grouping.

Considering that the preceding layer is more of an aid, the actual core of the TCM lies at the next abstraction level, the clustering layer. Here, information is handled as a *universe of individual elements* (logical documents), over which a previously designed grouping algorithm is carried out (thereby, let us note that transparency is held, as a variety of clustering

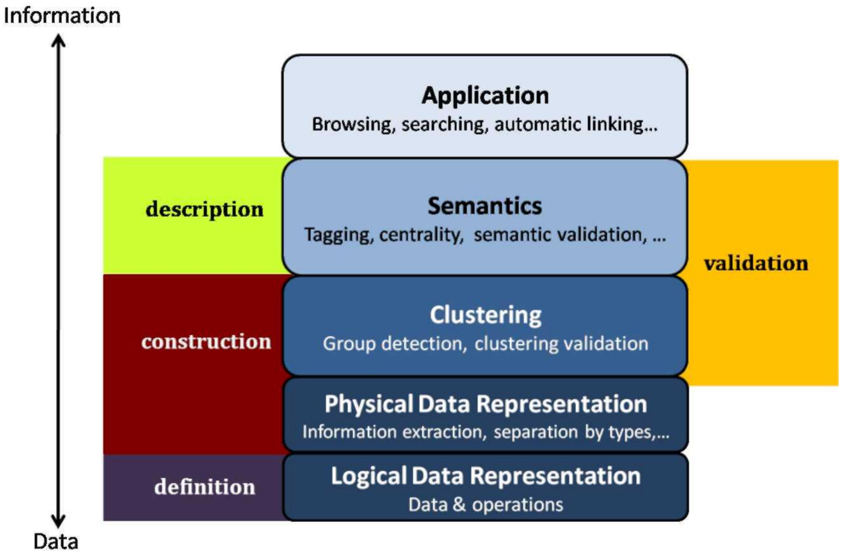

Information
Application
Browsing, searching, automatic linking...
description
Semantics
Tagging, centrality, semantic validation, ...
validation
construction
Clustering
Group detection, clustering validation
Physical Data Representation
Information extraction, separation by types,...
definition
Logical Data Representation
Data & operations
Data

Figure 3.1: TCM

techniques may be applied). Clusters from which collective traits can be obtained are then produced.

The semantics level, where meaning acquisition becomes more pronounced, works with *document conglomerations*. General assignments for this layer have been already discussed on the first section, but specific chores concern word and document ranking; as with the two preceding levels, different alternatives exist for doing these jobs. In last instance, thematically-related groups of documents with explicit properties, such as a name ("tag") and a compendium of the most central members ("summary") for use in several contexts are assembled.

At the final level, the application layer, *topical clusters with properties* (topics) are handled. How topics are specifically used depends on the context; however, let us provide a pair of examples. A possible application context is given by search; in that sense, the topic tag terms may be used as a quick way for users to find topics. Another application instance (probably more Wikipedia-related) could concern presenting a knowledge area in a "cloud" form, where representative or central articles are visualized with a larger font (see Figure 3.2). This last layer belongs to the TCM, but it is important to recall that it lies out of our scope; therefore, it is to remain as a "black box" (discussed but not developed).

Figure 3.3 states the aforementioned inputs and outputs of each layer. In addition, Table 3.1 presents the analogy between the OSI model and the TCM.

3.2 Topic extraction sub-tasks revisited

In a nutshell, let us recall what each task consists of, for later explaining at which levels it becomes executed.

Topic definition. Consists of formally representing topic extraction (in our case, via link-based clustering). On the TCM, this task provides the inter-layer *common language* ("rules of the game") for the three lower levels; however, it is properly executed at the

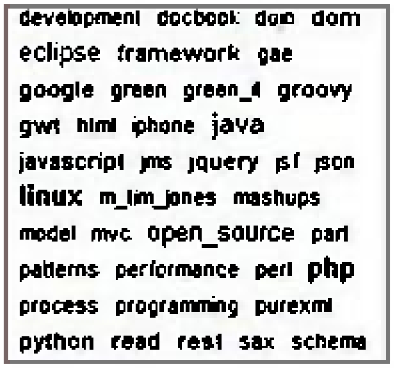

development docbook dojo dom
eclipse framework gae
google green green_it groovy
gwt html iphone java
javascript jms jquery jsf json
linux m_tim_jones mashups
model mvc open_source perl
patterns performance perl php
process programming purexml
python read rest sax schema

Figure 3.2: A topic "cloud".

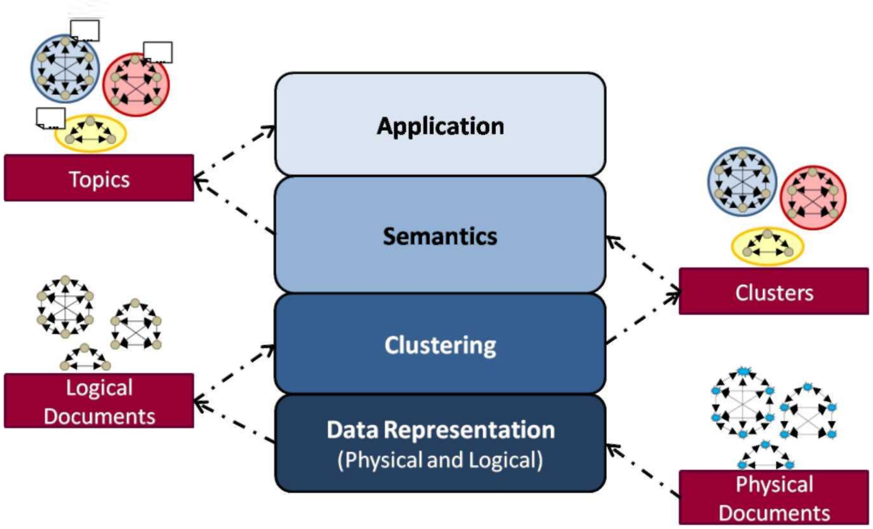


Figure 3.3: Layered model with inputs and outputs.

data representation layer. Actually, because it conceives the universe and guidelines for interpretation, it is, more exactly, carried out at the *logical data representation sublayer.*

Topic construction. Consists of enumerating the elements of the topical groups. It concretely starts at the physical data representation level, because a basic part of topic construction consists of cleansing and shaping our information so it is consistent with the formal definition. Nevertheless, as expected, the bulk of this sub-task is carried out at the clustering level; this alignment between topic construction and clustering, in fact, concentrates topic extraction around the construction endeavor. On one hand, this can derived from the fact of knowing that clustering concerns our base process; on the other side, definition, description, and validation design choices are driven by construction. For this reason, *our efforts shall be more focused on the development of this task.*

Topic validation. Consists of ratifying that our found clusters are indeed topics. Considered also as an important process, validation can be done at two levels: clustering and semantics. At the first layer, we might corroborate that our groups comply with the

Table 3.1: OSI analogy.

	OSI model	**TCM**
Abstraction levels	Seven abstraction layers	Four abstraction layers
Meaning acquisition	Bits, frames, packets,...	Physical documents, logical documents, clusters, topics
Facility provision	Frame markers	Tagging
Transparency	Different transport protocols allowed.	Different extraction and group detection techniques allowed.
Core	Lower layers constitute the core for communications.	The clustering layer represents the core for topic extraction.

cluster conventional definition (a group whose elements are more similar among themselves than with respect to the ones belonging to others) and are cohesive; at the second layer, we can validate that their bondage is topical as well.

Topic description. Consists of calculating topic properties; needless to say, it is carried out at the semantics level.

3.3 Topic definition

Topic definition, which takes place at the logical data representation layer, concerns the first sub-task to carry out. Let us, then, formalize the main components and operations of the extraction process and exemplify several of them as well.

If we would wish to decompose the extraction formal framework into coarse elements, we could talk about two general classes: basic elements (data) and extraction elements (operations). While the former are introduced to explain the context into which a topic is defined, the latter help to describe topic construction and description.

Basic elements.- Informational items.

- Corpus
- Document
- Topic

Extraction elements (functions).- Manage basic elements.

- Clustering function
- Property calculation methods

3.3.1 Basic elements

Basic elements comprise information views at different granularity levels.

Corpus

Seeing that the corpus is the most general element of the framework, we may count it as the universe of discourse, where all the available information is contained and over which methods are (finally) carried out. This universe can be divided into two essential components—namely content (text) and structure (hyperlinks). Following conventional representations, content may be thought of as a bag of words and structure can be represented by means of a directed graph. Therefore, a corpus is formalized as a duple

$$C = (G, W)$$

where:

- $G = (V, E)$ is a graph where the vertex set V represents documents and the edge set E represents hyperlinks among them,
- W is the set composed by the union of all (unique) words found in the collection's documents (a.k.a. vocabulary), and
- the corpus's size (determined by the number of documents) is additionally denoted by N. Note that $N = |V|$.

Keeping text and structure as separated entities results convenient, since we will mainly use the latter for topic extraction.

Document

A document, on the other hand, constitutes an atomic piece of information, and we can view it in terms of three important features: its title, text, and hyperlinks to other documents. Consequently, we will define it as a triple

$$d_j = (a_j, L_j, W_j)$$

where:

- a_j represents the document title (anchor text),
- L_j is the set of pages the document links to,
  - $L_j = \Gamma(d_j)$, where $\Gamma(v)$ represents the neighborhood of a vertex v
- And finally, $W_j \subset W$.

Regarding L_j, as we will see later, we can either take the neighborhood of a vertex v as the set of nodes *it points to*, the set of nodes it *is pointed by*, or both cases. We will denote this first case with Γ_o ("out-bound neighborhood"), and the second one with Γ_i ("in-bound neighborhood"), while considering that the complete neighborhood of a given node is the union of these. Formally:

$$\begin{aligned}
\Gamma_o(d_j) &= \{l | l \in V, (d_j, l) \in E\} \\
\Gamma_i(d_j) &= \{l | l \in V, (l, d_j) \in E\} \\
\Gamma(d_j) &= \Gamma_o(d_j) \cup \Gamma_i(d_j)
\end{aligned}$$

Topic

In an abstract sense, a topic concerns a subject matter or theme; in a more concrete way, we can see it as a document cluster with "topical" semantic properties. Consequently, it can be defined with respect to the set of documents it covers and the features held by this set. In that aspect, it is represented by a duple

$$\mathcal{T}_i = (\mathcal{C}_i, P_i)$$

where:

- $\mathcal{C}_i$ is the document set (cluster)
  - $\mathcal{C}_i \subset V$
  - $P_i = (t_i, R_i)$ is a set of topic properties, where:
    - t_i is the topic tag, denoted by a K set of keywords, $K \subset \left\{ w_i | w_i \in \bigcup_{0}^{|\mathcal{C}_i|} W_j \right\}$
    - R_i is a subset of the most representative documents according to a criterion, $R_i \subseteq \mathcal{C}_i$

Moreover, every topic $\mathcal{T}_i$ can be seen as belonging to a corpus set of topics, denoted by

$$\mathbb{T} = \left\{ \mathcal{T}_0, \ldots \mathcal{T}_{|\mathbb{T}|} \right\}$$

3.3.2 Extraction functions

Extraction functions produce clusters and their properties.

Clustering functions

The clustering function, broadly speaking, is a function that receives the corpus structure G as input and produces a set of document groups $\mathbb{C}$ based on this information:

$$\mathcal{X}(G) = \mathbb{C}$$

where $\mathbb{C} = \left\{ \mathcal{C}_0, \ldots \mathcal{C}_{|\mathbb{C}|} \right\}$.

In the case of Graph Local Clustering, we actually need to define two functions: a basic function that allows us to extract exactly *one* cluster and an extended one (which is the *clustering* function, properly) that uses the former to generate the grouping that covers (ideally) all the corpus.

With respect to the basic function, it processes a *partial view* of the collection by taking a document subset (a.k.a. *seed*) and building a cluster "around it".

$$F(S_i) = \mathcal{C}_i$$

Therefore, we can see the extended function as "iterating" over a *seed set* S to produce the document group set $\mathbb{C}$:

$$\begin{aligned}\mathbb{C} &= F_{S_i \in S}(S_i), \forall S_i \\ &= \mathcal{X}_{GLC}(S)\end{aligned}$$

Unlike global methods, GLC does not guarantee to assign every document to a cluster; for this reason, we have to additionally include coverage into our framework, which is the proportion of clustered documents:

$$c = \frac{|\mathbb{C}|}{N}$$

We will also refer to c as “document coverage” to avoid confusions later on.

Property functions

In our case, we count with two functions that yield topic properties: a tag calculation function (π_t), and a representative document calculation function (π_r). The first one receives as input the vocabulary from the topic (all the words found in the document cluster $\mathcal{C}_i$) and returns the topic tag or set of keywords; the second one receives as input the document set and returns a document subset composed of the most relevant documents. More formally:

$$\begin{aligned}\pi_t(\bigcup_{0}^{|\mathcal{C}_i|} W_j) &= t_i \\ \pi_r(\mathcal{C}_i) &= R_i\end{aligned}$$

3.3.3 Example

Let us consider the small graph illustrated in Figure 3.4, which actually serves to visually depict the essence of our corpus. For simplicity, let us assume that the text of the documents consists only of their title. Also, for a better understanding, we will label all nodes only with the first word of the article.

Document

For this document example, we can see that it is identified with its title (a_{nicole}), its neighbors (L_{nicole}), and a vocabulary (W_{nicole}) consisting the the two words that precisely conform the title.

$$\begin{aligned}\text{nicole} = (& \\ & \text{“Nicole Kidman”}, \\ & \{\text{australia, oscar, renee, russell}\}, \\ & \{\text{“nicole”, “kidman”}\} \\ )&\end{aligned}$$

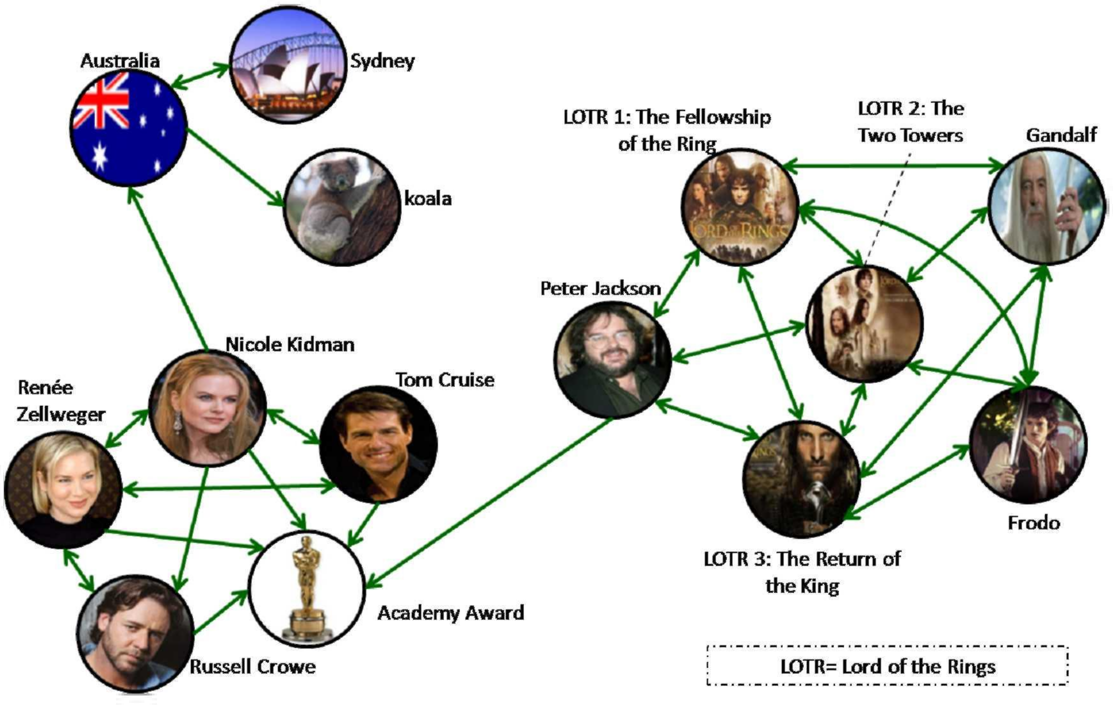


Figure 3.4: Example graph.

Topic

For this topic in particular, we can see that its document cluster ($\mathcal{C}_{\text{lotr}}$) consists of six elements; on the other hand, for the topic tag property (t_{lotr}) we are assuming that the most important keywords that make up this tag are the ones listed just after the document set. Finally, we are assuming as well that, for the representative document set (R_{lotr}), three elements were regarded as the most outstanding from the document set.

$$\begin{aligned}\text{lotr} \;=\; & (\\ & \quad \{\text{peter, lotr1, lotr2, lotr3, frodo, gandalf}\}\,, \\ & \quad (\{\text{“lord”, “rings”, “fellowship”, “towers”, “king”}\}\,, \\ & \quad \{\text{lotr1, lotr2, lotr3}\}) \\ &)\end{aligned}$$

Corpus

The corpus example portrays both the structure and content of our collection of illustrative purposes. On one hand, the first two elements concern the document graph ($G = (V, E)$), and the third one comprises the vocabulary of such documents (W).

$$
\begin{array}{l}
C = (\\
\quad (\{ \\
\qquad \begin{array}{lllll}
\text{australia,} & \text{frodo,} & \text{gandalf,} & \text{koala,} & \text{lotr1,} \\
\text{lotr2,} & \text{lotr3,} & \text{nicole,} & \text{oscar,} & \text{peter,} \\
\text{renee,} & \text{russell,} & \text{sydney,} & \text{tom} &
\end{array} \\
\quad \}, \\
\quad \{ \\
\qquad \begin{array}{llll}
\text{(australia, koala),} & \text{(australia, sydney),} & \text{(sydney, australia),} & \text{(nicole, australia),} \\
\text{(nicole, oscar),} & \text{(nicole, renee),} & \text{(nicole, russell),} & \text{(nicole, tom),} \\
\text{(renee, nicole),} & \text{(renee, oscar),} & \text{(renee, russell),} & \text{(renee, tom),} \\
\text{(russell, renee),} & \text{(russell, oscar),} & \text{(tom, nicole),} & \text{(tom, renee),} \\
\text{(tom, oscar),} & \text{(frodo, gandalf),} & \text{(frodo, lotr1),} & \text{(frodo, lotr2),} \\
\text{(frodo, lotr3),} & \text{(gandalf, frodo),} & \text{(gandalf, lotr1),} & \text{(gandalf, lotr2),} \\
\text{(gandalf, lotr3),} & \text{(lotr1, frodo),} & \text{(lotr1, gandalf),} & \text{(lotr1, lotr2),} \\
\text{(lotr1, lotr3),} & \text{(lotr1, peter),} & \text{(lotr2, frodo),} & \text{(lotr2, gandalf),} \\
\text{(lotr2, lotr1),} & \text{(lotr2, lotr3),} & \text{(lotr2, peter),} & \text{(lotr3, frodo),} \\
\text{(lotr3, gandalf),} & \text{(lotr3, lotr1),} & \text{(lotr3, lotr2),} & \text{(lotr3, peter),} \\
\text{(peter, lotr1),} & \text{(peter, lotr2),} & \text{(peter, lotr3),} & \text{(peter, oscar)}
\end{array} \\
\quad \}), \\
\quad \{ \\
\qquad \begin{array}{lllll}
\text{“academy”,} & \text{“australia”,} & \text{“award”,} & \text{“crowe”,} & \text{“cruise”,} \\
\text{“fellowship”,} & \text{“frodo”,} & \text{“gandalf”,} & \text{“jackson”,} & \text{“kidman”,} \\
\text{“king”,} & \text{“koala”,} & \text{“lord”,} & \text{“nicole”,} & \text{“peter”,} \\
\text{“renee”,} & \text{“return”,} & \text{“ring”,} & \text{“rings”,} & \text{“russell”,} \\
\text{“sydney”,} & \text{“tom”,} & \text{“two”,} & \text{“towers”,} & \text{“zellweger”}
\end{array} \\
\quad \} \\
)
\end{array}
$$

3.3.4 Topic definition: sub-task summary

To show compliance with the key questions provided at the problem statement and also as a means of recapitulation, a brief summary of each sub-task's main aspects is to be supplied after describing it. As for definition, Table 3.2 presents such recap.

3.4 Chapter summary

The topic extraction process is designed and developed as to comply with four main axes or *topic sub-tasks*: definition, construction, description, and validation. These tasks are aligned to an overall *topic extraction conceptual model* (TCM) that views topics as document clusters whose semantics reveal a thematic bondage; the model is inspired by layered architectures and consists of four abstraction levels: data representation (physical and logical), clustering, semantics, and applications. Definition and a part of construction are carried out at the representation level, while at the clustering level construction is finally accomplished; validation

Table 3.2: Main aspects of the definition sub-task.

Definition.- *Consists of establishing an explicit, formal definition for the topic extraction task in the context environment and the elements involved.*	
Key question	**Answer**
What is the intuitive definition of our task?	At the beginning of the chapter, topic extraction was intuitively defined as the *process in which topically-related document groups are discovered*, and a topic was presented as a *cluster whose semantics imply a common theme among its members.* A consistent paraphrase of the latter was given specifically at the Topic Definition section, by defining a topic as a *document cluster with "topical" semantic properties.*
Which elements make up the task?	A concise classification of the elements involved in hyperlink-based topic extraction was provided. Such classification consisted of two main types of elements: basic (data) and extraction (operations). Basic elements include the corpus, single documents, and topics. Extraction functions include clustering functions and property calculation methods.
In terms of the acknowledged elements, how do we formally define the task?	Formalizations for basic elements $(C, d_j, \mathcal{T}_i, \ldots)$ and extraction functions $(\mathcal{X}, \pi)$ were given.

is also accomplished at two levels, namely, clustering and semantics. Finally, description is done at the semantic level.

Regarding topic definition, an extraction *formal framework* that comprises the basic elements (document, corpus, topic) and operations (construction/basic function, clustering/extended function, property methods) of the process has been introduced. This framework is graph-theoretic, since it is suited for a hyperlinked environment such as the Web.

Within the current conceptual framework, it becomes possible now to detail topic construction and description.

Chapter 4

Approach: Topic Construction and Description

The aim of the current chapter is two-fold: explain how topics are constructed and portray how they are described. Regarding construction, an initial task corresponds to *data preparation* for clustering (physical data representation sub-layer of the TCM). On the other hand, before going deeper into the method, an overview—with a special emphasis on the class of problem to solve—is to be provided; actually, at this point we compare topic construction to the process of climbing all the peaks on a surface. Such process can be grounded as community search, which in turn may be considered as a form of graph clustering, and our general approach for carrying out this task concerns the use of GLC. Once having gone through this part, the basic topic construction algorithm is introduced, for afterwards explaining its fine tuning (backed up by exploratory experiments performed on generated Wikipedia sub-collections).

With respect to topic description, the goal is to state two topic properties, as to know: topic tags and outstanding members. For each property, we suggest specific methods in order to obtain it. Additionally, we show several examples of these descriptors.

4.1 Topic construction

Construction—*the enumeration*[1] *of topic elements*—is the most important sub-task of the extraction process; hence, it is the most critical one, and also the one in which the greatest part of our attention is to be centered. As we have seen before, this task is executed within the representation and clustering levels.

With regard to data representation, the aim is to *generate physical information pieces* that are suitable for clustering. While at a first glance data representation concerns a subject of little interest, as it is related with the physical management of the data, it is actually relevant to describe it for several reasons. First, the manner of *preparing data* is an integral part of the data mining process; in that sense, a depiction of physical data representation should be

[1] Note that throughout the rest of the chapter the term "construction" shall be used to mean *topic enumeration*. This obeys two primary reasons: being consistent with the name of our (four) topic sub-tasks and attempting to describe how our approach works (since clusters are built in a bottom-up fashion).

Chapter 4

Approach: Topic Construction and Description

The aim of the current chapter is two-fold: explain how topics are constructed and portray how they are described. Regarding construction, an initial task corresponds to *data preparation* for clustering (physical data representation sub-layer of the TCM). On the other hand, before going deeper into the method, an overview—with a special emphasis on the class of problem to solve—is to be provided; actually, at this point we compare topic construction to the process of climbing all the peaks on a surface. Such process can be grounded as community search, which in turn may be considered as a form of graph clustering, and our general approach for carrying out this task concerns the use of GLC. Once having gone through this part, the basic topic construction algorithm is introduced, for afterwards explaining its fine tuning (backed up by exploratory experiments performed on generated Wikipedia sub-collections).

With respect to topic description, the goal is to state two topic properties, as to know: topic tags and outstanding members. For each property, we suggest specific methods in order to obtain it. Additionally, we show several examples of these descriptors.

4.1 Topic construction

Construction—*the enumeration*[1] *of topic elements*—is the most important sub-task of the extraction process; hence, it is the most critical one, and also the one in which the greatest part of our attention is to be centered. As we have seen before, this task is executed within the representation and clustering levels.

With regard to data representation, the aim is to *generate physical information pieces* that are suitable for clustering. While at a first glance data representation concerns a subject of little interest, as it is related with the physical management of the data, it is actually relevant to describe it for several reasons. First, the manner of *preparing data* is an integral part of the data mining process; in that sense, a depiction of physical data representation should be

[1] Note that throughout the rest of the chapter the term "construction" shall be used to mean *topic enumeration*. This obeys two primary reasons: being consistent with the name of our (four) topic sub-tasks and attempting to describe how our approach works (since clusters are built in a bottom-up fashion).

included for the sake of completeness. Second, and most important, describing data representation in physical terms helps to establish a bridge between our conceptual framework and the clustering process; therefore, including this description eases comprehension and helps to fill construction voids. Third, explaining this aspect facilitates future extensions and/or related work. Consequently, a fragment of the current section explores aspects that have to do with extracting information of our interest: an overview of the process, available options, and the corresponding balance for using one option or the other.

The "juice" of the construction sub-task corresponds, logically, to *clustering*. As expected, the major part of the current section is devoted to describe this part of the extraction process; with regard to this, five main aspects—that range from broad to specific—are to be discussed. The first of these aspects corresponds to presenting an *abstract overview* of the construction approach, which is based on community detection. Interweaved with this overview is the second aspect, that aims to exhibit the *kind of problem* we are actually trying to solve via clustering. On the other side, the third aspect looks upon justifying our *general clustering approach*; finally, the fourth and fifth aspects have the purpose of *describing our two construction algorithms*. With the former aspect, the target two-fold: on one hand, to present the *concrete approach* that flows from our clustering approach, and on the other hand, to introduce the algorithms in *simple terms* for an initial understanding. With the latter aspect (fifth), the main idea is to discuss the*fine-tuning issues* that arise from applying the algorithms in our actual context.

4.1.1 Link information extraction (physical data representation)

The first step for topic construction consists of extracting link information from Wikipedia. According to our formal framework, we would be interested on gathering three informational entities: relations (links), anchor text (which is equivalent to collecting the articles' titles), and article content (only text). Since content does not confer the topic construction task (will be used for topic description), details about its extraction shall be described later—consequently depicting information extraction mostly in terms of links, but without forgetting that text availability is mandatory for information source selection.

A very important issue that it is necessary to clarify at this point is that we will only consider links to and from Wikipedia articles (that is, links internal to the collection). Therefore, references to external sites (e.g. the ones placed in the "External Links" section) will not be taken into account. Also, it is extremely relevant to point out that Wikipedia articles *do not make auto-references*; therefore, a given article of this collection never links itself.

Fortunately, for the most popular encyclopedia nowadays, there exist several sources from which our essential information can be extracted. These are briefly described next:

- **XML dump**.-Structured file that contains every article of the Wikipedia collection. Includes content (text, images), links, and metadata (revisions, timestamps).

- **WikiPrep**.-"Wikipedia Preprocessor" is a third-party tool that generates a set of files with Wikipedia information by using a PERL script. From this set, the two main files concern a "clean" version of the XML dump (e.g. markup is dropped from the articles' text and several metadata are removed) and a first-level category hierarchy.

- **Static HTML dumps**.-Set of HTML files (one file per article) arranged into a trie[2] structure.
- **Crawls**.- Set of HTML files; basically the same as above.
- **MediaWiki database**.-Database that contains all information from the XML dump. Content information is encapsulated into blobs[3] (one blob per article).

Before making a more thorough analysis of each alternative, it results convenient to state what we aim to produce with the source information, as well as some (subjective) properties that we consider as desirable for having an extraction easier to handle. In the first place, the expected final outcome—broadly speaking—is to have a database for accessing data (see Figure 4.1). Therefore, we need to be able to create data records with the least difficulty as possible. Having the former in mind, our desirable properties are the following:

∇ Easy link access

∇ Article title / anchor text availability

∇ Clean text

∇ Easy access to all articles

∇ Minimum amount of unnecessary material

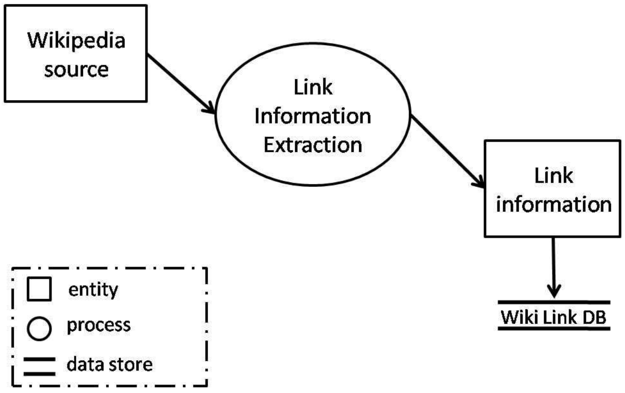


Figure 4.1: Data flow diagram: link information extraction.

Now, let us first start by analyzing the XML dump, which can be ultimately considered as the *source of sources* (this probably being its major advantage). A double-bladed feature of such source consists of its monolithic structure. On one hand, it concentrates all relevant data into the same place, thus avoiding the need of obtaining information from different sources and then gathering it altogether; but, at the same time, it makes it difficult to go through it (either for reading, scanning, or parsing). Now, two bigger tradeoffs than the previous concern

[2] Also known as a prefix tree, a trie consists of a dictionary-like tree data structure. Its name comes after the word "retrieval", and one of its most common uses concerns automatic spelling correction.

[3] blob=Binary Large Object

Table 4.1: XML dump page example.

```
<page>
    <title>Nicole Kidman</title>
    <id>21504</id>
    <revision>
      <id>237934169</id>
      <timestamp>2008-09-12T13:59:01Z</timestamp>
      <contributor>
        <ip>58.8.164.32</ip>
      </contributor>
      <comment>/* Filmography */</comment>
      <text xml:space="preserve">

'''Nicole Mary Kidman''', [[Order of Australia|AC]]
(born June 20, 1967), is an [[Academy Award]]-winning
actress. In 2006, she was the highest-paid actress
in the motion picture industry.&lt;ref&gt;
{{cite web | author=msnbc | title=
 Nicole Kidman highest paid female actor in film
 industry.| publisher=msnbc | date=November 30, 2006
 | url=http://www.msnbc.msn.com/id/15958023/}}
 &lt;/ref&gt;

After making various appearances in film
and [[television]], Kidman received her breakthrough
role in the 1989 thriller
''[[Dead Calm (film)|Dead Calm]]''...
```

Wiki markup (already discussed) and redirected pages. To better appreciate how a page looks like with the mentioned special markup, let us go to Table 4.1; as the reader can note, despite the fact that a scanning and/or parsing tool is required for processing this kind of file, the most serious issue is that links are denoted by text (see Table 4.6 for a summary of link formats), instead of an identifier—even when every page indeed has one. An obvious solution consists of mapping (at some cost) the anchor text with the page's respective identifier. In the case of redirections (see Table 4.2), matters gets worse, because a double mapping is required: one from the original to the redirected page and another from the anchor text to the identifier. Finally, to parse in an efficient manner and avoid starting out from scratch, Wikipedia creators recommend scripts for parsing (WikiPrep being one of them).

In that sense, the XML dump complies little with our desired properties: link access is hard, article access is not trivial (although all articles are present here), the text is far from being considered as clean, and there is a lot of irrelevant information.

With respect to HTML versions (see Table 4.3 for an illustration), even when they seem

Table 4.2: XML dump redirected page.

```
<page>
 <title>Applied Mathematics</title>
 <id>616</id>
 <revision>
  <id>161800205</id>
  <timestamp>2007-10-02T15:06:23Z</timestamp>
  <contributor>
   <username>LMF5000</username>
   <id>679103</id>
  </contributor>

  <comment>[[WP:AES|]]Redirected page
           to [[Applied mathematics]]
  </comment>

  <text xml:space="preserve">#redirect
      [[applied_mathematics]]
      {{R from other capitalisation}}
  </text>

  </revision>
</page>
```

Table 4.3: HTML page

```
<p>
  <b>Nicole Mary Kidman</b>,
  (born 20 June 1967) is an American-born
  Australian actress,  fashion model, singer, and
  <a href="/wiki/Humanitarian" title="Humanitarian">
  humanitarian </a>. Kidman has been a Goodwill
  Ambassador for
  <a href="/wiki/UNICEF" title="UNICEF">UNICEF</a>
  Australia since 1994...
```

an obvious and straightforward choice, they actually present more drawbacks than benefits. First, crawls are not encouraged in Wikipedia, and there are restrictive measures that prevent bots and spider crawlers from doing their job (not to mention if we try to download the whole collection). A second issue, probably not so meaningful as the previous one, is trie management. A third issue, and basically the most considerable one, is all the static useless content (headers, HTML markup, etc.); also, we inherit the problem of redirections and "mapping" (actually, more complicated, since we do not have identifiers here). So, according to our needs, we only see a single benefit with the HMTL bundles: on the contrary of the monolithic XML dump, they allow articles to be accessed individually, and this is more helpful for constructing the records of a database.

Regarding the desired properties, HTML only complies fully with the requirement of having titles available. Consequently, it can be stated as the least convenient source to use.

With respect to Mediawiki, this database can be generated from the XML dump by using PHP; despite conversion costs (which vary depending on the computer where the process is executed), an obvious advantage of this source is that it already stores article data in the desired medium, and queries can be readily executed over the information. Furthermore, individual tables for links and categories are available. Nonetheless, if we examine the DB[4] closely, it is possible to note that most of the tables (and even fields for content) concern administration information and metadata, such as revisions, users, statistics, etc. , which obscure to some point our vital information. Additionally, content is encapsulated into binary objects, so we would need to handle them.

With regard to the desirable properties, the Mediawiki option basically complies with every aspect, except for the amount of unnecessary material and article access (which is not entirely difficult, but is not straightforward either). Therefore, we can conclude that it can be a suitable source.

Finally, let us analyze Wikiprep. This script (which processes the XML dump to generate a refined XML) solves several of the inconveniences found on the other sources; first, it separates content from links by placing these within a separate XML tag (see Table 4.4).

[4] Actually, the DB schema is currently available at
http://upload.wikimedia.org/wikipedia/commons/4/41/Mediawiki-database-schema.png

Table 4.4: Wikiprep page.

```
<page id="21504" orglength="15641" newlength="12231"
 stub="0" categories="16" outlinks="153" urls="2">
  <title>Nicole Kidman</title>

  <categories>1005386 2550044 1094152 853688
   1346991 1346993 1115029...
  </categories>

  <urls>
   http://news.bbc.co.uk/2/hi/science/nature/4317536.stm
   http://www.imdb.com/name/nm0000173/
  </urls>

  <links>15818 34749 324 31882 577 753 211405
    38473 13887 28159 15044 108956...
  </links>

  <text>
   Nicole Mary Kidman (born June 20, 1967) is an
   Academy Award-winning American-Australian actress,
   producer and singer. She was born in Honolulu, Hawaii
   to Dr. Anthony David Kidman and Janelle Ann...
```

So, even though this file is also monolithic, the information is more organized (for our extraction purposes). Moreover, it solves redirections, eliminates markup (hence producing a very "clean" text), discards metadata, and provides other useful information, such as the structure of Wikipedia's category network. In that sense, the structure of this XML file facilitates link information extraction, which can be placed into a database for faster access and querying. The main drawback with this version is that only the 2005 XML version is provided; to obtain a more recent version, the script has to be executed over the current XML dump. Such process demands, among other things, a considerable amount of disk space.

Despite its shortcomings, Wikiprep complies best with all desired properties: access to articles and links is relatively simple, titles are available, most dispensable material is already disregarded, and the text is cleaner in comparison with the rest of the sources.

Evaluating the pros and cons of each alternative (summarized in Table 4.5), we select Wikiprep as the most adequate option. The most powerful argument for preferring it over the other Wikipedia sources is that the former has been created for research purposes, while the rest are used primarily as sources for creating mirror sites. In addition, its satisfactory compliance with our requirements places it over the rest.

A point that we believe it is important to justify is the use of the Wikiprep 2005 XML file, instead of a more recent one. The main reason behind this choice was the lack of resources for

Table 4.5: Summary for link extraction alternatives

Option	Provider	Description	Pros	Cons
XML dump	Wikimedia	Contains every article.	Basis for other versions	Wiki markup, monolithic structure
WikiPrep XML	Third-party	Processed file set.	For research purposes, link information decoupled, solves redirections	Only 2005 version readily available
MediaWiki DB	Wikimedia	DB version for the XML dump.	Query support	Conversion costs, object management
Static HTML dump	Wikimedia	Bundle of HTML pages in a directory structure.	Individual article access	Useless static content, trie management
Web Crawl	Self	HTML bundle gathered by a spider or bot.	Individual article access	Useless static content, not encouraged

Table 4.6: Summary of link formats.

Version	Link Format	Example
XML dump	Markup	`[[a link]]`
Wikiprep	XML tag	`<links>id_0,id_1,...</links>`
HTML pages	HTML tag	`<a href="a_link">A link.</a>`
Mediawiki	DB table (from, title)	`(0, "A link")`

generating a more recent Wikiprep XML version by executing the script over a newer XML dump (as stated in Wikiprep's page, the process requires a considerable amount of disk space to produce several intermediate files, and we ran out of it). Another issue that we discovered while checking the partial results obtained by running the script was that, apparently, some minor changes in the Wikipedia markup syntax—obviously not tracked by Wikiprep—caused some "pollution" in the articles' text (this was not a major problem, but made us to prefer the 2005 version for creating the index).

By accounting the earlier, we briefly illustrate the differences between Wikipedia 2005 and Wikipedia 2009 (Table 4.7). As we can see, even when the number of articles has increased considerably, the amount of links did not change dramatically; therefore, we can expect for the 2005 version to provide a fair approximation (in size and complexity) with regard to the most recent Wikipedia snapshot.

A final remark about link information extraction involves the *final article dataset*. With regard to the former, category and list pages were excluded from the database. This was meant for a couple of reasons: first, because we are trying to generate high quality topics without this help (hence considering the inclusion of such pages as "cheating"), and, second, because

Table 4.7: Wikipedia 2005 vs. 2009 version comparison

	Wikipedia 2005	**Wikipedia 2009**	**Difference**
Articles	1 million	4 million	75% larger
Links	20 million	30 million	33% larger

we prefer to instead use them as reference classes for evaluation.

4.1.2 Basic topic construction algorithm

To achieve the goal of embodying a collection's topics, we intend to *search for highly interlinked groups* inside the corpus; this approach has several distinctive features:

(1) Inherently uses a graph representation.

(2) Assumes that topics will tend to concentrate into cohesive subgroups or community-like structures.

(3) Finds overlapping topics

Regarding (1), a graph representation is necessary in order to provide a hyperlink-based solution; therefore, this feature can be better understood as a requirement that our approach complies with.

Now, with respect to the intuitive notion stated in (2), one of our main assumptions is that topics resemble the structure of social communities (see Table 4.8), specially in the aspect of interactions: community members interact more among themselves than with respect to others outside the community. Similarly, we believe that documents of a certain topic will tend to link and be linked more frequently by members of the same topic than by non-members (documents of different topics). The previous assumption is not "hanging in the vacuum", but rather is strengthened by supporting literature; for example, as we have seen in other chapters, it is common to treat topics as communities ([42],[49],[77]). Also, the work by Menczer [92], establishes a precedent for several empirically-proven statements:

∇ "Link-content conjecture"—Pages are linked by similar content pages.

∇ "Link-cluster conjecture"—Pages relevant to a broad topic (query) are at a close distance from each other, where closeness is measured by path length.

Table 4.8: Analogy between a community (of persons) and a topic.

	Community	**Topic**
Group members	People	Documents
Bonding element	Common interests	Common thematic

We also believe the "topics behaving like communities" assumption is specially valid for academic Web collections, where information is supposed to be impartial and non-profitable (well, maybe except for Wikipedia donations). As a consequence, the corpus should be free of situations where competing sites deny to link each other for strategic, commercial purposes; this is actually a matter of consideration for algorithms that mine the Web's structure, and has moved authors towards "indirect linking" alternatives, such as hubs and authorities or co-citation and bibliographic coupling.

Another assumption, which is derived from the relation between topics and communities, consists of establishing a positive correlation between group density[5] (also referred to as "cohesion") and "topicality": *the more cohesive a group shows to be, the more likely it is to represent a topic*. In fact, the former constitutes our starting point, since we can describe our topic construction approach as being able to find clusters of *maximum cohesion*[6]; this leads us as well to depict our approach in terms of optimization.

To understand better the kind of optimization problem we are dealing with, let us first provide a metaphor by visualizing topics as the "peaks" of a given surface, which is our search space (see Figure 4.2); then, the construction task must care about two primary aspects: finding the location of each peak (*coverage*) and climbing to the top of it (*quality*). If coverage is seen as a horizontal axis and quality as a vertical one, then we can realize that we actually have to handle a multi-objective optimization problem; obviously, a configuration lying on the Pareto front (Figure 4.3) would be our optimal solution, but for initially approaching the problem (which is our case for now), any feasible solution (in the terms we discuss next) might be enough.

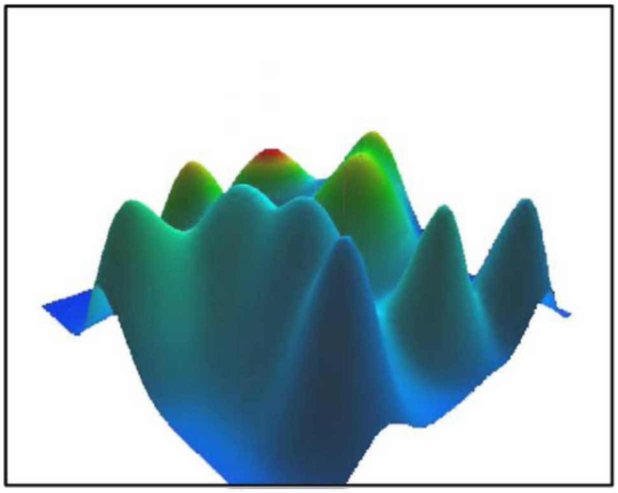


Figure 4.2: Topics as peaks on a surface.

It is significant to point out that, for our purposes, coverage and quality do not have the same importance. In that sense, we are more committed to quality, and it is our main concern for building the construction algorithm. Coverage is not by any cost to be underestimated,

[5] Let us note that, throughout the chapter, the terms of *cohesion*, *fitness function*, *density*, and *relative density* shall be used interchangeably. In any case, they refer to the same notion.

[6] The term "maximum" here is not employed strictly, as this would imply that we are looking for "cliques", which genuinely represent complete maximal subgraphs. In our case, the structure to detect is more relaxed and basically consists of a group whose internal link number is equal to or surpasses its external link number (a community).

but it will be second in importance for several reasons. First, it is far more subjective than quality, because it is not trivial to state whether *all* the topics of a collection have been found; even if there was an external structure (e.g. a directory) for comparison, still the *discovery* of previously unknown topics is hard to assess. Thus, for simplifying this issue, we might instead measure the ratio *clustered documents* to evaluate coverage. A second reason to relegate coverage is that it actually competes with quality, up to the point where an increase in the former implies a decrement in the latter. For example, if we are constrained by time, an extensive search to find an optimal cluster is not always possible; therefore, we would have to accept a cluster of less quality to complete the process on time. So, quality is to be chosen over coverage, as long as this second aspect is not severely compromised. *Consequently, our approach is capable of handling (softly) the two axes of the optimization problem.* By softly, we mean that the final choice is not automatically enforced.

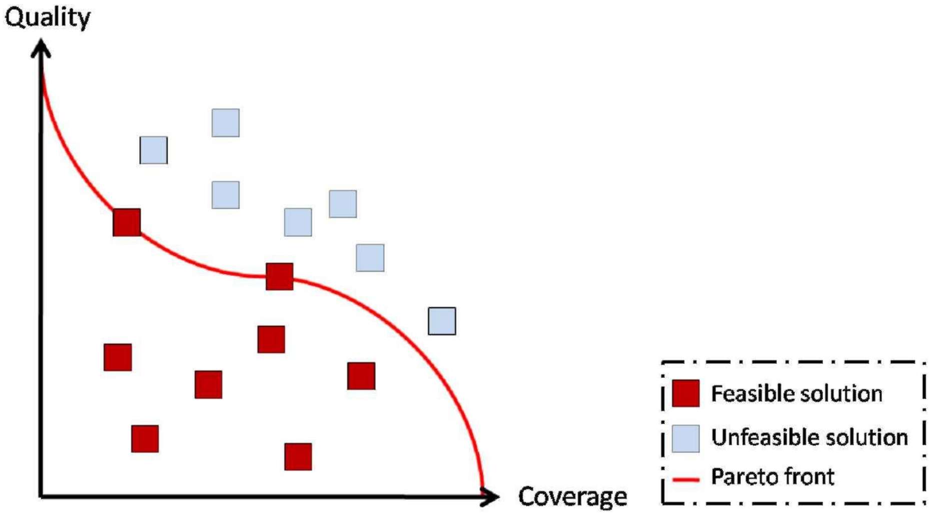


Figure 4.3: Pareto optimality.

Returning to our visual metaphor, an interesting point is peak height. While we attempt to find as many peaks as possible, the fact that some of them are more elevated than others becomes irrelevant (as long as they are all peaks). This makes us to realize that we are not seeking global, but *local* optima; to achieve the former is by no means trivial. In fact, a common problem that modularity-based graph partitioning algorithms have to face is how to avoid producing a single cluster containing all nodes (which happens to be the global optimum). Furthermore, common blind optimization techniques usually search for the global optimum and, in some other cases, the global optima. *Consequently, our approach is capable of finding local optima.*

Not content with attempting to solve a multi-objective problem in a multidimensional space (each corpus document accounting as one different dimension) where the goal is to obtain all optima, we want to be able to generate *overlapping clusters* (point (3) on our approach feature list) as well. The argument for producing such groupings is two-fold: it seems more natural to conceive a document as being able to belong to distinct topics, and this leads also to consider that, for our particular context, (disjoint) partitions can lower quality. *Consequently, our approach is able to produce a non-exclusive clustering.*

Actually, generating overlapping clusters is something that has not been done before with the general approach we are about to discuss.

Using the GLC approach

The corner stone of our topic construction algorithm is the *GLC approach*; in that sense, our general design is guided by the following "principles":

(i) The entity to cluster is a graph.

(ii) Each cluster is to be created in the vicinity of a given point (node) by optimizing a graph-theoretic fitness function via local search.

(iii) A cluster does not necessarily depend on other clusters for its creation.

As we can see, this approach enables our topic construction approach to acquire the three features mentioned at the previous section (graph-based, community-based, overlapping). Let us explain this with some more detail.

First, by being a purely link-based approach (i), it offers the capability to cluster with just link information—unlike other link-based methods that require content for working (e.g. HITS and hybrids). Moreover, by establishing that clusters are to be created in the locality of a given point (ii), which can be seen as equivalent to community identification, approaches flowing from GLC can find local optima consisting of cohesive subgroups. Third, by considering that clusters are independently constructed (iii), we can have overlapping groups.

Furthermore, the GLC approach is:

Prepared to deal with complexity Because it works locally and produces each group independently, it copes with large-sized graphs and allows parallelization in a *natural* way (this probably constituting its major advantage). This enables us to work with collections of tens or hundreds of thousands of documents.

Suited for directed graphs While there are approaches (e.g. maximum flow communities, modularity maximization, chameleon clustering) that require the input graph to be transformed into a certain type (undirected, unweighted, . . .), so far GLC does not have a restriction with regard to this aspect. Therefore, we can use our Wikipedia graph *as it is*.

Clear and traceable Unlike other methods (spectral clustering, SOM's, and other "numerical" approaches), it does not use obscure calculations, and thus is easier to trace—specially for enhancement the specific algorithm we are employing.

Founded on optimization techniques Its functioning is based on known local search strategies, and this makes the approach more *solid*.

Now, considering that GLC is only our *skeleton*, several specific design choices have to be made in order to build a *concrete* algorithm suited to our purpose of finding topics.

Concrete GLC algorithm

Recalling from the formal framework and according to point (iii) of the GLC approach, our GLC algorithm can be seen as consisting of two functions: the *basic* function (F) and the *extended* function ($\mathcal{X}$). The former is intended for the construction of a *single* cluster, and is used repeatedly by the latter to obtain a clustering (*set* of clusters) over the entire collection. In that sense, to avoid confusion, we will refer to the basic function as the *construction function*, and the extended function as the *clustering function*. Also, it is relevant to note that while the construction function is related to quality, the clustering function is instead aligned towards coverage; therefore, the design of the construction function is to have a higher priority.

Construction function

Two components or "parameters" that have to be defined for designing a basic version of the construction function are the *search strategy* and the *fitness function*; our approach consists of *hill climbing* and *relative density*, respectively. Let us first describe each of these components and then discuss their selection.

Inspired by the movement of a mountain climber, the *hill climbing (greedy local search) strategy* attempts to improve a given initial solution at each step by examining its neighborhood (that is, moves always "uphill"). If the current solution cannot be improved further, the algorithm stops (a *local* optimum has been found). However, if we aim for the global optimum, the algorithm is prone to "getting stuck" in a local optimum that does not necessarily represent the best value of the search space (this can be inconvenient for several contexts). The three fundamental design parameters of this strategy are:

Creation of the initial solution. This involves, for instance, deciding whether to start from a random solution.

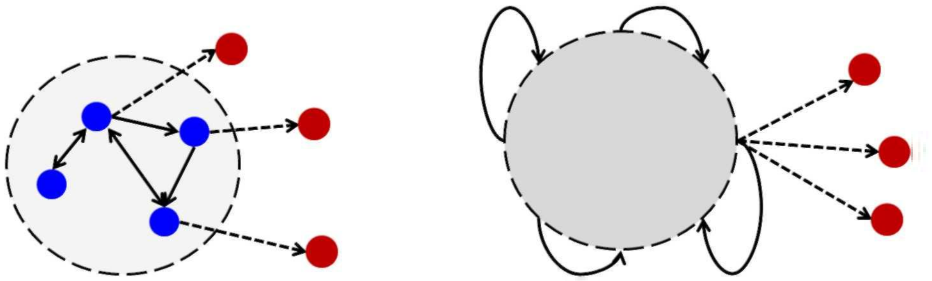
Choice of the neighborhood. This parameter comprises the disjunction of either searching on a small neighborhood (lower costs, but lower quality) or a large one (higher costs, but higher quality).

Improvement strategy. There are two options here: choosing at each step the first improvement or choosing the best improvement (implies searching the entire neighborhood).

This is the basic and easiest local search heuristic.

Denoted as ρ in Section 2.1.4, *relative density* has been defined as a cluster's *ratio of internal links*. For example, let us assume that we have a cluster $\mathcal{C}$ like the one depicted in Figure 4.4a; as we can see, this cluster (which contains four elements) has 6 internal links, and 9 links in total. Consequently, $\rho(\mathcal{C}) = \frac{6}{9} = 0.67$. Another way to visualize relative density is by considering the cluster as a node in pseudo-graph[7](see Figure 4.4b) where every link between a pair of internal elements is seen as a self-loop; then, relative density results from dividing the number of self-loops by the degree of the "pseudo-node".

[7] A *pseudo-graph* (or pseudo-graph in our case) is a graph where it is allowed to have multiple edges for the same pair of nodes and/or for the same node ("self-loops"). This kind of graphs are also known as *multi-graphs*, although this second name is discouraged when there are self-loops.



(a) Element ("typical") view

(b) Pseudo-graph view

Figure 4.4: Different forms of visualizing a cluster's internal links

Once having described these two components in general terms, let us state some of the advantages that result from their selection (both individually and as a combination). On one hand, with respect to hill climbing, an ironical remark for this option consists of acknowledging that one of its less convenient features when dealing with global optimization, *finding local optima*, can ultimately be used to propel our construction function (keeping in mind that this also implies having a fuzzy delimitation between "low" local optima and poorly constructed clusters). Furthermore, it has been stated that although the algorithm is simple, it has been successful in a variety of combinatorial optimization problems [119]. Also, it only requires three parameters to be defined, and this facilitates its implementation and testing; of course, more parameters will subsequently be added to cope with the domain, but precisely for this reason it seems convenient to have as few parameters as possible at the beginning. This helps us to realize two important issues: it appears appropriate to set parameters that are domain-oriented, i.e., imposed by our specific context, and the number of parameters is critical, as an exhaustive search for the right combination of these implies more effort when this number increases (we might, in fact, achieve a "combinatorial explosion" effect).

With regard to relative density, more than anything, it is a metric whose values fall within a clear range $[0, 1]$; specially for exploratory tests, this attribute helps us to have, at a simple glance, a clearer notion of the clusters' cohesion. Furthermore, relative density allows to measure how "communitarian" a cluster is; for instance, because we know that a community (in its weak sense) has more internal than external links, it is possible to realize that communities are clusters with $\rho >= 0.5$, and this is useful for filtering results, since it enables to have a uniform criterion for discarding groups that actually can be considered as noise (we will go over this later on).

In an overall sense, the combination of hill climbing and relative density follows an Ockham's razor principle[8], since the main purpose is to start with the simplest form of construction and add sophistication as needed. Actually, the use of this principle is recommended in local search strategy comparison works, such as the one of Pirlot [111]. In that sense, as recently stated, we prefer to tune parameters strictly related with our domain (topic construction) than

[8] This principle aims for simplicity, as it assumes that this kind of solution tends to be the best one. According to the Merriam-Webster dictionary, Ockham's razor (also written as *Occam's razor*) is "a scientific and philosophic rule that states that entities should not be multiplied unnecessarily, which is interpreted as requiring that the simplest of competing theories be preferred to the more complex or that explanations of unknown phenomena be sought first in terms of known quantities".

to tune parameters of the search strategy *per sé*. Moreover, the selected combination has already been used to gather preliminary results and has so far achieved an empirical satisfactory performance, according to our initial evaluation criteria [46]. Nevertheless, it is important to remark that there is a considerable number of other search strategies (e.g., see the text by Sait and Youssef for traditional methods [119] and Weise's discussion of other specific and recent ones [146]) and fitness functions (e.g. the group quality metrics described in Section 2.1.4) available. Regarding this aspect, an exhaustive evaluation of all possible combinations lies out of our scope; as a result, we cannot either assure or deny that our chosen combination is the best one. Perhaps this exhaustive search for the optimal GLC combination for the topic extraction domain could be left as an open issue for future works.

The construction function algorithm that uses the aforementioned combination is shown in Algorithm 1; it seems important to remark that this is more of a didactic version.

Algorithm 1 Construction function.

Description: Receives as input a seed S (initial set of documents) and returns a document cluster C_i. A new element is added to the cluster at each iteration by choosing the candidate l_{best} that yields the best density improvement; when density can no longer be increased, the algorithm stops. Each time a new element is added, its neighbors become candidates for cluster inclusion at the next iteration.

```
function CONSTRUCT-GLC-TOPIC($S$)

    $\mathcal{C}_i \leftarrow S$                                  ▷ Initialize the cluster with the input seed.
    repeat
        $\rho_{\text{curr}} \leftarrow \rho(\mathcal{C}_i)$                      ▷ Store current density in $\rho_{\text{curr}}$

            ▷ The candidate set $L$ is generated from the neighborhood of the current cluster $\mathcal{C}_i$; that
        is, from the union of the neighbors of the cluster's elements. Note that $L$ and $\mathcal{C}_i$ should always be
        disjoint sets: cluster members are not to be considered in the candidate set anymore, even if they
        keep appearing in the neighborhood of newly-added elements.
        $L \leftarrow \{L_1 \cup \ldots L_{|\mathcal{C}_i|}\}$
        remove $\mathcal{C}_i$ from $L$

        $l_{\text{best}} \leftarrow \arg\max_{l \in L} \rho(\mathcal{C}_i \cup l)$

            ▷ For the sake of simplicity, let us assume that if $l_{\text{best}}$ does not improve the value given by
        $\rho_{\text{curr}}$, it becomes a void element and $C_i$ is not affected by its addition (therefore yielding the same
        density).
        add $l_{\text{best}}$ to $\mathcal{C}_i$
        $\rho_{\text{new}} \leftarrow \rho(\mathcal{C}_i)$                       ▷ Store new density in $\rho_{\text{new}}$
    until $\rho_{\text{curr}} = \rho_{\text{new}}$

    return $\mathcal{C}_i$
end function
```

Table 4.9 illustrates how the construction function works by using our small graph from

Chapter 3.

Table 4.9: Basic construction example.

Initial state: $\mathcal{C} = \{\text{lotr1}\}, \rho(\mathcal{C}) = 0$					
		$\mathcal{C} \cup c$			
	Candidate (c)	deg_{int}	deg_{ext}	deg	ρ
Iteration 1:					
	frodo	**2**	**7**	**9**	**0.22**
	gandalf	2	7	9	0.22
	lotr2	2	8	10	0.2
	lotr3	2	8	10	0.2
	peter	2	8	10	0.2
$\mathcal{C} = \{\text{lotr1, frodo}\}$					
Iteration 2:					
	gandalf	**6**	**7**	**13**	**0.46**
	lotr2	6	8	14	0.429
	lotr3	6	8	14	0.429
	peter	4	9	13	0.308
$\mathcal{C} = \{\text{lotr1, frodo, gandalf}\}$					
Iteration 3:					
	lotr2	**12**	**6**	**18**	**0.667**
	lotr3	12	6	18	0.667
	peter	8	9	17	0.471
$\mathcal{C} = \{\text{lotr1, frodo, gandalf, lotr2}\}$					
Iteration 4:					
	lotr3	**20**	**3**	**23**	**0.870**
	peter	16	6	22	0.727
$\mathcal{C} = \{\text{lotr1, frodo, gandalf, lotr2, lotr3}\}$					
Iteration 5:					
	peter	**26**	**1**	**27**	**0.963**
$\mathcal{C} = \{\text{lotr1, frodo, gandalf, lotr2, lotr3, peter}\}$					
Iteration 6:					
	oscar	**27**	**0**	**27**	**1.0**
$\mathcal{C} = \{\text{lotr1, frodo, gandalf, lotr2, lotr3, peter, oscar}\}$					

CLUSTERING FUNCTION

Regarding the clustering algorithm (portrayed in Algorithm 2), its basic behavior is fairly simple. The two aspects of such behavior consist of creating a seed list (such that 50% of document coverage can be assured at least) and selecting the next seed for the construction function to use.

Regarding the first aspect, our first approach consisted of creating a seed list of a fixed size k, where k is obtained by applying the "thumb-rule" used for calculating the number of centroids in k-means [87]; this rule, as Eq. 4.1 shows, simply consists of choosing as many seeds as the square root of half the collection's size ($|C|$). However, with this initial list only a small fraction of the corpus was being covered by the time of clustering (less than 5%). A more radical alternative is to construct an *exhaustive dynamic seed list* (actually, this is the one currently used by our algorithm) by including all documents ($\mathbb{S} = V$) and iteratively eliminating from the list those that have already been included into a cluster. When the list is empty, the clustering is considered as finished; for a construction function that has no element removal (we will discuss element removal later), this method should achieve a document coverage of 100%. Details with respect to seed ordering include a more fine tuning, and thus shall be discussed on the next section.

$$k \approx (\frac{|C|}{2})^{\frac{1}{2}} \tag{4.1}$$

Algorithm 2 Clustering Function

Description: Receives as input a set of seeds $\mathbb{S}$ and returns a clustering (group of groups) $\mathbb{C}$. Every time an individual cluster $\mathcal{C}$ is created, all of its elements (including the seed S_i where $\mathcal{C}$ was obtained from) are removed from $\mathbb{S}$. The clustering process terminates when there are no more seeds to process.

```
function CLUSTERING($\mathbb{S}$)                ▷ Construction of topics from a seed set

    repeat
        $S_i \leftarrow$ CHOOSE-NEXT-SEED($\mathbb{S}$)        ▷ Get the next seed to process
        $\mathcal{C} \leftarrow$ CONSTRUCT-GLC-TOPIC($S_i$)
        add $\mathcal{C}$ to $\mathbb{C}$
        remove $\mathcal{C}$ from $\mathbb{S}$
    until $\mathbb{S} = \emptyset$

    return $\mathbb{C}$                               ▷ Return set of clusters
end function
```

4.1.3 Algorithm details and fine tuning

The algorithms presented on the previous section have as its main goal to introduce the approach in a simple, general form. However, several substantial details and design choices still remain to be revealed—for instance, the concrete neighborhood searching procedure. As a result, the current section aims to describe such details and present support for the design decisions involved in the algorithm's fine tuning. In some instances, results from exploratory experiments shall be provided as evidence.

About enhancements in general, these obey the parameters of local search and can also be divided according to whether they are for the construction function or the clustering function. In that sense, we wish to provide a classification (Table 4.10)

Element removal.- Cluster refinement by element elimination.

Candidate ordering.- Order in which nodes are revised.

Neighborhood type.- Whether to choose all the neighbors of a node for the candidate set or only the ones linked by the node.

Seed ordering.- Criterion for ordering the seed list.

Seed expansion.- Number of elements the seed should include.

Secondary cluster.- Determination of a "rest" cluster during construction, i.e., deciding what procedure to follow with removed elements.

Quantum Time limit inclusion.

Table 4.10: Enhancement classification

	Initial solution	**Neighborhood choice**	**Improval strategy**
Construction	Seed expansion	Neighborhood type	Element removal Candidate ordering Quantum
Clustering	Seed selection Secondary cluster	–	Quantum

For evaluating clusterings with distinct enhancement values, several general criteria have been defined (Figure 4.5); these flow from two primary requirements: *feasibility* and *quality*. While quality comprises *cohesion*, *document coverage*, and *redundancy*, feasibility concerns *time* and *memory* restrictions.

Cohesion.- We will use this term for referring to our formerly defined concept of "quality" and avoid confusions. Now, with respect to this criterion, enhancements that show to improve it or not demerit it severely (in the case of savings or heuristics to cope with feasibility) are to be awarded a better evaluation.

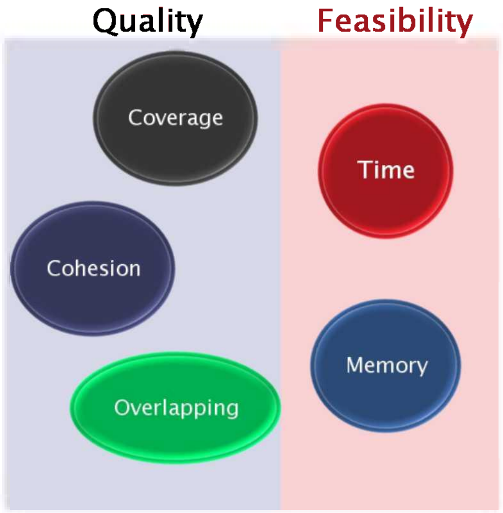


Figure 4.5: Evaluation criteria.

Coverage.- As it was stated earlier, coverage is less important than cohesion; in that sense, refinements that decrease it are not to be dismissed, unless they severely compromise it. If the document coverage is considered to be low, those enhancements that improve it are to receive a favorable evaluation.

Redundancy.- By "redundancy", we refer to having a considerable number of clusters that basically confer the same documents (a very high overlapping). For our quality criteria, this is the one of the least importance; consequently, it is to be considered as a "soft" or desirable requirement, in the sense that our main concern with it takes place when it affects time severely (a redundant cluster is a waste of time). A final consideration confers how redundancy is to be measured; while conventionally it implies pairwise comparisons for registering the average overlapping, we will measure it in an approximate form by dividing coverage (c) by the total number of document occurrences in the clustering—that is, the sum of cluster sizes ($|\mathcal{C}_1| \ldots |\mathcal{C}_{|\mathbb{C}|}|$). See Eq. 4.2.

Memory.- The primary concern of this heading is to avoid refinements that exceed the available memory. Logically, any refinement that surpasses our memory limitations is to be directly discarded.

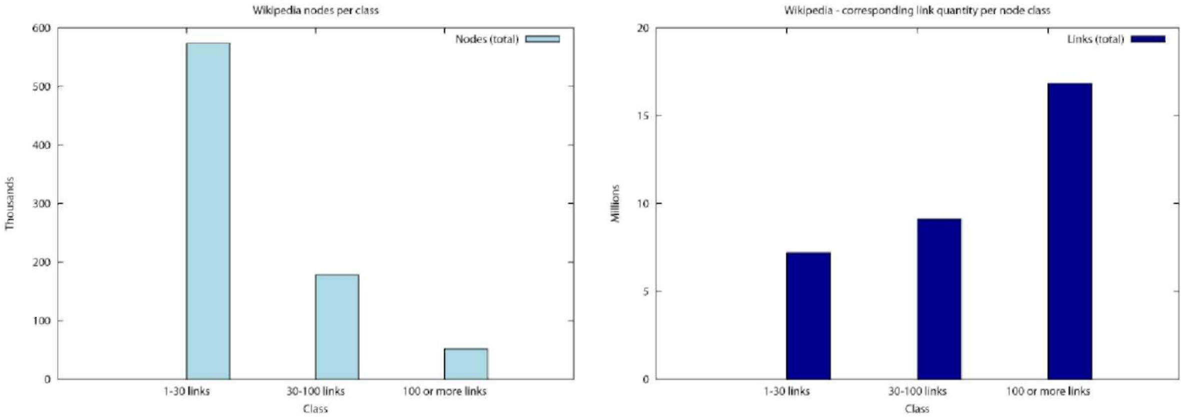
Time.- This limitation consists of not letting our enhancements to exceed a "considerable" time; this measurement is not strict.

$$r = \frac{c}{|\mathcal{C}_1| + \ldots + |\mathcal{C}_{|\mathbb{C}|}|} \tag{4.2}$$

Wikipedia sub-collections

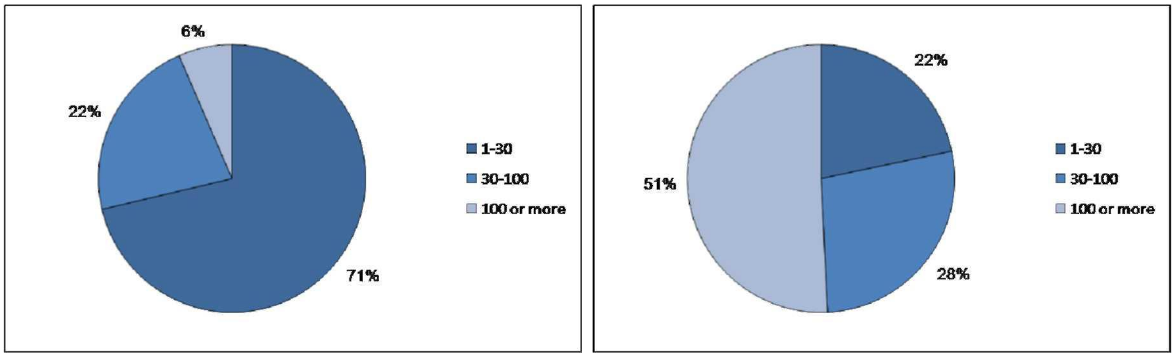
Because testing variations of the basic algorithm with the complete Wikipedia collection can be considerably expensive (mostly in terms of time), a viable option is to take distinct fractions of the corpus and automatically generate smaller *sub-collections* for running pilot tests.

A key issue to be firstly approached for the generation of sub-collections is the *selection criterion*. For instance, an intuitive option is to execute queries and gather the matching sub-collections. However, we believe the former is not the best alternative for several reasons. On one hand, if we are working with hyperlinks, it is essential to assure that the resulting sub-graph will be *connected*; so, unless the generated collections are augmented with procedures similar to the ones used by HITS or maximum-flow communities, it is not so trivial to obtain connected corpora from simple queries. Another consideration that leads us towards searching for another alternative—probably the strongest for dismissing query usage—concerns avoiding *bias*. To have an "unbiased" prune (that additionally provides an assorted collection of topics with a higher probability than a simple query), it seems fair to prune based on a more "neutral" trait, such as a node portion that is densely linked. To find this portion, it seems convenient to explore Wikipedia's structure and analyze how links appear to be distributed in relation to articles (nodes).



(a) Node quantity (b) Link quantity

Figure 4.6: Wikipedia node and link quantities

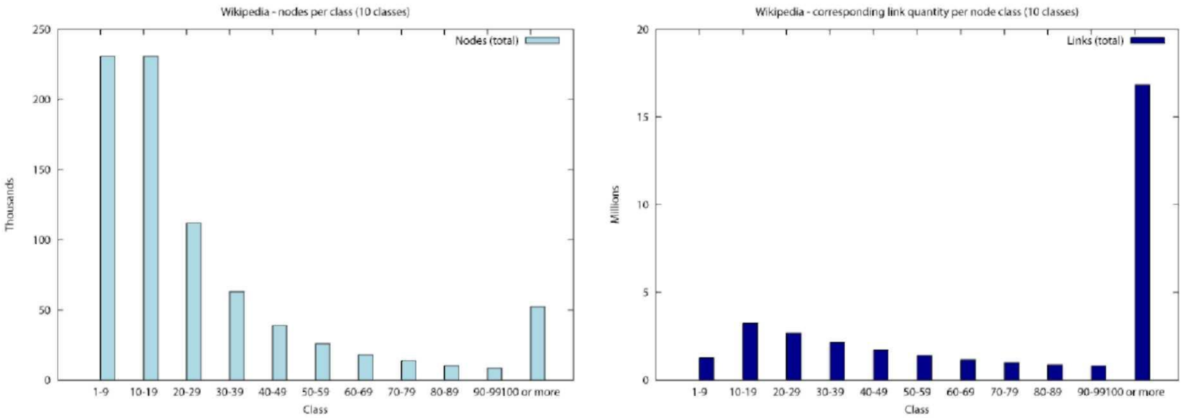


(a) Node percentage (b) Link percentage

Figure 4.7: Wikipedia node and link percentages per class

If we try to classify nodes according to their amount of links (both in and out-links), they can fall into three intuitive, broad categories: "sparse" (1-30), "medium" (31-99), and "dense"

(100 or more). This classification would appear as illustrated in Figure 4.6a; additionally, Figure 4.6b presents the corresponding amount of total links per class. By analyzing also the percentages that these quantities represent, we can see that approximately 70% of the nodes covers roughly 20% of the existing links (very much like a Pareto distribution). A more fine-grained distribution is shown in Figure 4.8.



(a) Node quantity

(b) Link quantity

Figure 4.8: Wikipedia nodes

Link distribution findings lead us to consider taking, perhaps, a portion of nodes that does not lie on the sparse class. On one hand, as we can see, nodes from other classes are denser and more robust for running our algorithm; also, since we will need to erase arcs, we need to make sure that we will be left with enough connections to make tests. As a result, it seems adequate to select a considerable number of nodes having more than 30 arcs—however noting that this range would still be broad if we wanted to generate a small sub-collection. Regarding that aspect, the size of the sub-corpus will actually depend on the specific purpose it is to be created for; obtaining a sub-graph of a given size is not trivial, though, but we can estimate the size that will result from selecting a certain “link range” by counting the number of nodes whose links fall into the same class.

The procedure for creating a sub-collection that we carried out involves three basic steps (besides choosing a link range that seems to correspond to the desired size): select nodes within the range, erase arcs that link a selected node to a non-selected one and vice versa, and eliminating nodes left unconnected (in the eventual case where all the arcs of a certain node are deleted on the previous step). This procedure is shown more formally in Algorithm 3. Regarding implementation, selection and deletion can be easily accomplished with conventional query language operations; the process is not computationally heavy, as it usually takes a more than reasonable amount of time (minutes, in the case of a personal computer).

For our pilot tests, two sub-collections were generated: one representing the “happy days” (*Quantumpedia*) and the other one depicting a more realistic case (*Micropedia*). Outstanding properties for these corpora are shown in Table 4.11. Regarding “Quantumpedia”, it has been considered as an easy case not only because of its size, but also for containing a considerable number of closed or nearly-closed communities. The aforesaid situation takes place mainly due to the amount of pruning that is being done (since more than 90% of the nodes and

Algorithm 3 Procedure to generate a sub-collection.

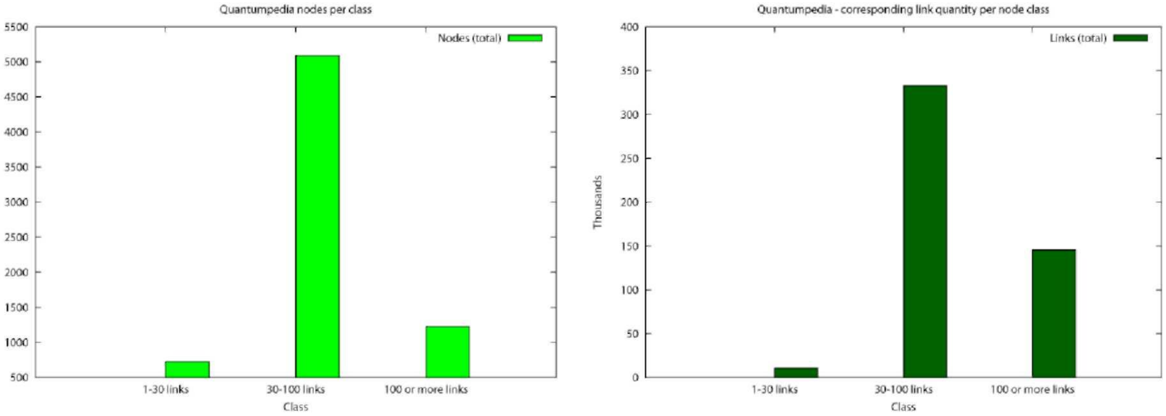
Description: Receives an “arc-amount” range that starts at *lowerBound* and ends at *upperBound*, and eliminates from the vertex set V those elements that do not belong to the range (along with their corresponding arcs).

```
procedure PRUNE(rangeLowerLimit, rangeUpperLimit)

    for all $v \in V$ do                                   ▷ Select vertices in range
        if rangeLowerLimit $< \Gamma(v) <$ rangeUpperLimit then
            add $v$ to $V_p$
        else
            remove $v$ from $V$
        end if
    end for

    for all $e \in E, e = (u, v)$ do                       ▷ Delete arcs with endpoints outside selection
        if $u \vee v \notin V_p$ then
            remove $e$ from $E$
        end if
    end for

    for all $v \in V$ do                                   ▷ Delete disconnected vertices
        if $v$ is disconnected then
            remove $v$ from $V$
        end if
    end for

end procedure
```

links are eliminated); this causes cohesive (or "cliquish") groups of nodes to be left uncovered. For instance, suppose the existence of a group of 100 elements, where each node links and is linked to the rest (this yields at least 99 links and falls within the range for keeping); internal links of this group are conserved because they all will tend to fall on the range, but external links are likely to be erased with a very high probability, because linked / linking nodes are to fall outside the range. What makes this probability so high is that almost every node is to be out of the range (in that sense, note that this instance of our pruning algorithm represents a naive, "cheap" method for exposing communities). It is also important to note that, while Quantumpedia indeed contains several "islands", the greatest strongly connected component includes more than 70% of the sub-collection's nodes (about 5,000 vertices), which is actually an interesting property, considering that Wikipedia's largest SCC concerns approximately 85% of the nodes. With respect to link distribution, Figures 4.9 and 4.10 show quantities and percentages according to node classes, respectively. As we may observe, despite the massive pruning, the majority of the kept nodes still falls into dense classes.

Table 4.11: Wikpedia sub-collections

	Link range	**Vertices**	**% kept**	**Arcs**	**% kept**
Quantumpedia	95-150	7,039	0.77%	452,072	2.11%
Micropedia	60-100	37,358	4.1%	743,518	3.47%

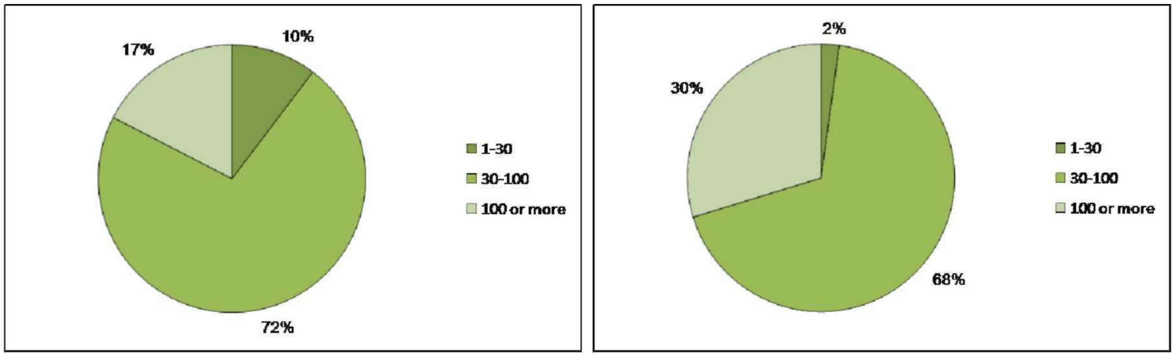


(a) Node quantity

(b) Link quantity

Figure 4.9: Quantumpedia node and link quantities

A final remark regarding "Quantumpedia" is that its creation responds not only to the need of observing the algorithm's performance with a basic case, but the need to initially assess its performance in a comparative fashion as well; with respect to this facet, we have discovered that comparison experiments with "classical" approaches are basically unrealizable at the Wikipedia level because of scaling reasons (which therefore gives us an insight about the advantages of our method), and it was necessary to generate this very small sub-collection for applying these approaches and comparing their results against ours. Details about this exploratory analysis are discussed in [47].

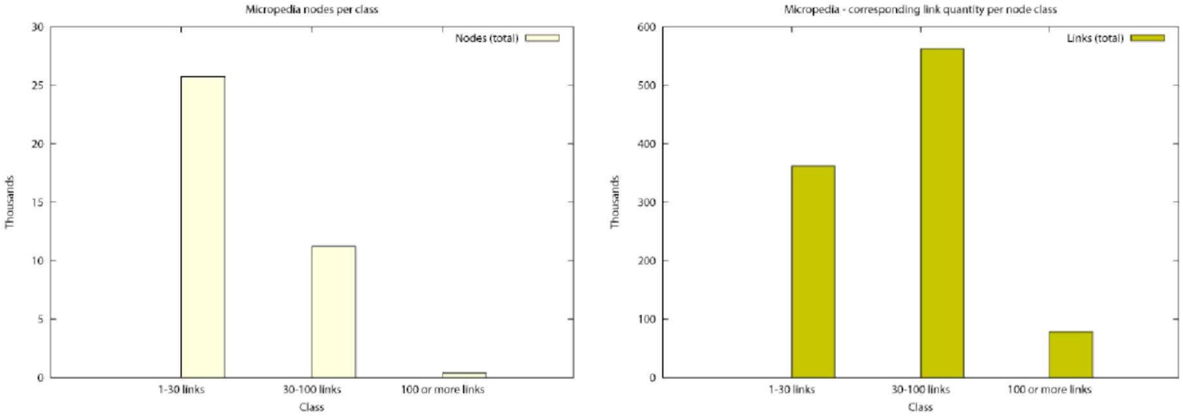


(a) Node percentage

(b) Link percentage

Figure 4.10: Quantumpedia node and link percentages

Regarding "Micropedia", its selected link range obeys one principal purpose: generating a sub-collection with a higher complexity than Quantumpedia, but light enough to be clustered in a "rational" time. To comply with this pair of tension points, several ranges were tested on a trial-and-error basis; finally, a range of 60-100 links per node was chosen as the best alternative for Micropedia. Link distribution for this sub-corpus can be appreciated in Figures 4.11 and 4.12; it is interesting to notice that on this Wikipedia fragment, some nodes of the densest class (100 links) did not get their links removed with the pruning.



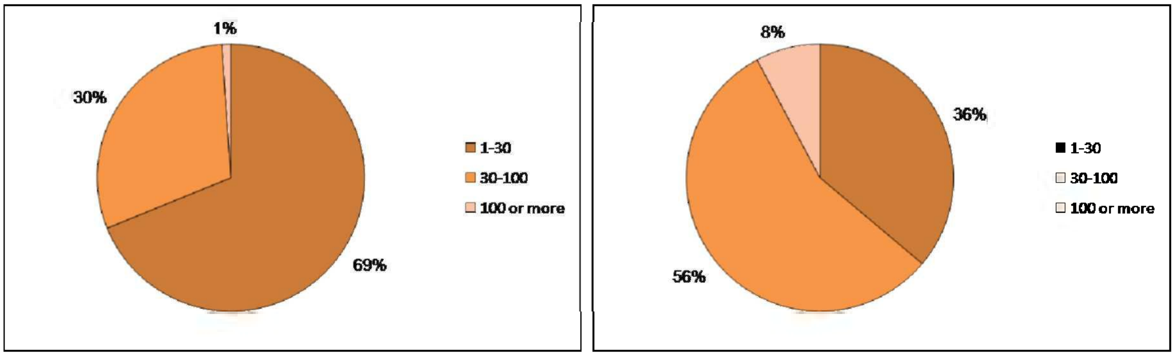
(a) Node quantity

(b) Link quantity

Figure 4.11: Micropedia node and link quantities

A final statement about Micropedia concerns its importance for our exploratory experiments; because Quantumpedia is not a very complex case, results obtained with Micropedia are to receive a higher relevance. In that aspect, we are more fond to showing results with this sub-collection.

As the reader can note, the gathered sub-corpora are relatively small if compared to the total size of Wikipedia; as it has been already discussed, such decision was based on the fact that exploratory tests over the whole corpus are not feasible due to time limitations, and it has also been stated that such decision implies both advantages and disadvantages. Regarding



(a) Node percentage

(b) Link percentage

Figure 4.12: Micropedia node and link percentages

disadvantages, we must have in mind that a different behavior could take place when performing tests over the entire corpus (which is larger and much more complex). For this reason, whenever appropriate, we will also offer views with isolated clusters—considering as well that proper clustering properties (document coverage and redundancy, for example) are not measurable by obtaining a considerable number of individual groups. Regarding advantages, an important aspect is that we can use these small corpora to quickly discard expensive or poor-result yielding alternatives (similar to having a *lower bound*). Furthermore, tests over these corpora can show to be meaningful if we take into account that our sub-collection sizes are equiparable to (and exceed in some cases) those reported in literature. Table 4.12 presents a collection of such sizes (both from Wikipedia and link analysis relevant works).

Table 4.12: Collection sizes reported in literature

Author	**Method**	**Corpus**	**Vertices**	**Arcs**
Girvan	Edge betweenness	Computer-generated graph	128	922*
Botafogo	Bi-connected components	Hypertexts	353	2,751
Meneses	Graphs and vectors	Central American webs	456	3,283*
Wartena	Text co-occurrences	Dutch Wikipedia	758	5,458*
Kleinberg	HITS	Query Web crawls (base set)	5,000	36,000*
Huang	Random walks	Query Web crawls	6,634	65,536
Virtanen	GLC	Chilean Web	32,148	1,357,661
Girvan	Modularity Maximization	Amazon.com purchasing network	409,687	2,464,630
*= Number of arcs is not reported. A rough estimation is calculated by taking into account that the reported out-degree of webpages is of 7.2 on average [22].				

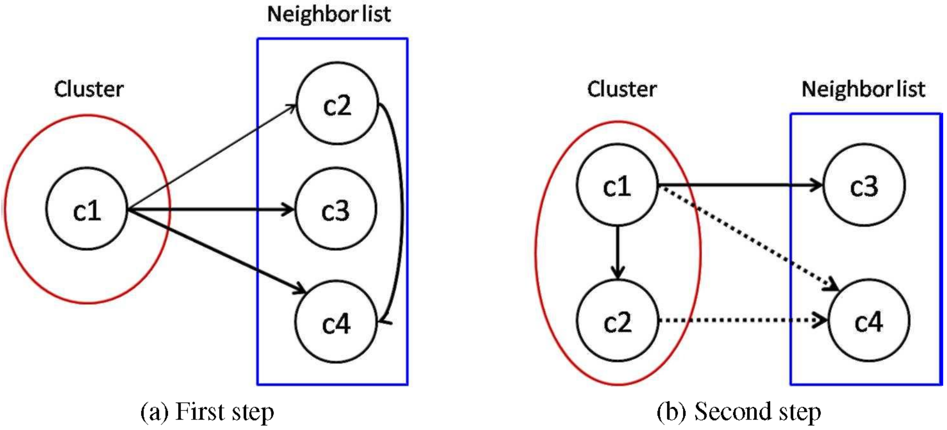
Candidate ordering

Line 7 of Algorithm 1 is the heart of the construction function–although it is merely the general, and to some extent "ideal" version of the improvement strategy. Consequently, let us start by describing this strategy with more detail, and then introduce the heuristics used for attaining high quality quicklier.

The first aspect concerns selecting the essential improvement method; as already stated, the two available possibilities are "first improvement" and "steepest descent" (best improvement). If we consider that a) the improvement strategy is carried out every time *one* neighbor is to be chosen and that b) in our domain neighborhoods are likely to get large for a significant node portion of the graph, first improvement seems the most viable choice. In that sense, our specific implementation of local search is First-Choice-Hill-Climbing (FCHC[9]).

Therefore, a fundamental issue for achieving the best results without having to go through the entire neighborhood repeatedly is to *place the most promising candidates at the top of the neighbor list*. Heuristically, this is equivalent to finding a criterion for ordering candidates that provides "hints" about the worthiness of each one of them, without producing a considerable overhead. This last aspect is particularly delicate, because element addition to the cluster is, in fact, the most expensive operation.

So, we need to have candidate information available, but at the same time avoiding to execute extra queries on our link database. On that side, a plausible alternative is to look for *tacit* data that we can use—perhaps data that can be gathered with each iteration. A datum of this type corresponds to counting the *number of internal links that a candidate would contribute with if added to the cluster*.



(a) First step (b) Second step

Figure 4.13: Candidate ordering heuristic example.

First, let us explain what we mean by the "number of internal links a candidate would contribute with if added to the cluster". Assuming that we have an initial configuration such as the one depicted in Figure 4.13a, it is possible to see that, by adding either candidate of the neighbor set (c_2, c_3, c_4), the internal degree of the cluster would be equal to 1. Now, if we chose for some reason c_2 to be included into the cluster, the configuration would look as shown

[9] A.k.a. "simple hill-climbing". A stochastic variation of FCHC consists of revising candidates at random and selecting the first one that shows improvement [118].

in Figure 4.13b. Visually, we can note that if we add c_3 on the next iteration, our number of internal links would increase to 2; on the contrary, if we include c_4, the number would increase to 3. Considering that we have no other information available for pre-evaluating these candidates, exploring first c_4 seems reasonable.

The prospect number of internal links can be easily calculated by keeping track of the number of times each neighbor has appeared on the neighbor sets of included nodes—that is, we incrementally update how many of the nodes of the current cluster link the neighbors that remain "outside" of the group. To exemplify this calculation, let us assume the existence of a *neighbor table* T, which consists of a set of entries in the following form:

$$(\text{neighbor}, \text{occurrences})$$

So, for instance, the initial neighbor table for our previous example would be $T = \{(c_2, 1), (c_3, 1), (c_4, 1)\}$; after the inclusion of c_2, the table would be updated and left as $T = \{(c_3, 1), (c_4, 2)\}$. By recording the occurrences of each neighbor, we are able to sort candidates by their prospect number of internal links. Another advantage from storing this information is that we are creating a "lookup table", and this avoids having to obtain the same data over and over again. This approach is coherent with efficient strategies, such as top-down dynamic programming.

As we will see later, we can either consider as neighbors those nodes that both link and are linked by the current cluster, or only those that link to it. In the first case, even though the neighborhood size increases considerably, this enables us to have more precise information about candidates if we sum the amount of links from the cluster to the candidate and the amount of links from the candidate to the cluster—assuming that if a candidate links and is linked by the cluster, the probability of increasing density is higher. Furthermore, this avoids placing popular or gregarious pages at the top of our candidate list—at the expense, however, of storing more data in memory.

When balancing the tradeoffs between getting "hint" information and incurring into additional costs (such as more memory use), a very clear ultimate consideration concerns recognizing that we will only have *partial information* available. In that sense, attempting to have all candidate information at hand not only degenerates into a brute-force choice that contradicts the approach of local search, but conceals as well the benefits of using an approximation. Logically, this approximation is not expected to act as an oracle either, as it is fond to misleadings because it does not have a complete view of the environment. Regarding this point, it seems important to mention that our heuristic may not always select the best candidate at a given step (especially when the number of prospect links of a candidate is much smaller than its neighborhood). For such reason, it must be evaluated.

To assess the efficiency of the proposed heuristic, three possible candidate orderings were considered:

- ◇ Descending
- ◇ Ascending
- ◇ No order

Table 4.13: Candidate ordering.

Quantumpedia

	Cohesion			**Coverage**	**# of clusters**	**Time** (hh:mm:ss)
	Min.	**Max.**	**Avg.**			
Descending	0.167	1.0	0.9	97.99%	330	00:02:24
Ascending	0.13	1.0	0.9	68.36	2,498	03:57:12
No order	0.34	1.0	0.93	98.68	242	00:02:28

Micropedia

	Cohesion			**Coverage**	**# of clusters**	**Time** (hh:mm:ss)
	Min.	**Max.**	**Avg.**			
Descending	0.125	1.0	0.613	81.46%	9,091	01:33:36
Ascending	*DISCARDED*					
No order	0.167	1.0	0.76	95.41%	2,803	04:48:08

As we can see from these results (Table 4.13), the descending heuristic achieves quality at a low cost.

A phenomenon of consideration that arises when sorting candidates according to this criterion is that, since the optimal solution is not guaranteed at each step of the construction process, the algorithm may take another construction path—not necessarily of lower density (that is, it may actually find a better solution); nevertheless, this behavior may also cause the algorithm to take more time before it finds out that it cannot add any more members to the cluster (which would have not happened by taking the best member at each iteration). Ultimately, this is a risk also of using FCHC. Moreover, judging by the results obtained from exploratory experiments, we see that generally this is not the case.

A pattern not necessarily related to this feature is given by observing the behavior of the clustering with our sub-collections. For instance, let us note that in a small scale, it is hard to see resource savings with the heuristic, since both orderings (no order and descending) yield similar results while spending more or less the same time. However, when scaling to a larger sub-corpus, this difference becomes much more notorious. This leads us to find out that, with a "happy days" corpus (where communities are very well identified), enhancements become superfluous to some extent, but when scaling is required, such enhancements turn into essential material.

Removal strategy

Seeking for the maximum density does not only include adding elements; it could involve element *removal* as well. As a result, we are not solely concerned with ensuring a local maximum by saturating element addition into a cluster, but also by saturating element subtraction. For this reason, a mechanism for removing vertices from the cluster in a systematical way has been implemented.

Before stating the criterion for deciding which elements to remove, it might be convenient to previously analyze the behavior of the construction algorithm with regard to several aspects. The first aspect consists of a *decreasing easiness of improvement.* If we consider that our starting point is a one-element seed, then the initial density is 0 and *any* chosen neighbor improves cohesion at the first step of the algorithm; then, as construction progresses and more steps (iterations) are given, the current density is more difficult to improve. What is more, we may even note that changes in density become smaller with each iteration on a usual basis. The former leads us to consider that perhaps the first chosen elements would have been less attractive if explored on a later step. The second aspect concerns *straying from the initial path while seeking for a local optimum.* For the sake of visualization, let us consider that our search is not uphill, but downhill; in that sense, we are placed on a given point of the search space with our seed and each local maximum exerts an "attraction force" according to its deepness. Therefore, it may happen that we start going in the direction of a certain local maximum while suddenly "stumbling" upon a deeper spot; since the construction function aims to maximize density, it will leave the initial path and go towards the one that seems to have a higher cohesion (not to mention if this path leads to or is near to the global optimum, which has $\rho = 1.0$). The former comes as a direct consequence of neighborhood expansion as the algorithm progresses; in that sense, since more neighbors can be added for consideration at each step, the construction process has a "wider view" of the search space and can opt to move to a higher local maximum instead of keeping climbing at the current peak it is exploring.

Both discussed aspects raise the possibility of eliminating the aforementioned elements if, *at the end, they show to be "weak" by making the cluster less cohesive than it could be without them.* As the reader may infer, the previous statement needs to be quantified, and quantified in such a way that we can assure that removal of these elements will result in a higher relative density (and not the opposite, of course).

To determine exactly which elements to remove from the cluster, we can start by assuming that each element contributes to the current number of internal and external links at a certain extent. Then, low contribution elements are the ones that contribute more to the number of external links, or *contribute less (than the rest) to the current number of internal links*; this last notion could be interpreted as a *marginal contribution.*

In order to model this contribution metric, an important concept to consider, undoubtedly, is the total amount of links that are internal because of the presence of a certain node in the cluster; let us relate this link set with the function $\ell(v, \mathcal{C})$ (Eq. 4.3). Now, the ratio between the number of these links and the degree of a node, can be used to represent its *absolute contribution* (Eq. 4.4); to make this proportion relative, ρ could be used for normalization. Our empirically modeled formula for relative contribution is shown in Eq. 4.5. This formula can be embedded into a removal function that refines a cluster by deleting nodes with a contribution less than a given threshold (Algorithm 4). We have found, throughout a series of trials, that this threshold corresponds to 1.0.

$$\ell(v, \mathcal{C}) = \{(u, v) \vee (v, u) : u \wedge v \in \mathcal{C}\} \tag{4.3}$$

$$c_{\text{abs}}(v, \mathcal{C}) = \frac{|\ell(v, \mathcal{C})|}{deg_{\text{out}}(v)} \tag{4.4}$$

$$c(v, \mathcal{C}) = \frac{c_{\text{abs}}(v, \mathcal{C})}{\rho(\mathcal{C})} \tag{4.5}$$

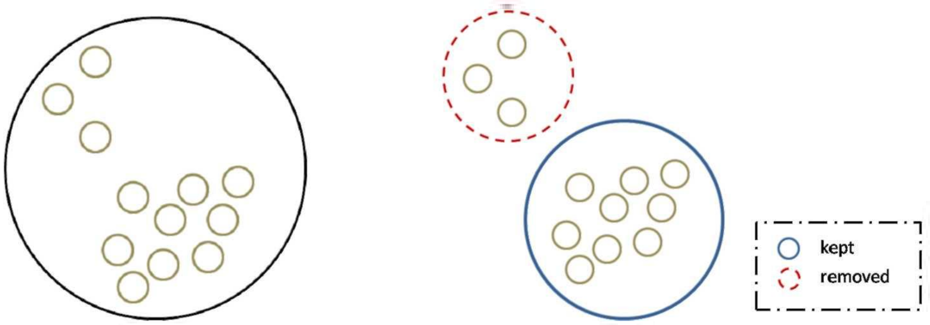
Algorithm 4 Element removal

Description: Receives a cluster $\mathcal{C}$ and returns a refined version of this same cluster $\mathcal{C}_r$, in which low-contribution elements are no longer present. This function is composed of two basic cycles. The innermost cycle searches for elements below the contribution threshold (0.1) and adds them into a removal set R; afterwards, R is subtracted from $\mathcal{C}$. The outermost cycle repeats this procedure until one of two conditions is met: either the cluster becomes "stable" (all elements are above the threshold, thus R maintains itself empty) or there are not enough elements (less than 3) to continue.

```
function REMOVAL($\mathcal{C}$)

    repeat                                        ▷ Outermost cycle (removal step)
        $R \leftarrow \emptyset$

        for all $v \in \mathcal{C}$ do            ▷ Innermost cycle (weak element search)
            $c_v \leftarrow c(v, \mathcal{C})$
            if $c_v < 1.0$ then                   ▷ Low-contribution element
                add $v$ to $R$
            end if
        end for

        remove $R$ from $\mathcal{C}$
    until $(R = \emptyset) \vee (|\mathcal{C}| < 3)$

    return $\mathcal{C}$                          ▷ Return refined cluster
end function
```

As we will see more clearly when the tuned algorithm is presented, the removal process is carried out *after* the cluster has been constructed, i.e., after we finish adding all possible elements. Such choice obeys several motives. On one hand, if we wanted to alternate addition and deletion, then we would first need to establish when this "switch" between operations should take place (e.g., after a k number of steps, at random, when a certain amount of change has been detected, etc.). Then, we have the issue of deciding whether this choice is adequate or not. As a matter of fact, on our first trials, where we attempted to alternate removal with addition, on several occasions the clusters would get completely undone—which only meant for us that our switching criterion was probably not appropriate. Furthermore, because our removal procedure is based on contribution, and the contribution of each node is modified at each addition step, it could be the case for nodes that seem to contribute little (e.g., nodes with a high degree whose links are mostly external at the beginning) end up being removed. Following this line, if our procedure would be designed not to add these nodes again on subsequent steps, we have the chance of missing nodes that could have made our cluster denser; if the design allowed for these nodes to be added again, we are performing more operations than necessary (there seems to be no case on removing a node just to add it again a few steps later). However, we also realize that doing removal at the very end has tradeoffs as well; probably the most important one is that, doing deletion operations while also adding elements could help to achieve a higher density at the end (just as natural pruning helps a plant to grow better).

Semantically, the removal process (illustrated in Figure 4.14) constitutes eliminating "off-topic" nodes (outliers)—broadly speaking, this implies the *preservation of the most dominant theme* in a multi-topic cluster. As a result, this strategy partly addresses the topic drift issue [59], which has been reported as conflictive for several algorithms, like HITS [123].



(a) Before removal (b) After removal

Figure 4.14: Removal strategy

To have a clearer picture of the removal process, let us present several actual clusters from Wikipedia sub-collections (Table 4.15); as we can see, the left column presents the refined clusters and the right column shows the set of removed elements. For the first cluster the removed articles belong more to the field of computer science and seem not to be very related to each other. In the second and third examples, we can see that elements left out belong to the same thematic, but not necessarily to the community represented by the cluster. For instance, we have an article of video gaming, but it is not about the famous Mega Man game; similarly, "The top 5 reasons you can't blame" is indeed a TV program about sports (football mostly), but actually has little to do inside a community of the Super Bowl. Finally, in some cases we can actually see more clearly the removed elements as outliers of the cluster; such is the case of an alchemy article inside a video game (which looks more like a tangential link [4]). From these examples we can, as a matter of fact, draw some intuitive traits for the kind of elements removed:

- Can be topically related but are not actually part of the discovered community
- Represent a tangential relationship
- Can either be related or unrelated among themselves
- Can be outliers

Table 4.14: Removal example.

Iteration 1: $\mathcal{C} = \{$nicole, australia, koala, sydney$\}$, $\rho(\mathcal{C}) = 0.5$				
v	deg_{int}	deg_{ext}	deg	$c(v, \mathcal{C})$
nicole	**1**	**5**	**0.2**	**0.4**
australia	3	2	1.5	3
koala	1	0	∞	-
sydney	2	1	2	4
Iteration 2: $\mathcal{C} = \{$nicole, australia, koala, sydney$\}$, $\rho(\mathcal{C}) = 1.0$				
australia	2	2	1	1
koala	1	0	∞	-
sydney	2	1	2	2

To assess the effectiveness of the removal procedure, several sets of experiments were carried out; these were executed both at the clustering (by grouping the sub-collections) and construction levels (by extracting isolated clusters), and were designed with the aim of supporting three hypotheses:

1. Removal of elements below the contribution threshold only increases density
2. Removal of other elements decreases density
3. In general, removal of "low" contribution elements increases quality

Let us now describe with more detail, as well, the designed experiments:

- Construction level

"Positive removal".- Deleting elements with contribution below the threshold (1.0) to observe if density increases (correctness)

Table 4.15: Examples of refined clusters.

Cluster	**Removed elements**
Statistics Probability Theory Bayes theorem Bayesian probability Random variable Expected value ⋮	Artificial Intelligence Zipf's law Data clustering Voice tag Generative Topographic Mapping Random forest ⋮
Super Bowl II Super Bowl III Super Bowl IV Super Bowl XXIX Super Bowl XXX ⋮	The Top 5 Reasons You Can't Blame Dan Marino Brett Favre ⋮
Mega Man (NES) Mega Man X4 Mega Man Legends Mega Man Soccer Mega Man 2 ⋮	1999 in video gaming
Castlevania: Lament of Innocence Castlevania: Simon's Quest Castlevania: Harmony of Dissonance Castlevania: Dracula's Curse ⋮	Philosopher's Stone

"Negative removal".- Deleting elements with contribution above the threshold to observe if density decreases (completeness)

- Clustering level

  Result comparison.- Contrasting results obtained by clustering without element removal against results with removal; clustering being carried out over the subcollections.

From the previous list, it is clear to see that each experiment is aligned with one of the hypotheses; in that sense, positive removal aims to support hypothesis 1, negative removal, hypothesis 2, and result comparison, hypothesis 3.

Positive and negative removal experiments were carried out by randomly constructing 10,000 clusters (whose relative density, among other data, was recorded before and after the removal procedure taking place). As we can see from Table 4.16, the main finding with respect to positive removal is that there were no cases where element deletion would result in

Table 4.16: Removal results.

Positive removal

	% of RD increase	Nodes removed		Links removed	
		#	%	#	%
Min.	0.00%	0	0.00%	0	0.00%
Max.	356.00%	515	86.5%	9407	88.2%
Avg.	8.9%	7.2	11.9%	60.1	10.9%

Negative removal

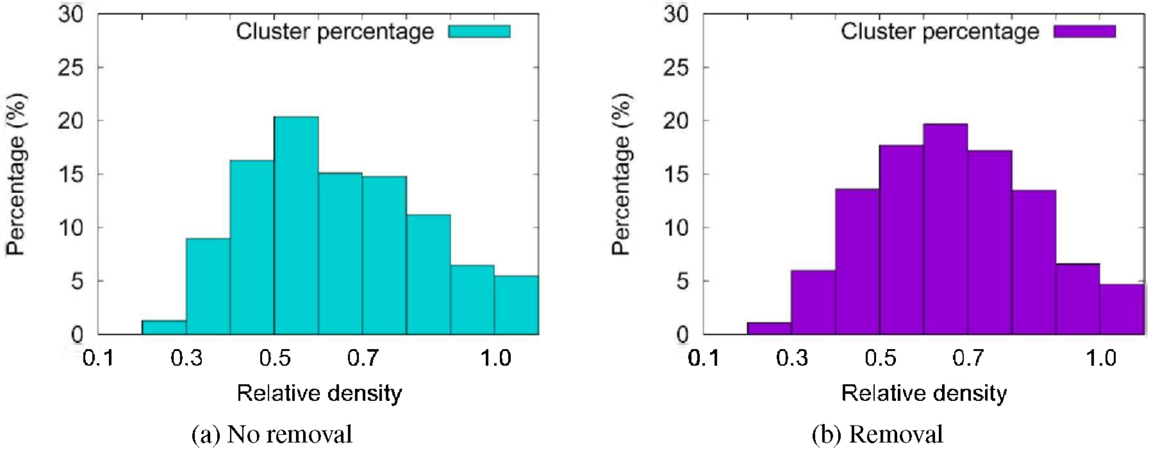
	% of RD increase	Nodes removed		Links removed	
		#	%	#	%
Min.	-100%	1	0.01%	1	0.0003%
Max.	80%	567	98.6%	47989	100.00%
Avg.	-29.8%	136.5	38.1%	7475.5	50.2%

a decrease of relative density; on the contrary, density only showed to increase with removal the of low contribution elements (in cases where there were none of these, density, of course, remained the same). On the other hand, while the increase in density was on average modest ($\approx$ 10%), there were instances in which the initial density multiplied several times. Consequently, these results support our first hypothesis.

Regarding negative removal, results from the same table allow to observe that removal of elements above the contribution threshold does not always yield a decrease in density. By proving failure in completeness (and consequent rejection of our second hypothesis), this finding challenges the overall effectiveness of the contribution formula, at least apparently; however, after revising each individual case, we gained a clearer insight for such outcomes. On the first place, we discovered that, on each run, less than 1% of the instances obtained improvement with negative deletion (that is, an average of five cases showed this behavior); furthermore, when analyzing the properties of these instances, we observed that a *small size* (typically less than fifteen elements in the cluster) was a trait shared by them all. Even when these findings do not dismiss the fact that the formula requires tuning to achieve completeness, it remains clear that this behavior is not dominant, while acknowledging that cluster size also becomes important for the sensitiveness of the removal strategy.

Concerning this last aspect, a key question as well is comprised by *convergence*; since the algorithm is executed until no more elements below the threshold are found, it could happen that the cluster becomes empty before complying with such condition. In that sense, the first runs of positive and negative removal ratified this fact by showing that all elements could become deleted, specially when having two elements left, because contribution for one of them will invariably be less than 1.0, as one will tend to contribute more than the other. Although this behavior is not dominant either (happened in approximately 6% of the cases), it demands modifications to the removal strategy (as a matter of fact, Algorithm 4 already includes this modification in Line 11, where removal is terminated when having a small amount of vertices left).

With respect to result comparison, Figure 4.15 shows density histograms for Micropedia with and without removal (Figures 4.14 and 4.15a, respectively). As we can observe from the illustration, clusters with high cohesion $(0.5, 0.6, 0.7, \ldots)$ are more common when applying the removal strategy; furthermore, density on average is higher. Therefore, by accounting these results, the third hypothesis can be considered as proved. Nevertheless, this quality improvement unfortunately is at the cost of document coverage, which showed to decrease 10% on average; the former takes place because the removal strategy does not reassign deleted elements to other clusters (it is being assumed that they will eventually appear into other clusters where they become more strongly attached).



(a) No removal

(b) Removal

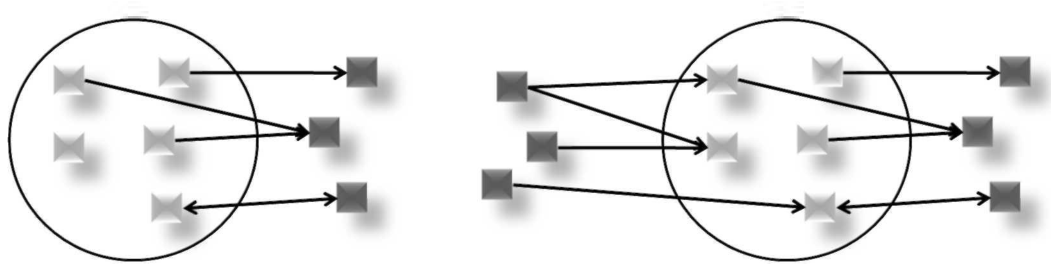
Figure 4.15: Removal strategy evaluation.

This last drawback and the other disadvantages found with our exploratory experiments lead us to summarize and discuss the removal strategy's *limitations*. Besides the ones just mentioned, another important issue one concerns size-sensitiveness; in that sense, the removal algorithm performs better with clusters that have twenty or more elements. To overcome some of these limitations, we could, for example, enforce the algorithm to stop when reaching a size equal to two elements, and this would guarantee convergence. Also, we could employ several alternatives (like creating a "rest" cluster that contains erased elements) to improve coverage.

To close the current section, we can state—on the other hand—that the removal strategy so far has showed to be *correct* and to improve quality at the clustering level; above all, element elimination helps to ensure maximum-density clusters, helps to create more *introvert* structures, and does not present high demands of time and memory. As a result, removal is to be considered as an integral part of topic construction, and thus is to be finally included.

Neighborhoods

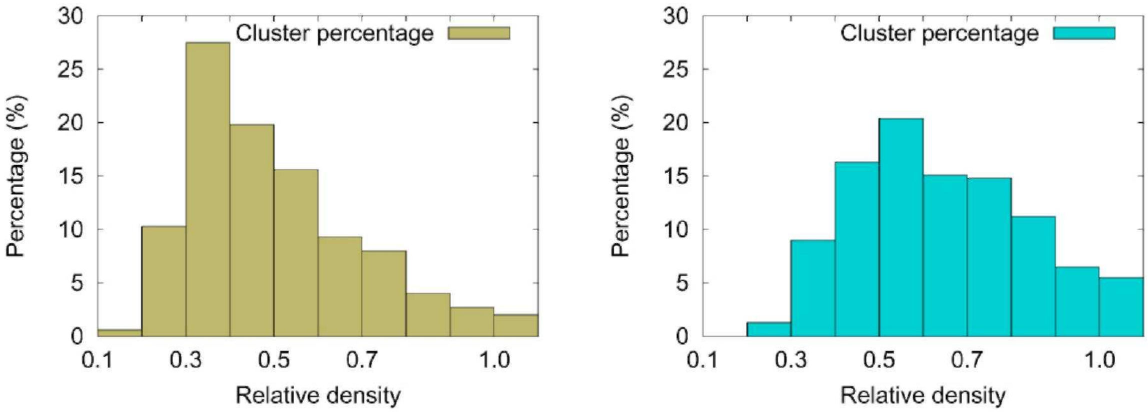
Another issue that arises is whether to consider as candidates all the neighbors of a given node (*complete neighborhood*) or only those that the node points to (*partial neighborhood*), as shown in Figures 4.16b and 4.16a, respectively. On one hand, the first option is the one that seems most reasonable, but partial neighborhoods seem logical as well if we recognize that a Web page by itself visibly contains only links (additionally, this option was utilized for preliminary experiments, whose results showed to be cohesive).



(a) Partial (b) Complete

Figure 4.16: Neighborhood views

To observe more precisely how neighborhood type affects density, both options were tested with the sub-collections. From Figure 4.17, we can derive that clusters with higher cohesion are obtained by using complete neighborhoods; in fact, improvement is highly noticeable. In that sense, the choice for complete neighborhoods not only gains advantage because it enables a wider view for the construction function, but also for this empirical evidence.



(a) One direction (partial) (b) Two directions (complete)

Figure 4.17: Neighborhood evaluation.

Nevertheless, at a large scale (whole Wikipedia), the employment of complete neighborhoods incurs in elevated costs regarding time, thus showing to drop effectiveness considerably. Actually, the problem at this scale is related to a well-known search issue: *very large neighborhoods* (see the survey by Ahuja et al., where this problem and several alternatives for overcoming it are discussed with respect to the TSP problem [2]). So, with our sub-collections, the previously mentioned issue is not easily detected (although for Micropedia, execution time jumped from ten minutes to almost an hour) or represent serious performance degradation, but when clustering all of Wikipedia, the problem is immediately visible, as a considerable number of nodes has an extensive amount of neighbors. Therefore, it seems preferable to avoid using complete neighborhoods.

Seed expansion

Yet another design choice that has to be made is whether or not to expand (one-element) seeds, and how to carry out the expansion if this is to be done. The reason behind the consideration of this enhancement concerns, basically, assuming that a multi-element seed will result in a more *focused* cluster; for instance, a seed article of "Artificial Intelligence" by itself may eventually "evolve" into a computer science cluster or a mathematics topic, but a set consisting of several field-related articles (e.g. {"AI", "Machine Learning", "Computer Intelligence", "NLP", ...}) could result into a group that appears to be more consistent with the initial input.

Two suitable candidates for expansion are mutual links and structural similarity. Regarding the first, it consists of incorporating those nodes that simultaneously link to and are linked by the seed; the latter corresponds to selecting a given number of neighbors that show to have a similarity above a given threshold.

As for mutual links, these constitute an intuitive option, considering that a bidirectional relation implies a stronger tie between two documents and indicates that they probably belong to the same topic. For example, if we take the "United States" article, which is linked by a considerable amount of documents, it seems logical to assume that the core for a topic referring to this country will be made up of those articles that "USA" links back.

Structural similarity has already been discussed [149, 108], since it concerns a pairwise measurement for finding meaningful groups based on links; it has traditionally been utilized for building cluster "cores", and for this motive it regards a reasonable technique for expanding seeds (in fact, for a simple case with a small number of nodes and connections, it could even be used as a "cheap" clustering process). Unfortunately, the use of structural similarity incurs into an additional cost, as it requires the inclusion of two extra parameters: a similarity threshold and the maximum number of neighbors to choose (namely, μ and ϵ).

Because structural similarity requires to test combinations of μ and ϵ, it is not trivial to assess its effectiveness exhaustively. In that sense, exploration was carried out by setting $\mu = 0.1$ and $\epsilon = 10$ and 20; results on the sub-collections show an improved average density, although the improvement is not very significant ($2 \sim 3\%$). On the same side, with the intent of overriding complexity, the number of neighbors was relaxed (thus allowing *any* amount of neighbors equal to or exceeding μ); another reason for dropping this parameter was to balance the number of neighbors in proportion to the amount of links. Nevertheless, observations drawn from these experiments uncovered a potential risk of seed expansion in general: a fast growth in neighborhood size. Another risk for expansion, regardless of the approach, is that sometimes the elements added for expansion anyways end as deleted by the removal strategy. In addition, for structural similarity (besides parameter tuning), neighbor enumeration has high costs for documents with many links.

Therefore, this enhancement—at least in appearance—does not seem convenient, since it introduces overhead and improvements have shown, so far, to be of little significance.

Secondary cluster

By removing elements from the final cluster and providing a wider view of neighborhoods, we gain quality, but at the same time, we are more prone to generating redundant clusters (even when the clustering mechanism is originally designed to try to avoid this problem). Such situation arises with topic digressions; as it was discussed previously, when the neighborhood increases (either by having a wider view since the beginning or because with iterations it has grown to a considerable extent) it is possible to have a topic drift. Furthermore, we have also seen that it is not uncommon for this shift to be made to a *more dominant topic*. Being this the case, the algorithm is most likely to remove the documents belonging to the less dominant topic(s); because documents are never blocked from appearing into other clusters, it seems quite logical to assume that leaving cluster removed elements in the seed list (which, in fact, means doing nothing to handle the eliminated set) increases their probability of eventually being collected into other groups. Nevertheless, the former intuition has showed to cause the aforesaid higher degree of redundancy, for the simple reason that seeds consisting of elements previously from another cluster tend to "fall" on the same topic, consequently generating an identical (or almost identical) cluster from which they were removed in the first place. And the same situation is repeated over and over again for the rest of the removed elements... not only from one but from a considerable number of clusters. The main problem here, however, is not redundancy *per sé*, but the amount of time that is wasted by creating repeated clusters.

As the reader can infer, the solution to this observed issue lies in the elimination of removed elements from the seed list as well; in that way, we can avoid duplicities or very similar clusters. Nonetheless, an interesting question that arises from this consideration is whether this set of elements has by itself some kind of meaning; if it does, then it seems reasonable to consider it as part of the clustering, thus conforming a "rest" or "secondary" cluster derived from the discovery of a more dominant topic.

To test the effects of secondary clusters, two kinds of experiments were carried out around this feature. The first type was devoted to observing if the deletion of these clusters from the seed list was in fact helpful for reducing redundancy; the former was assessed by clustering our sub-collections with/without eliminating secondary clusters and measuring the number of created clusters and redundancy (while considering that quality or coverage were not degraded either). The second type consisted of evaluating the preservation of secondary clusters into the grouping, and was carried out by comparing the density levels with/without including secondary clusters into the final clustering (let us note that on these other experiments, secondary clusters were deleted from the seed list, regardless of whether they were kept or not).

Results for the first set of experiments are shown in Table 4.17. As we can see, the effectiveness of secondary cluster deletion from the seed list is corroborated by the diminution of redundancy, number of generated clusters, and time spent (note that for Micropedia, time was reduced almost by 50%); at the same time, quality and coverage were basically preserved (they oscillate between one clustering and the other, but not in a significant form).

Regarding the second set of experiments, Figure 4.18 illustrates density levels for Micropedia with and without considering secondary clusters. With respect to this aspect, it seems clear that, for the usual, this kind of clusters is not very cohesive (this can, consequently, be interpreted as a lack of meaning). What is more, quality is severely compromised; besides these obtained results, another reason for not incorporating this feature into the clustering function is that it actually appears as contradictory to our approach of work, because we always seek for the *maximum-cohesion* clusters. Therefore, both from theoretical and experimental points of view, secondary cluster inclusion as part of the final grouping is to be discarded.

A more severe effect of redundancy that has not yet been discussed because it does not affect the clustering itself, but rather its evaluation, is given by the creation of *deceptive*

Table 4.17: Seconday cluster experiments.

Quantumpedia

	Redundancy	# of clusters	Time	Density	Coverage
No secondary clusters	2.5	306	00:02:20	0.91	98%
Secondary clusters	1.5	174	00:01:26	0.95	97.5%

Micropedia

	Redundancy	# of clusters	Time	Density	Coverage
No secondary clusters	17.6	8,761	01:30:14	0.56	82%
Secondary clusters	8.9	5,060	00:45:24	0.57	80%

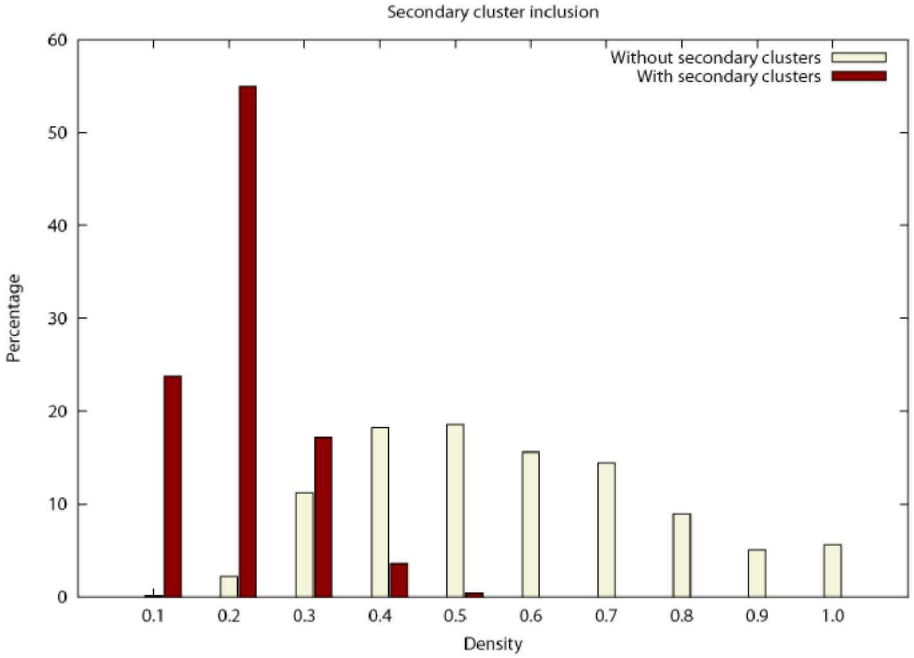


Figure 4.18: Cohesion (density) with secondary clusters.

results. On one side, if the most part of the repeated clusters are dominant-theme groups, and we know that these groups tend to have a high relative density, then it is very likely to have "inflated" quality scores (while the opposite can also take place, this situation is more rare)—especially if we do not handle redundancy. On the other side, "twin" cluster tracking is not a trivial task, since it requires pairwise comparisons (either with the complete set of documents or just a part) for knowing *exactly* which groups are repeated. The former concerns an additional motivation for secondary cluster management, and also serves to remark that the results of our exploratory experiments have been "normalized" by taking out of consideration those groups that appear to be repeated. To accomplish the detection of this kind of groups in a quickly, approximate fashion, a *cluster signature* (composed of the relative density, number of internal links, and number of external links) is currently employed.

So, finally, we can see that secondary cluster consideration is useful for reducing redundancy. However, the inclusion of these clusters in our constructed groups deteriorates quality; an intermediate—presumably more intelligent—option is to add to the clustering only those secondary groups that show to be cohesive. A final remark regarding this feature concerns "deeper level" clusters; as elements can also be eliminated from this secondary cluster, this opens the possibility to even have a "tertiary" cluster, and so on. Considering that the preservation and analysis of clusters at such depths could recur more into overhead than into an actual benefit (and more since our primary reason for creating secondary clusters is not to find less dominant topics but to shorten the seed list and save time), let us note that no further clusters will be obtained from the secondary one. The final removal function with secondary cluster recovery is given by Algorithm 5

Algorithm 5 Element removal with secondary cluster recovery

Description: The procedure is almost identical to conventional removal, but removed elements (R) are added to the secondary cluster ($\mathcal{C}'$) at each step. In addition, instead of returning a single (refined) cluster $\mathcal{C}$, a *cluster duple* $D_\mathcal{C}$ is produced as output.

```
function REMOVAL($\mathcal{C}$)

    $\mathcal{C}' \leftarrow \emptyset$
    repeat                                          ▷ Outermost cycle (removal step)
        $R \leftarrow \emptyset$

        for all $v \in \mathcal{C}$ do                         ▷ Innermost cycle (weak element search)
            $c_v \leftarrow c(v, \mathcal{C})$
            if $c_v < 1.0$ then                              ▷ Low-contribution element
                add $v$ to $R$
            end if
        end for

        remove $R$ from $\mathcal{C}$
        add $R$ to $\mathcal{C}'$
    until $(R = \emptyset) \vee (|\mathcal{C}| < 3)$

    $D_\mathcal{C} \leftarrow (\mathcal{C}, \mathcal{C}')$
    return $D_\mathcal{C}$                                   ▷ Return a cluster duple
end function
```

Cluster job scheduling: seed ordering and quantum

With the dynamic seed list, an obvious question that emerges is whether the order in which seed documents are treated affects results. For example, if we take seeds by ascending degree, do we have a larger amount of smaller clusters than if we take them by descending order? Or, is it different to take seeds by ascending or descending id order than to take them at random?

With this enhancement, the main interest is to know if the order in which the seeds are taken to conform the clusters affects our results.

Therefore, we tried six different options for sorting the input seeds:

- By in-degree
  - Ascending and descending
- By out-degree
  - Ascending and descending
- By degree (total)
  - Ascending and descending

Table 4.18: Seed ordering

Quantumpedia

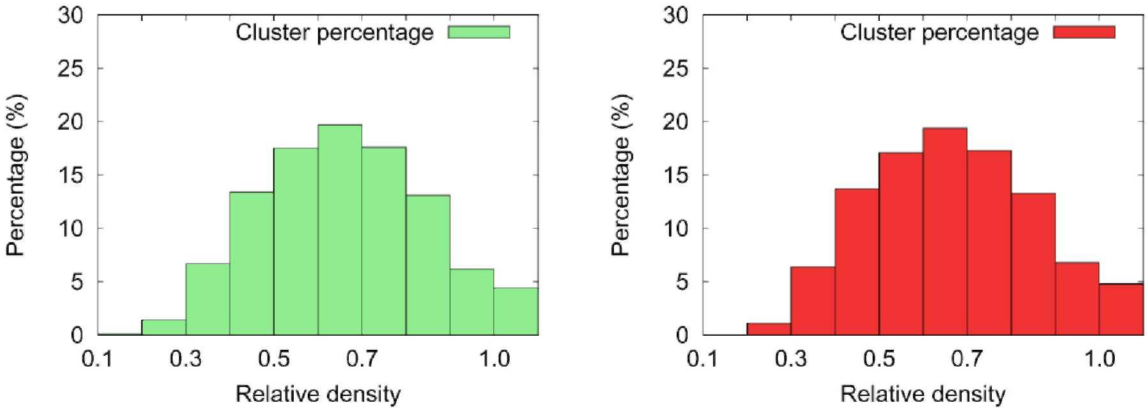
	Cohesion			**Coverage**	**# of clusters**	**Time (hh:mm:ss)**
	Min.	**Max.**	**Avg.**			
No order	0.167	1.0	0.9	97.99%	330	00:02:24
In asc.	0.167	1.0	0.9	97.4%	208	00:01:26
Out asc.	0.167	1.0	0.895	97.98%	318	00:01:52
Total asc.	0.167	1.0	0.892	97.99%	305	00:02:08
In desc.	0.167	1.0	0.911	97.46%	231	00:01:48
Out desc.	0.167	1.0	0.917	98.01%	306	00:02:20
Total desc.	0.167	1.0	0.915	98.01%	321	00:01:58

Micropedia

	Cohesion			**Coverage**	**# of clusters**	**Time (hh:mm:ss)**
	Min.	**Max.**	**Avg.**			
No order	0.125	1.0	0.613	81.5%	9,091	01:33:36
In asc.	0.125	1.0	0.607	81.7%	6,538	01:09:06
Out asc.	0.125	1.0	0.602	81.1%	9,321	01:34:11
Total asc.	0.125	1.0	0.604	81.7%	8,921	01:32:43
In desc.	0.125	1.0	0.612	81.4%	7,297	01:13:27
Out desc.	0.125	1.0	0.621	81.7%	8,761	01:30:14
Total desc.	0.125	1.0	0.618	81.7%	9,135	01:32:50

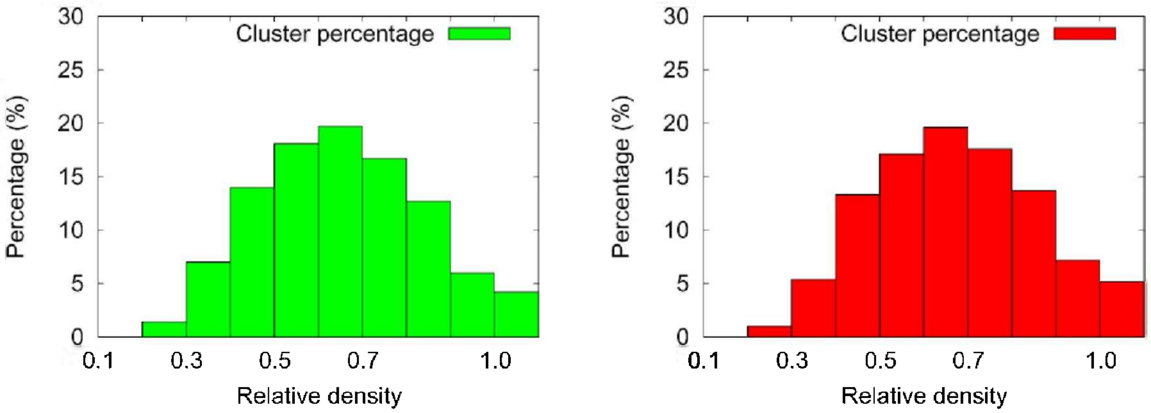
Drawing conclusions from these experiments, descending options seem to achieve a higher quality, while in-degree usually tends to spare the least time, however producing less coverage (which is logical due to not every page having other pages that link to it, therefore we have less seeds to treat and this can derive finally in less clusters, so less time). Nevertheless, the bottom line here is that these options attain *a similar behavior*.

Even when anyone of these six options seems suitable for seed ordering, an issue that is left open is whether results keep the same similarity when scaling to the Wikipedia level.



(a) Ascending (b) Descending

Figure 4.19: In-degree

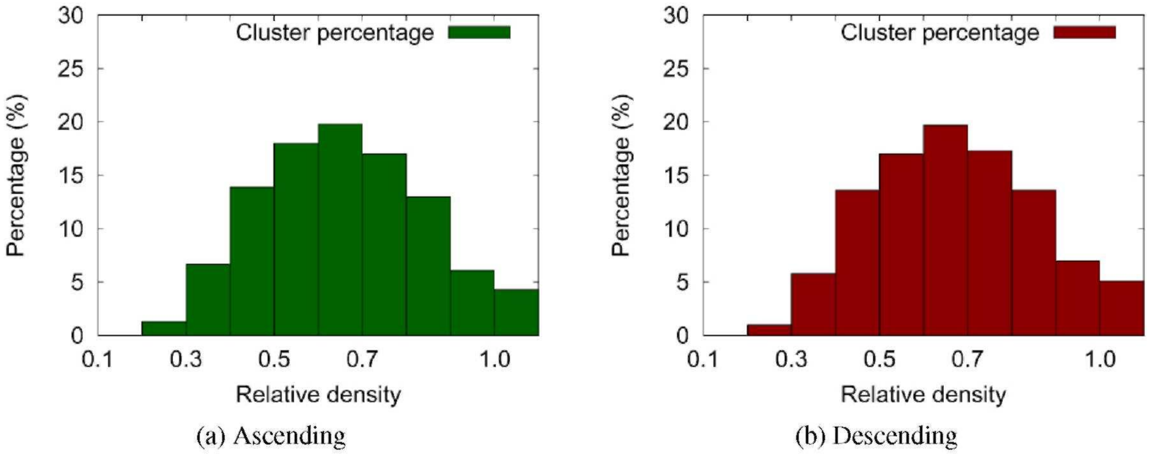


(a) Ascending (b) Descending

Figure 4.20: Out-degree

Furthermore, a potential risk is that sorting by degree leaves (not on purpose) topically similar nodes close to each other; this, in turn, implies that if a cluster does not gather all the nodes of the topic at once—which is very prone to happen if we apply a time limit, for instance—, we might be exploring the same space over and over again. This same phenomenon can be produced by taking seeds with regard to their id order, as related documents are usually contiguous to each other. Consequently, for the sake of coverage and time restrictions, the most secure option seems to be given by a *random seed selection*.

An additional consideration is quantum inclusion; to start with, the term "quantum" belongs—in this case—to the operating systems jargon. It is employed when referring to *round robin* [110], which is a type of scheduling where every process is granted a (usually short) amount of time (known as *quantum*) for execution. After the given time is over, the process is put "on hold" if it is not finished and the next process on the *queue* is put to work, with the same quantum; once a round of processes has concluded, the quantum is awarded again to those that were left incomplete. A scheduling scheme of this kind is ordinarily used



(a) Ascending

(b) Descending

Figure 4.21: Degree

to prevent *process starvation*—that is, a process having to wait an "unbearable" period of time for being carried out (generally because a longer process is taking place).

As we can see, these scheduling concepts can be aligned to our clustering task (especially at a large scale). On one hand, the construction of a cluster can be seen as a process to be completed (obviously, the execution time is not known beforehand), and the seed list resembles the process queue. Similarly, process starvation in our context can be understood as clusters that do not get to be constructed because, in realistic terms, there is no time left—time which was spared in the completion of longer-period-consuming clusters. This event (illustrated in Figure 4.22, where each box represents a different cluster) is unfortunate not only because it undermines coverage, but also because quality can be affected to some extent (particularly when there were high-density, short jobs coming ahead). Another factor that motivates quantum inclusion is that time-consuming clusters, for the usual, look like having a "long tail"; with each iteration, improvement is less and time spared to find the candidate that yields it is more. This is equivalent to stating that these clusters gather a considerable density within the first iterations and this quantity is improved "little by little" on the last ones. Therefore, it seems reasonable (acknowledging time limitations) to interrupt construction after a certain period of time—assuming thereby that the obtained density during this period is not going to be very low. Because an interruption suggests no turning back, another key aspect is whether or not it seems feasible to execute clustering rounds; although this alternative might actually lead to the eventual completion of some cluster jobs, seeing that another set of these jobs could take a more than considerable period for execution, rounds do not seem realizable. As a result, quantum inclusion can be thought more of as the time limit parameter used in SA.

Quantum involves, just as the rest of the enhancements, several issues as well. On one hand, if it is exceeded, the resulting cluster gets *truncated*, since element addition is not concluded (note that the time limit, therefore, is only set for inclusion, as it is the most time-consuming operation). Going deeper into this detail, it can be stated that the interruption of a single group actually affects the whole clustering—even when multiple rounds were feasible; the former happens because elements that could have been added to the cluster with more time

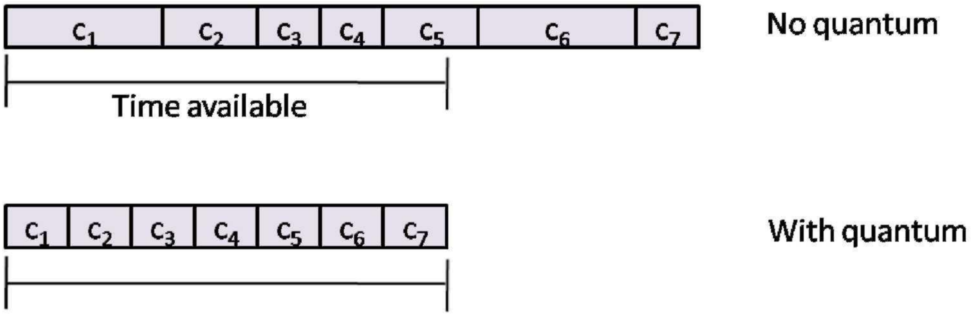


Figure 4.22: Quantum utilization for clustering.

available are to still remain in seed list and, thus, be processed. Of course, this raises the question of whether quantum does really improve quality or degrades it. However, for satisfying time limitations (more with the current resources), this risk has to be taken. Another complication with quantum is that it acts as a *large scale* feature; the previous trait implies that, with our current sub-collections, it simply is not applicable (all clusters are constructed so quickly that a quantum time is unfit). Consequently, it is hard to quantify its effectiveness, unless it is tested with a larger sub-corpus or the complete collection. Because of this particularity, it shall still be discussed on the next chapter. However, the essential conclusion regarding this enhancement still holds: in despite of its disadvantages, at the Wikipedia level, doing without the quantum seems practically unrealizable.

Because we have observed that at each iteration the increase in density is smaller, an alternative to quantum use consists of measuring the change in density from one iteration to the other and stopping when this change reaches a certain threshold. A variant of this last possibility, perhaps also more flexible than a static quantum, is to grant more time—without interrupting the process—to a cluster whose construction shows to be making a constant progress (and therefore could be negatively affected by a rigid truncation)[10].

Seed ordering and quantum inclusion can be seen as pertaining to the same concept: *cluster scheduling*. In that sense, they could be treated as equivalent or complementary to each other—under certain circumstances.

Tailored algorithms

Table 4.19 summarizes the values chosen for each enhancement parameter. These details are depicted in Algorithms 6 (auxiliary functions correspond to Algorithms 7, 8, and 5) and 9.

Algorithmic complexity

One last consideration concerns *algorithmic complexity*. For our case, the basic operation is the addition of a new element to a cluster for increasing relative density. To carry out this addition, we have to iterate over the candidates that make up the neighborhood to find the first one that improves the current density, and this is done as many times as we attempt to

[10] This criterion would be similar to the one used by several traffic light systems, which keep the green light for a longer time if a considerable number of vehicles is crossing that intersection.

Table 4.19: Enhancement values.

Feature	**Value**
Candidate ordering	Sorted in descending order according to number of prospective internal links.
Neighborhood choice	First choice (first candidate that improves current density).
Element removal	Included; applied while having at least three elements left.
Neighborhood type	Partial.
Seed selection	At random.
Seed expansion	Not included. Seeds consist of just one element.
Secondary clusters	Included; secondary clusters are erased from the dynamic seed list, but kept only if they show to be cohesive.
Quantum	Included; quantum value is variable, although estimated to last only several minutes.

increase the size of the cluster. Ultimately, the operation is executed for every seed of the seed list that is processed. In that sense, we have three nested cycles:

1. Basic cycle that searches for addition: iterates over the neighborhood to find a fit candidate (bounded by neighborhood size)

2. Intermediate cycle that carries out the search every time we attempt to include a new element into the current cluster (bounded by cluster size)

3. Outermost cycle that creates a cluster for every seed of the list (bounded by seed list size)

If each loop is taken individually (i.e., without taking the other ones into account), we can see that the execution is done n times at the most—recalling that $n = |V|$. This yields, roughly speaking, a worst case complexity of $O(n^3)$; however, this worst case can be considered as rare, mainly because the approach works locally. As a consequence, the usual number of iterations per cycle is actually much smaller than n. Also, because the loops are not totally independent from each other, it becomes rather difficult for them to reach n at a same time; for example, the seed list is shortened by taking out those elements that get clustered. In that sense, perhaps the worst scenario takes place when the graph is *unclusterable* (e.g., if it were complete and unweighted).

With respect to the average number of iterations for each cycle, according to the results we will see on Chapter 5, the average neighborhood size of a cluster $\mathcal{C}$ is $\Gamma(\mathcal{C}) = 1,083$, while the size of a cluster is $|\mathcal{C}| = 130$, and the seed list size (equivalent to the size of the final cluster set) is $|\mathbb{C}| = 300,000$. Therefore, we have evidence for actually stating that $|\Gamma(\mathcal{C})| \ll n$ and $|\mathcal{C}| \ll n$.

Algorithm 6 Detailed construction function.

Description: Receives a seed S, a time limit of *quantum* units, and a neighborhood type (partial or complete) in order to produce a cluster duple $(\mathcal{C}_i, \mathcal{C}'_i)$, where the primary cluster ($\mathcal{C}$) is generated through the *addition* of elements and the secondary cluster ($\mathcal{C}'_i$) results from the *deletion* of elements from the primary group. Addition is carried out by selecting first the cluster neighbor that improves relative density; if the search for such neighbor takes more than quantum units, the addition step terminates. For details on deletion, see Algorithm 5.

```
function CONSTRUCT-GLC-TOPIC($S$, quantum, neighborhood-type)

    UPDATE-CLUSTER($S$, neighborhood-type)
    repeat
        $\rho_{\text{curr}} \leftarrow \rho(\mathcal{C}_i)$, time $= 0$
        $L \leftarrow \{L_1 \cup \ldots L_{|\mathcal{C}_i|}\}$
        remove $\mathcal{C}$ from $L$
        UPDATE-NEIGHBOR-TABLE(T)
        SORT(T)

        $L_{\text{first-choice}} \leftarrow \emptyset$
        while (foundCandidate = false) $\vee$ (no more entries left to explore in $T$) do
            $(l,$ occurrences$)$ $\leftarrow$ next entry from table $T$
            if $\rho(\mathcal{C}_i \cup l) > \rho_{\text{curr}}$ then
                add $l$ to $L_{\text{first-choice}}$
                $\mathcal{C}_i \leftarrow$ UPDATE-CLUSTER($L_{\text{first-choice}}$, neighborhood-type)
                foundCandidate = true
            end if
            increment time by one unit
        end while
        $\rho_{\text{new}} \leftarrow \rho(\mathcal{C}_i)$
    until $(\rho_{\text{new}} = \rho_{\text{curr}}) \vee$ (time $>$ quantum)

    if $|\mathcal{C}_i| > 2$ then
        $(\mathcal{C}_i, \mathcal{C}'_i) \leftarrow$ REMOVAL($\mathcal{C}_i$)          ▷ Secondary cluster recovery
    end if

    return $(\mathcal{C}_i, \mathcal{C}'_i)$
end function
```

Algorithm 7 Cluster update

Description: This procedure is composed of two steps: 1) adding a set η of new members to the cluster, 2) updating the neighbor table with the neighborhood of these new elements. Let us note that this procedure is executed in two types of occasions: a) when the cluster is initialized with the seed, and b) when a suitable candidate for the cluster is found (this happens at each step). Although we currently employ singleton seeds and new members are added one at a time, note that the use of a set (η) allows the eventual utilization of compound seeds (which should be a transparent aspect for this method).

```
procedure UPDATE-CLUSTER($\eta$, neighborhood-type)

    for all $n \in \eta$ do                                  ▷ New member addition.
        add $n$ to $\mathcal{C}$
                                                             ▷ Cluster neighborhood update.
        if neighborhood-type = partial then
            $L_n \leftarrow \Gamma_o(n)$
        else
            $L_n \leftarrow \Gamma(n)$
        end if

        UPDATE-NEIGHBOR-TABLE($L_n$)
    end for

end procedure
```

Algorithm 8 Neighbor table update

Description: Receives an L set of neighbors (in + out *links*) and updates the neighbor table by doing, per neighbor, one of two operations: a) if the neighbor does not exist in the table, a new entry is created and the number of occurrences for such entry is set to 1, b) if the neighbor already exists in the table, its entry is modified by incrementing the number of occurrences.

```
procedure UPDATE-NEIGHBOR-TABLE($L$)

    for all $l \in L$ do
        if $l \notin \mathcal{C}$ then                    ▷ The neighbor set should be disjoint from $\mathcal{C}$

            if $l$ exists in $T$ then
                $T \leftarrow (l, \text{occurrences}_l + 1)$          ▷ Update table entry
            else
                $T \leftarrow (l, 1)$                              ▷ Add a new table entry
            end if

        end if
    end for
end procedure
```

Algorithm 9 Detailed clustering function

Description: Additional to the $\mathbb{S}$ set of seeds, the tailored clustering function receives a *time limit* of quantum units and the *neighborhood type* to use for construction. Furthermore, because we are considering secondary clusters (produced from the removal procedure at the construction function), these are subtracted from the seed set (just as the primary group) and added to the cluster *only* if their density lies above an *acceptance threshold.*

```
function CLUSTERING($\mathbb{S}$, quantum, neighborhood-type)

    repeat
        $S_i \leftarrow$ CHOOSE-RANDOM-SEED($\mathbb{S}$)                ▷ Get the next seed at random

        $(\mathcal{C}, \mathcal{C}') \leftarrow$ CONSTRUCT-TOPIC-GLC($S_i$, quantum, neighborhood-type)
        add $\mathcal{C}$ to $\mathbb{C}$

        if $\mathcal{C}' >$ acceptance threshold then
            add $\mathcal{C}'$ to $\mathbb{C}$
        end if

        remove $\mathcal{C}$ from $\mathbb{S}$
        remove $\mathcal{C}'$ from $\mathbb{S}$
    until $\mathbb{S} = \emptyset$

    return $\mathbb{C}$                                         ▷ Return set of clusters
end function
```

4.1.4 Topic construction: sub-task summary

A concise summary of the related aspects of the construction endeavor is presented in Table 4.20.

Table 4.20: Main aspects of the construction sub-task.

<table>
<tr><td colspan="2"><b>Construction.-</b> <i>Regards the development of a base mechanism for enumerating topic members; in other words, it consists of providing a process for mapping each document into one or more topics.</i></td></tr>
<tr><th>Key question</th><th>Related aspect</th></tr>
<tr><td><i>How is the mapping going to take place?</i></td><td>Mapping obeys the <i>Graph Local Clustering</i> approach. So, a document belongs to those groups where it:
a) is reachable from (in the local vicinity)
b) shows to improve cohesion</td></tr>
<tr><td><i>How is information going to be managed?</i></td><td><table><tr><th>Function</th><th>Inputs</th><th>Outputs</th></tr><tr><td>Construction</td><td>a seed $S \in \mathbb{S}$</td><td>a cluster $\mathcal{C}$</td></tr><tr><td>Clustering</td><td>a seed list $\mathbb{S}$</td><td>a set of clusters $\mathbb{C}$</td></tr></table></td></tr>
<tr><td><i>What are the parameters?</i></td><td>We can talk about two types of parameters: general and specific.

<i>General</i> parameters are set for creating a concrete GLC approach:
1) the fitness function
2) the search strategy
An additional detail (not necessarily a parameter) concerns the way of selecting and managing a seed list.

<i>Specific</i> parameters correspond to fine-tuning issues:
1) element removal consideration
2) candidate element ordering
3) neighborhood type
4) seed ordering
5) seed expansion consideration
6) secondary cluster consideration
7) time limit consideration</td></tr>
</table>

4.2 Topic description

The aim of topic description is to calculate properties for a given set of documents, such that these have a more explicit topical meaning. In that sense, we have opted to focus on two main properties: 1) the topic's most outstanding members and 2) a topic tag . These two make

4.1.4 Topic construction: sub-task summary

A concise summary of the related aspects of the construction endeavor is presented in Table 4.20.

Table 4.20: Main aspects of the construction sub-task.

<table>
<tr><td colspan="2"><b>Construction.-</b> <i>Regards the development of a base mechanism for enumerating topic members; in other words, it consists of providing a process for mapping each document into one or more topics.</i></td></tr>
<tr><td><b>Key question</b></td><td><b>Related aspect</b></td></tr>
<tr><td><i>How is the mapping going to take place?</i></td><td>Mapping obeys the <i>Graph Local Clustering</i> approach. So, a document belongs to those groups where it:
a) is reachable from (in the local vicinity)
b) shows to improve cohesion</td></tr>
<tr><td><i>How is information going to be managed?</i></td><td><table><tr><th>Function</th><th>Inputs</th><th>Outputs</th></tr><tr><td>Construction</td><td>a seed $S \in \mathbb{S}$</td><td>a cluster $\mathcal{C}$</td></tr><tr><td>Clustering</td><td>a seed list $\mathbb{S}$</td><td>a set of clusters $\mathbb{C}$</td></tr></table></td></tr>
<tr><td><i>What are the parameters?</i></td><td>We can talk about two types of parameters: general and specific.

<i>General</i> parameters are set for creating a concrete GLC approach:
1) the fitness function
2) the search strategy
An additional detail (not necessarily a parameter) concerns the way of selecting and managing a seed list.

<i>Specific</i> parameters correspond to fine-tuning issues:
1) element removal consideration
2) candidate element ordering
3) neighborhood type
4) seed ordering
5) seed expansion consideration
6) secondary cluster consideration
7) time limit consideration</td></tr>
</table>

4.2 Topic description

The aim of topic description is to calculate properties for a given set of documents, such that these have a more explicit topical meaning. In that sense, we have opted to focus on two main properties: 1) the topic's most outstanding members and 2) a topic tag . These two make the topic more manageable and not only provide a fair summary, but also serve to put the topic more in shape. Seen from another point of view, these properties can be accounted as *metadata*, or "structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage and information source" (NISO[11] definition). Yet another conception for topic description concerns the "data abstraction" step of the clustering task, as defined by Jain and Dubes [64].

Although the importance of topic description should not be underestimated at any cost, this sub-task can be considered as less critical than topic construction. Consequently, the general approach for description can be that of a "semi-transparent"[12] or *gray box* (black + white box): black in the sense that the focus is on *selecting* appropriate previously-established components for calculating properties, and white in the sense of carrying out *minor modifications* on these components to make them cope with our working context.

Throughout the rest of the current section, a deeper explanation of each property is given; justification of the property by itself and its chosen calculation method is provided, along with some examples (extracted from our Wikipedia sub-collections) and brief discussions.

4.2.1 Topic outstanding members (representative documents)

The aim of the outstanding members descriptor is to capture the essence of a topical cluster; in that sense, we could consider it as the "heart" of the topic. Connecting with our formal framework, this property corresponds to R_i and is produced by the function $\pi_r(C_i)$.

The inclusion of this feature obeys two principal interrelated reasons: structuring the topic in question and providing a manageable piece of information (in fact, such piece is helpful for doing experiments with users). Regarding the former, since outstanding member selection implies ordering documents by their relative importance, it becomes possible to *gain insight about the overall thematic* by looking at a representative document subgroup (which would be the case if we had a Pareto, Zipfian, or another power law distribution with respect to gained knowledge versus observed document number). Now, besides acting as an aid for the end user, representative document selection also is helpful for *presenting* an application-usable topic synthesis. For instance, taking the top n members of each group is useful for generating visualizations such as the one introduced by Herr and Holloway, which spots Wikipedia's most actively revised (a.k.a. controversial) articles [58] (see Figure 4.23). Another advantage consists of having significant *samples* for evaluation—especially when the evaluation type is expensive to carry out with the whole document set.

As suggested above, obtaining the outstanding members of a cluster is synonymous for *document ranking*. Moreover, because our working context is given by a hyperlinked environment, ranking matters can be turned into the *calculation of centrality/prestige*[13]. Certainly, these metrics are not extraneous for Web data, as they are calculated in Google's famous PageRank and the HITS (topic distillation) approach.

[11] National Information Standards Organization

[12] Semi-transparent boxes imply an understanding of the applied data mining model [12].

[13] For the sake of reader comprehension, let us note that centrality and prestige are being used as interchangeable terms, although for directed graphs the correct term to use is "prestige".

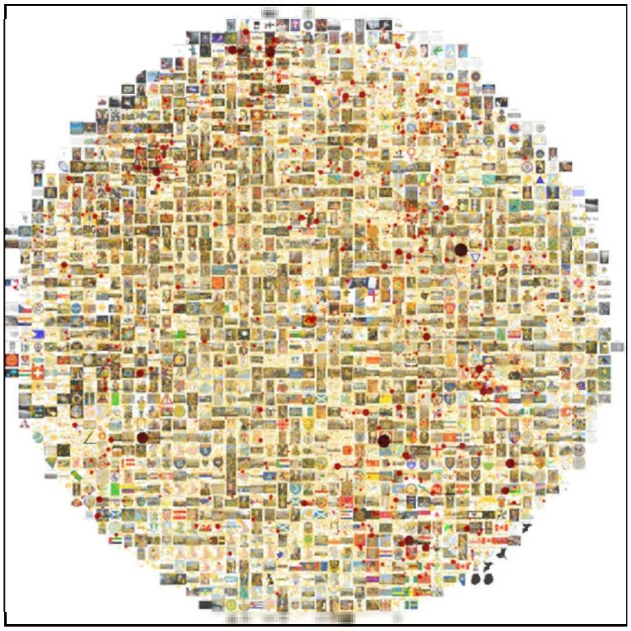


Figure 4.23: Wikipedia revision-based mosaic.

For prestige calculation, there are several alternative measures, which have already been described in previous chapters. From these, we find *degree centrality* as convenient and useful, despite its lack of sophistication, for several motives. First, it concerns a light-weight metric, as it is the one that demands less resources and operations (i.e. eigenvector centrality involves matrix computations, and closeness and betweenness centralities require geodesic path calculations). Second—and most important—, degree centrality is best suited for measuring *local relevance*, as noted by several authors (Scott [129], Feldman and Sanger [39]); this trait is significant because, more than being interested on knowing a node's prestige on the entire network, we are committed to finding its prominence on the neighborhood of the cluster it belongs to. Third, degree centrality is acquainted with our link-count working approach. Consequently, this type of centrality becomes our *criterion* for deriving R_i.

Regarding white box details, the only pertinent modification consists of solely counting as neighbors of a node those vertices that are inside the cluster. For example, if a node "A" has ten neighbors in the document graph, but only seven belong to the cluster where "A" is in, then its neighborhood size is seven. This calculation is done (independently) for every cluster "A" is in. The formula for *cluster degree centrality* is shown in Eq. 4.6.

$$\begin{aligned} C_D(v) &= \frac{\Gamma(v, \mathcal{C})}{|\mathcal{C}|}, \text{ where} \\ \Gamma(v, \mathcal{C}) &= \{n | n \in \mathcal{C}, (v, n) \vee (n, v) \in E\} \end{aligned} \tag{4.6}$$

Another variable involved within the calculation of outstanding members is the number of them to select ($|R_i|$). This actually is application dependent, as there might be instances where just a small amount of significant documents is needed (e.g. , 3 or 5) or instances where the complete document set is solicited (in that sense, the ranking *per sè* is the property of interest).

The function for obtaining representative documents is condensed in Algorithm 10, and an example is illustrated in Table 4.21.

Table 4.21: Document ranking example.

Cluster	**Centrality Scores**	**Ranking**
Computer program	Smalltalk, 1.28	(1) Smalltalk
Compiler	Objective-C, 1.25	(2) Objective-C
Datatype	Tcl, 1.2	(3) Tcl
Scripting language	COBOL, 1.18	(4) COBOL
Class (computer science)	Logo prog. lang., 1.15	(5) Logo prog. lang.
Virtual machine	Visual Basic .NET, 1.15	(6) Visual Basic .NET
Reflection (computer science)	ML prog. lang., 1.15	(7) ML prog. language
	PL/SQL, 1.15	(8) PL/SQL
Recursion	Haskell prog. lang., 1.15	(9) Haskell prog. lang.
Functional programming	Assembly language, 1.15	(10) Assembly lang.
Control flow	⋮	⋮
Imperative programming		
ALGOL 68		
Common Lisp		
Smalltalk		
ML programming language		
⋮		

Of course, not every aspect of the outstanding members feature is advantageous. For instance, centrality could prove to be less useful if there is a lack of central points in the topic (which could be the case if we have communities of "homogeneous" articles, such as counties, car models, etc.). To address this issue, first we would need to analyze what the representative document subset is to be used for; if the aim, for instance, is to provide a "peek" of the topic, there is no serious problem (although we would have invested resources in calculating centrality when a simple random sample would have sufficed). On the contrary, if a ranking is desired anyways, secondary (probably non-structural) or alternative methods would have to be considered; for example, an attractive option concerns using *eigenvector centrality*, as it has shown to be successful for the Web (HITS and PageRank being based on this type of calculation) and actually does not represent serious costs because the size of the clusters is relatively small when compared to the size of the whole collection.

Returning to the point of disadvantages, by creating topic partial views we also run the risk of generating "elitist" facets of the thematic at hand by presenting only certain documents. Once again, this depends on the final application (ultimately, the full document set is always available, and $|R_i|$ can be set to $|\mathcal{C}|$). However, a means for overcoming such situation could consist of adding flexibility to the property, as to specify whether the sorting is ascending or descending, for example.

Algorithm 10 Outstanding members function

Description: Receives a document set (cluster) $\mathcal{C}_i$ and a number k of desired outstanding members to produce a document subset R_i. Document ranking according to degree centrality is performed, and the top k documents are selected as members of R_i.

```
function RepDocs($\mathcal{C}_i, k$)                    ▷ Get $k$ most central documents

    $R^* =$ ranking of $\mathcal{C}_i$ according to $C_D$

    for $j \leftarrow 1, k$ do
        add $r_j \in R^*$ to $R_i$
    end for

    return $R_i$
end function
```

4.2.2 Topic tags

The aim of the tag descriptor is to *name* a topical cluster—in other words, to materialize the theme that gathers the document set together. With regard to our formal framework, this property corresponds to t_i and is produced by the function $\pi_t(\mathcal{C}_i)$.

Tags, managed usually as *keyword collections*, make a topic usable, both for humans and machines: humans quickly *understand* what the topic is about and machines are able to *search* the topic. Furthermore, tags are becoming universally accepted as descriptors for WWW resources [139], [133]. What is more, this feature allows to have a *complementary categorization*, which can be used to, for example, construct a hierarchy of topics (as done in works like the one of Brooks and Montanez [23]). Even when tags are traditionally associated with a kind of user input, we can also generate them automatically (which is our intention); while user tagging is by far more popular, the automatic creation of tags is not uncommon on the Web.

Regarding the method for keyword extraction, a straightforward option is to use the classical tf-idf term weighting scheme. Not only is this scheme known to be effective, but has also been successfully utilized for automatic tagging; for instance, the already mentioned work by Brooks and Montanez [23] employs this measurement to classify blog entries (using the top-three highest scored terms), while Chirita *et al.* [28] use a slight variant of this scheme to create personalized tags for Web pages. Therefore, tf-idf seems an adequate choice for extracting the most relevant keywords of a document cluster.

As with the representative document subset, a design issue corresponds to the number of terms that compose the tag ($|K_i|$). While at least a pair is reasonable, the maximum number of tag terms is application-dependent, as five terms could suffice for enabling search and a more considerable number would be probably needed to create a tag-cloud[14]. As a consequence, the tag length is left open.

[14] A tag-cloud comprises a list of the most popular tags in the system; such list is usually displayed in alphabetical order, and visually weighted by font size. When a user clicks on one of the tags, an ordered list of the resources described by the tag is displayed, as well as a list of other related tags [56].

In order to adapt the tf-idf scheme to our context, each document cluster is treated as a pseudo-document; therefore, frequency measurements are carried out as if the terms found in the cluster all belonged to a single piece of information. Another consideration concerns *stemming*; because it is convenient to avoid redundant terms in our final tag, we keep track of the used stems. So, if a keyword's stem has not already been introduced by another term, this keyword is added to the final tag. For example, note that in Table 4.22, the term "cell" is omitted from the tag, since another term ("cells") with the same stem ("cell") already appears in the tag. On the other hand, the procedure to generate a topic tag is shown in Algorithm 11

Algorithm 11 Tag generating function

Description: Receives the vocabulary from the document cluster (denoted by W_C) and a number k of desired keywords as input for composing a topic tag (a string of terms), which is returned as result. To determine the keywords that will be part of the tag, a word ranking according to tf-idf weights is performed first. Before adding a keyword to the final tag, the stem of such keyword is verified against a set of stems (conveniently called *Stems*); if the stem is already a member of this set, the word is discarded. Keywords continue to be added one by one until the tag meets the desired length.

```
function GENERATE-TAG($W_C, k$)                          ▷ Get a tag of length $k$

    $W^* =$ ranking of $W_C$ according to tf-idf

    for $j \leftarrow 1, k$ do
        stem$_j$ ← STEM($w_j$)
        if stem$_j \notin$ Stems then
            concatenate $w_j \in W^*$ to tag
            add stem$_j$ to Stems
        end if
    end for

    return tag
end function
```

Keyword selection is analogous to *word ranking*. Hence, an implication here corresponds to acknowledging the inherent use of *text* for the tag property; consequently, management of the documents' text is necessary. For that matter, let us discuss data representation for this kind of information source (which was not approached before in order to give more relevance to link information extraction).

At a physical level, text extraction mainly concerns *indexing* (see Figure 4.24). This way, document term frequencies are stored and easily accessed. As for specific indexing options, the two viable alternatives are Lucene[15] and Schmidt's Wikipedia indexer[16]. The latter, although less robust and popular, is actually a variation of the first one and can work directly with the XML dump; the former has been widely used for research purposes, but

[15] More information at `http://apache.lucene.org`.

[16] Available at `http://schmidt.devlib.org/software/lucene-wikipedia.html`.

needs a set of either HTML or plain-text files (which are achievable with Wikiprep). For these reasons, Lucene has been considered as the best choice for indexing, so far.

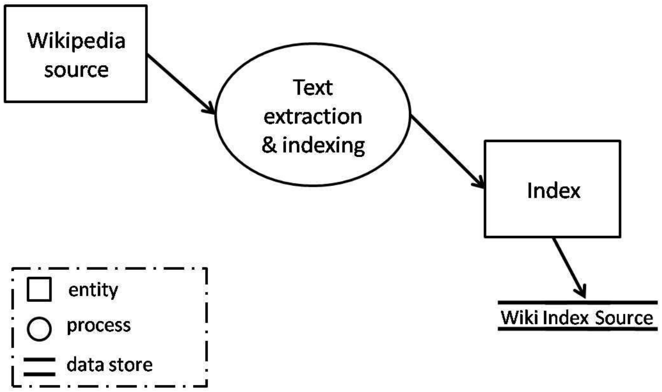


Figure 4.24: Text indexing.

Working with tags also implies several challenges to overcome. For instance, as stated by Marlow *et al.* [88], word ambiguity (synonymy and polysemy) can hinder the effectiveness of these descriptors; in fact, this issue is inherited from text usage, and has been one of the reasons for preferring hyperlinks. Related to this problem are the limitations imposed by individual words (perhaps phrases or collocations could have more expressive power). To address these issues, which are present in every text mining context, language resources (e.g. Wordnet) could be added to tag generation; that way, words could be mapped to *concepts*, thereby precising more the semantics of the cluster while approaching ambiguity. However, this is more of a future direction.

Another issue is given by scalability; although this aspect is not so crucial at this level (clusters are smaller), it still requires consideration. Because tf-idf calculation can become slower when having several thousands of terms, a "light" version for this metric could be given by the exclusive use of anchor text (titles). Nevertheless, the importance of scalability would have to be counterweighted against the importance of quality.

4.2.3 Examples

Some examples extracted from our Wikipedia sub-collection clusters can be appreciated in Table 4.23; a tag length of five keywords and three representative documents were chosen to illustrate topic properties.

Table 4.22: Keyword ranking example.

Cluster	**Tf-Idf Score**	**Stem**	**Ranking**
Blood transfusion	blood, 0.09	(blood)	(1) blood
Antibody	cells, 0.07	(cell)	(2) cell
White blood cell	cell, 0.049	(cell)	(3) cells
Basophil granulocyte	lymphoma, 0.048	(lymphoma)	(4) lymphoma
Blood plasma	antibodies, 0.046	(antibodi)	(5) antibodies
Eosinophil granulocyte	coagulation, 0.042	(coagul)	(6) coagulation
Lymphocyte	platelets, 0.041	(platelet)	(7) platelets
Neutrophil granulocyte	disease, 0.037	(diseas)	(8) disease
Coagulation	platelet, 0.033	(platelet)	(9) platelet
Platelet	factor, 0.032	(factor)	(10) factor
⋮	⋮		⋮
Final tag: "blood, cell, lymphoma, antibodies"			

Table 4.23: Examples of topic properties

t **(tag)**	R **(most representative documents)**
software, windows, system, computer, server	Computer software Windows NT Random access memory
hiv, aids, infection, antiretroviral, virus	Antiretroviral drug AIDS pandemic AIDS in the United States
jedi, wars, star, palpatine, skywalker	Star Wars: Clone Wars Star Wars Jedi Knight: Jedi Academy Star Wars Jedi Knight: Dark Forces II
algebra, vector, space, frac, group	Field (mathematics) Topological space Determinant Matrix (mathematics) Polynomial

4.2.4 Topic description: sub-task summary

To close the current section, we present in Table 4.24 the most relevant aspects of the description sub-task.

Table 4.24: Main aspects of the description sub-task.

Description.- *Concerns stipulation of topic properties and their calculation.*	
Key question	**Related aspect**
What properties are we interested in?	In a general way, we are interested in a *ranking* of a topic's documents and a ranking of a topic's vocabulary. Specifically, these properties are presented as *subsets* derived from these rankings: a term tag (t) and a collection of the most representative documents (R_i).
How are they going to be obtained?	Tags are obtained by ranking the topic's vocabulary with a *tf-idf term weighting scheme*; stemming is also employed to avoid producing redundant tag components. Outstanding members are achieved by sorting with respect to *degree centrality*.

4.3 Chapter summary

Topic construction can be regarded as the most important task of the extraction process. A first aspect of construction is the *link information extraction* endeavor, as it prepares data for the clustering procedure; for this aspect, several options were compared and evaluated according to specific criteria. Our final choice was to use Wikiprep as our source: it enables an easy access to articles and links, anchor text (titles) is available, most irrelevant material is already discarded, and the text is neat.

The bulk of construction—and the core of the whole extraction process–falls upon hyperlink-based document clustering. Our clustering approach, in abstract terms, is heavily related to *community detection*, because it searches for highly inter-linked overlapping groups. Consequently, the method is *structure-based* (graph theoretic) and assumes that topics will tend to concentrate into *maximum-cohesion subgroups*, which can be visualized as the local optima (peaks) on a multidimensional surface. Furthermore, the corner stone of the approach relies on the *Graph Local Clustering* (GLC) approach; not only has this working approach been selected because it complies with our general idea for tackling topic construction, but also for other additional advantages: an inherent ability to deal with complexity, conceptual clarity, adaptation to our (graph-theoretic) domain, and its theoretical foundations.

Another important point regarding construction comprises the concrete method, which consists of two fundamental algorithms (consistent with our formal framework), namely the *construction and clustering functions*: the former extracts a single cluster from a seed document set and is used repeatedly by the latter to produce a grouping of the collection. Recalling that the two general parameters of the construction function imply a local search strategy and

4.2.4 Topic description: sub-task summary

To close the current section, we present in Table 4.24 the most relevant aspects of the description sub-task.

Table 4.24: Main aspects of the description sub-task.

Description.- *Concerns stipulation of topic properties and their calculation.*	
Key question	**Related aspect**
What properties are we interested in?	In a general way, we are interested in a *ranking* of a topic's documents and a ranking of a topic's vocabulary. Specifically, these properties are presented as *subsets* derived from these rankings: a term tag (t) and a collection of the most representative documents (R_i).
How are they going to be obtained?	Tags are obtained by ranking the topic's vocabulary with a *tf-idf term weighting scheme*; stemming is also employed to avoid producing redundant tag components. Outstanding members are achieved by sorting with respect to *degree centrality*.

4.3 Chapter summary

Topic construction can be regarded as the most important task of the extraction process. A first aspect of construction is the *link information extraction* endeavor, as it prepares data for the clustering procedure; for this aspect, several options were compared and evaluated according to specific criteria. Our final choice was to use Wikiprep as our source: it enables an easy access to articles and links, anchor text (titles) is available, most irrelevant material is already discarded, and the text is neat.

The bulk of construction—and the core of the whole extraction process–falls upon hyperlink-based document clustering. Our clustering approach, in abstract terms, is heavily related to *community detection*, because it searches for highly inter-linked overlapping groups. Consequently, the method is *structure-based* (graph theoretic) and assumes that topics will tend to concentrate into *maximum-cohesion subgroups*, which can be visualized as the local optima (peaks) on a multidimensional surface. Furthermore, the corner stone of the approach relies on the *Graph Local Clustering* (GLC) approach; not only has this working approach been selected because it complies with our general idea for tackling topic construction, but also for other additional advantages: an inherent ability to deal with complexity, conceptual clarity, adaptation to our (graph-theoretic) domain, and its theoretical foundations.

Another important point regarding construction comprises the concrete method, which consists of two fundamental algorithms (consistent with our formal framework), namely the *construction and clustering functions*: the former extracts a single cluster from a seed document set and is used repeatedly by the latter to produce a grouping of the collection. Recalling that the two general parameters of the construction function imply a local search strategy and a cohesion fitness function, we have chosen to employ first choice *hill-climbing* and *relative density*, respectively. Such selection is based on the Ockham's razor principle: start with a simple base and add sophistication as needed. In that sense, we decided to explore seven possible *enhancements*: a removal strategy (finally included), candidate ordering (descending by prospective internal links), neighborhood type (partial), seed selection (at random), seed expansion (not included), secondary clusters (considered), and quantum addition. Inclusion and tuning of these refinements was carried out by observing the algorithm's behavior with (Wikipedia) sub-collections of smaller size.

For topic description, two main properties are being considered: a representative document subset and a tag. *Degree centrality* is suggested for generating a ranking of the cluster documents and being able to select the k most important ones. On the other hand, a *weighting scheme* based on text frequency is used for ranking words and generating the topic tag.

Once having defined how construction and description take place, we are able to carry out our methods over the case study corpus in order to extract topics. Of course, to complete the extraction process, such results would have to be evaluated.

Chapter 5

Topic validation: Experiments and Results

The primary intention of the topic validation sub-task consists of supporting a fundamental hypothesis: *the discovered groups are topics*. As a consequence, the aim of the evaluation procedure is to *provide evidence of the clusters' topical coherence* (also referred to as "topicness" or "topicality"). To properly get into validation, first it is necessary to describe the clustering performed on the Wikipedia corpus and condense the most important results (i.e., report the inputs, parameters, and outputs). Once having this clear, details pertaining to the validation methods (e.g., setup, results, and findings) can be approached.

The used validation schemes correspond to *internal* and *external* evaluation; from each scheme, we have selected two techniques—therefore utilizing four distinct ways to support our main hypothesis. On one hand, internal evaluation, which solely relies on the groups' properties, is mainly being used to get an "initial glance" of the quality (and "topicness") of our clusters. To carry out this evaluation, *cluster compliance* and the *golden threshold* techniques are employed. For the former, we take a sample of our topics and contrast intra-cluster similarity (a.k.a. compactness) versus inter-cluster similarity (a.k.a. separation) with visual matrices; in our case, we assume that alternative information sources (e.g. text) suggest a topical bond. With respect to the latter, we intend to prove group cohesion by means of a collective quality metric from SNA.

Regarding external evaluation, which consists of comparing the obtained clusters against an established model, we have opted for making an alignment against Wikipedia's category network (source for *reference classes*) and for using human judgment via *outlier detection* tests. For the first case, we are assuming that if a cluster matches a Wikipedia category, it definitely is a topic. For the second, we rely on users' criteria in order to confirm that our results correspond to topics. In general, these external techniques seem more straightforward because they are carried out at the semantic level and leave less margin for error.

An issue relevant to mention concerns the *validation scope*; because our main attempt is to show that our clusters have a topical bondage, we are mostly committed with evaluating the *construction* method. Consequently, assessing topic description (properties) is not our central aim for the present work.

5.1 Clustering Wikipedia

To prove the construction approach, our clustering algorithm was applied to the Wikipedia corpus (for specifications about this collection, see Table 5.1). Before presenting the setup of the clustering procedure, it is convenient to state that it was *inclusive*, in the sense that all content articles were considered for grouping, without making distinctions among them; thus, the whole corpus was used.

Table 5.1: Clustering data source (Wikipedia) information

Version		2005 (pre-processed with Wikiprep)
Original data set		
	Articles	911,029
	Links	21,424,034
Content-exclusive data set*		
	Articles	803,902
	Links	19,278,524
Strongly connected components		Giant SCC with 85% of nodes [36]

*= Excluding categories, lists, and unconnected nodes

A summary of the particular settings for the clustering is given by Table 5.2. As we can see, a quantum (time limit) of five minutes per cluster was set, the seed list was explored in a random order, and we chose to use only partial neighborhoods.

Table 5.2: Clustering settings

WIKIPEDIA CLUSTERING

▷ Partial neighborhoods
▷ Random seed ordering
▷ Quantum: 5 minutes (per cluster)
▷ Element removal
▷ Candidate ordering (internal link contribution)
▷ First improvement
▷ Secondary cluster (accepted if $\rho >= 0.3$)

An overview of the clustering results is shown in Table 5.3 (later on, several concrete examples of clusters will be presented as well). On one hand, as we can see, basically an 80% of the collection was placed into a corresponding document group, and the average size of these groups was of 100, approx.

It is important to note that, up to this point, we only present one run of the clustering algorithm at the Wikipedia scale. However, it is highly desirable (if not necessary for certain purposes) to eventually carry out more runs with the intent of improving the reliability of our results and the robustness of the proposed model.

Table 5.3: Wikipedia clustering summary

	WIKIPEDIA CLUSTERS
# of clusters	317,093
Coverage (docs.)	629, 538
	78%
Cluster size	
Avg.	121
Min.	2
Max.	3,819
Cohesion	
Avg.	0.3
Min.	0.014
Max.	1.0

5.1.1 Result filtering

An important part of the data mining process consists of *getting useful results* [138]. This step, on its own, implies filtering out information that does not seem promising—logically being aware that this decision will also incur into costs, such as losing elements that were actually good (up to this point, we still do not know that) and decreasing coverage. Nevertheless, because we have made clear that quality is more prominent for us than coverage, filtering results seems necessary.

As we can see from the summary of our clustering, several groups exhibit a very low density (e.g., 0.02). While this could be due to them representing topics of little cohesion, it could also be the case that these clusters are not well constructed (e.g. the seed was not appropriate, the algorithm converged too soon, etc.); as a consequence, it seems difficult to tell if these groups will actually serve. Then, a reasonable choice is to dismiss such clusters.

The above statement introduces an additional problem: how to decide if a cluster should be dismissed or not. A straightforward criterion that allows us to judge based on an external accepted definition and avoid dubious procedures consists of keeping only those clusters that represent communities in the weakest sense; that is, preserving groups whose density is equal to or greater than 0.5. Although this filter is indeed strict, it seems preferable to have a small number of dense groups than a considerable number of clusters whose quality is not even be backed up by a strong cohesion.

Table 5.4 presents clustering information after filtering. It is interesting to note that, despite of roughly preserving 10% of our clusters, document coverage did not drop dramatically. In fact, half of it is actually given by the modest quantity of chosen groups; as a result, we could talk about these groups as constituting the *nucleus* of the clustering. Another interesting datum concerns the average size of the clusters, which is larger on these selected portions. Also, let us note that the two clustering might as well be seen in combination.

Along the rest of the chapter, different validation forms shall be applied to these results to appreciate their quality.

Table 5.4: Filtered results

	FILTERED
# of clusters	55,805
Coverage (docs.)	348,238
	43.3%
Cluster size	
Avg.	263
Min.	2
Max.	3,819
Cohesion	
Avg.	0.61
Min.	0.5
Max.	1.0

5.2 Internal validation

The principal intention of internal validation with respect to our results is to provide an *initial overview* of their quality. Moreover, with this type of evaluation it is possible to carry out topic validation both at the *clustering* and *semantic* levels. For our case, to prove internal quality, we are to apply both the golden threshold and cluster compliance sub-types.

Our main assumptions here are the following:

1. *Traces of intra-cluster similarity (compactness) and inter-cluster dissimilarity (separation) indicate well-formed groups; in addition, if the cluster compliance metric is unrelated to the topic extraction information source (hyperlinks), this suggests that the groups are topics.*

2. *Proving cohesion by alternative means points out well-formed clusters as well.*

As we have seen, there is more than one cluster compliance metric available for evaluation; this makes an adequate selection to become necessary. Regarding this aspect, there are several complementary criteria that we might take into consideration:

Link and text-based metrics The former, on one hand, serve as an alternative, parallel way of proving quality by using the same information source (hyperlinks). The latter, on the other hand, are valid since it has been proved that a document is similar in content to the documents that link to it [92]; furthermore, text-based approaches in general are valuable because they result to be completely *orthogonal* (complementary) with respect to a link-based approach. In that sense, using text to validate hyperlinks truly confides a *semantic measurement.*

Similarity and dissimilarity metrics Similarity, on its own, is more oriented towards intra-cluster proximity; dissimilarity is more focused on inter-cluster proximity. Consequently, considering both of them yields a wider perspective.

Generic and domain-specific metrics While generic metrics can be seen as reliable, standard measurements that are applicable in a considerable number of different contexts, domain-specific validation methods (where we consider Wikipedia as our domain) constitute an option specifically tailored for our working context.

Attempting to cover all of these facets, three metrics have been chosen: *cosine similarity*, *semantic relatedness*, and the *Jaccard index* (see Table 5.5 for a detailed classification). Besides this characterization, let us further justify the use of each metric:

Table 5.5: Metric classification

	Cosine	**Semantic Relatedness**	**Jaccard**
Information source	Text	Links	Links
Proximity type	Similarity	Dissimilarity	Similarity
Universality	Generic	Domain-specific	Generic

Cosine similarity. Despite of its shortcomings (e.g., does not take ambiguity into account), it is a classical—and yet simple—measurement for fields related to information retrieval. Also, because it constitutes a pure-text approach[1], it allows to validate at the semantic level.

Semantic relatedness (distance). Similar in spirit to the notion of *co-citation*, this is actually a distance metric for Wikipedia articles that was inspired by the Normalized Google Distance [31]; it takes into account the number of pages that link to two articles separately and to their intersection. If the distance is 0, it means that the pair of documents is linked by the same sources; if the documents do not share linking pages, the distance becomes infinite. It represents an alternative link analysis approach and is specific for the Wikipedia corpus.

Jaccard index. This is a well-known metric, and can be adapted to our context by taking each document as a *set of links* (out-links, specifically). As a matter of fact, using the Jaccard index this way resembles a similarity based on *bibliographic coupling*.

For the golden threshold, the two metrics whose quality threshold is known are *modularity* and the *relative strength ratio* (Eq. 2.17 on Chapter 2, Section 2.1.4). Because the first one is suited for evaluating partitions, the latter seems more appropriate. As a consequence, a fair cluster quality can be shown by surpassing a ratio of 1.0.

5.2.1 Setup

For the cluster compliance evaluation, let us note that the basic operation consists of pairwise comparisons between documents. Therefore, cluster overall proximity is taken as the *average*

[1] Even though cosine similarity is usually employed with word weight vectors, it actually can be used with other vector types (e.g. links).

proximity that results from these comparisons. With regard to intra-cluster similarity, the calculations are between documents of the same cluster; otherwise, they are between documents belonging to two different clusters.

Also, as we may recall from previous chapters, pairwise comparisons are expensive in terms of, both, time and space. Acknowledging these limitations, two actions were carried out:

1. Each cluster was characterized by means of its representative document subset (R_i), being 30 the size of such subset. Besides reducing calculation costs, this assures that we deal with the essence of each topic.

2. A systematic sample of 100 groups was extracted from our clustering. In a systematic sample, each element is chosen after k steps, where k results from dividing the total number of elements by the desired sample size. Unlike a simple random sample, which by chance may take elements that all lie within the same portion of the population, systematic sampling always selects members at different points of the element list. For this reason, we believe it is more convenient to use.

Eq. 5.1 shows how the value of each matrix cell was calculated: by taking the average proximity (where proximity is either similarity or distance) that results from performing pairwise comparisons between *documents*, either of the same cluster (this would correspond to intra-cluster proximity or *compactness*) or from different clusters (this would, analogously, be equivalent to inter-cluster proximity or *separation*). Furthermore, Eq. 5.2 presents how semantic relatedness between a pair of documents is calculated (as the reader can see, this calculation is mostly based on in-links); note that $\mathbb{W}$ stands for the total number of Wikipedia articles. On the other hand, Eq. 5.3 shows we specifically compute the Jaccard Index, i.e., by taking a document as the set of its corresponding out-links. With regard to cosine similarity, this calculation was previously discussed and presented in Eq. 2.30 of Section 2.1.9.

$$\begin{aligned}
\text{compactness}(\mathcal{C}_i) &= \frac{\sum\limits_{d_a,d_b\in\mathcal{C}_i} \text{proximity}(d_a,d_b)}{|\mathcal{C}_i|(|\mathcal{C}_i|-1)} \\
\text{separation}(\mathcal{C}_i,\mathcal{C}_j) &= \frac{\sum\limits_{d_a\in\mathcal{C}_i,d_b\in\mathcal{C}_j} \text{proximity}(d_a,d_b)}{|\mathcal{C}_i|\cdot|\mathcal{C}_j|}\text{, where:} \\
\text{proximity}(d_a,d_b) &= \text{cosim}(d_a,d_b) \vee sr(d_a,d_b) \vee J(d_a,d_b)
\end{aligned} \tag{5.1}$$

$$sr(d_a,d_b) = \frac{\log(\max|\Gamma_i(d_a)|,|\Gamma_i(d_b)|) - \log(|\Gamma_i(d_a)\cap\Gamma_i(d_b)|)}{\log(\mathbb{W}) - \log(\min(|\Gamma_i(d_a)|,|\Gamma_i(d_b)|))} \tag{5.2}$$

$$J(d_a,d_b) = \frac{|\Gamma_o(d_a)\cap\Gamma_o(d_b)|}{|\Gamma_o(d_a)\cup\Gamma_o(d_b)|} \tag{5.3}$$

With respect to the golden threshold evaluation, because the requirements are more modest and several of the needed data had already been collected at clustering time, the relative strength ratio was obtained for all of our (filtered) results.

5.2.2 Results and discussion

Results with regard to cluster compliance are shown as proximity matrices in Figures 5.1, 5.2, and 5.3. With the purpose of reviewing these visual structures, let us synthesize the principal elements of the matrices:

- The main diagonal should outstand in a higher or lower intensity if the clusters are well done.
- All matrices are symmetric.

Because the figures themselves portray results better than a thorough explanation, it seems more beneficial to solely complement these graphical descriptions. Regarding cosine and Jaccard similarities, it is relevant to note that very high values are usually not expected, as documents would have to be nearly identical for this to happen. Logically, this involves having low intensity cells in general; however, a diagonal pattern can be clearly noted on the two illustrations (probably this pattern is better appreciated on the Jaccard matrix).

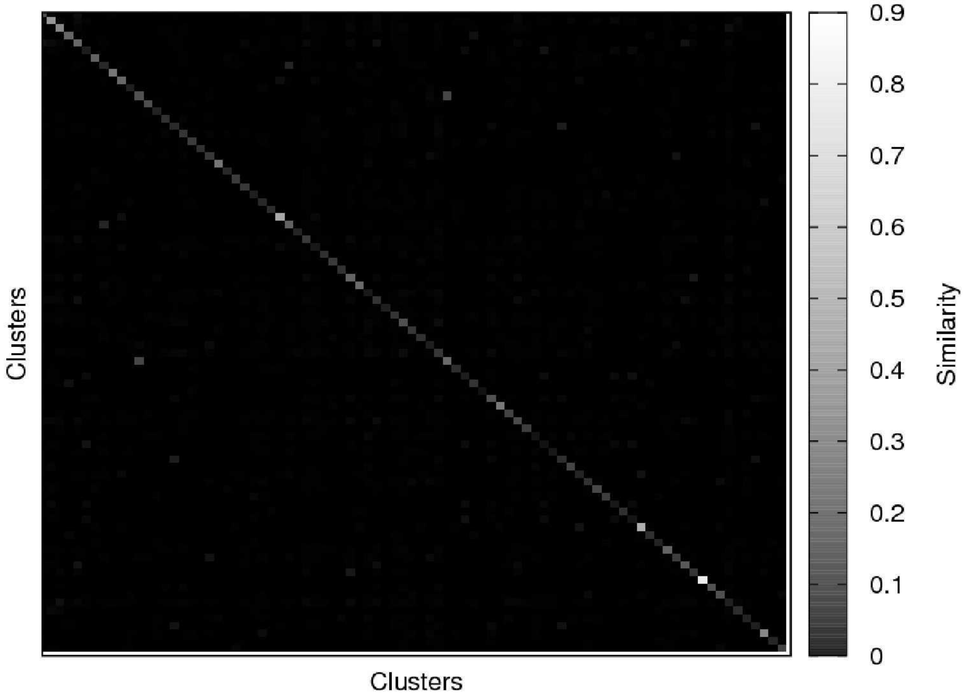


Figure 5.1: Cosine similarity

With respect to semantic relatedness, because its possible values are not bounded by a clean range (unlike the past metrics, where similarity lies between 0 and 1), it seems more difficult to notice the diagonal pattern visually. As a result, we have opted to instead present a matrix composed of the amount of infinite values found when comparing a pair of clusters. Recalling what this value stands for, when two documents have no common citing sources, the distance (semantic relatedness) between them goes to infinite. Therefore, the number of infinites should be high when comparing two different clusters and low when comparing elements of the same cluster. As we can see, this pattern is accomplished by our results.

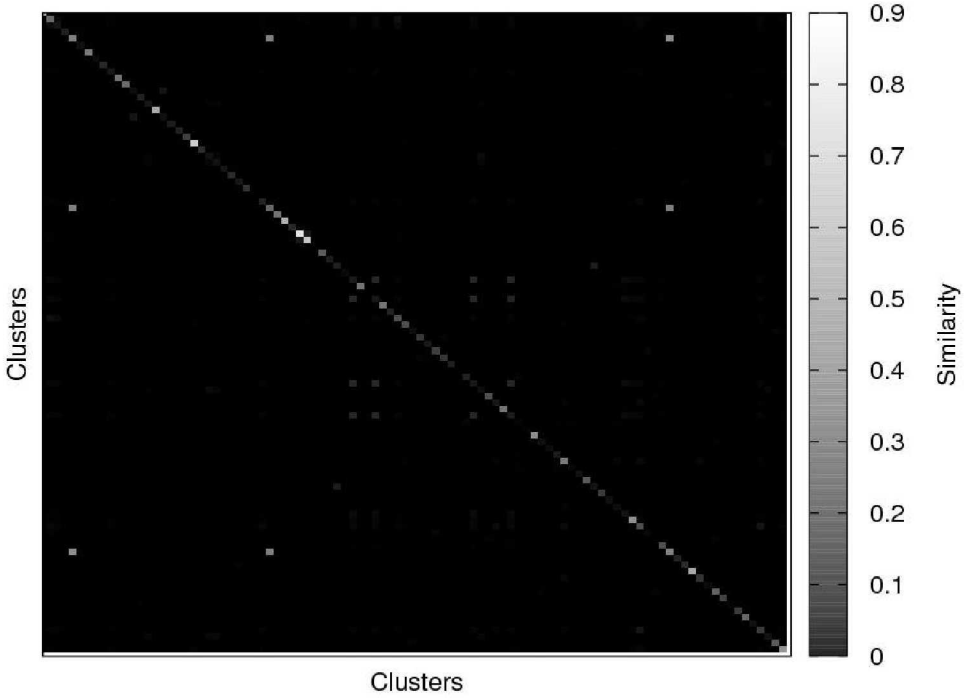


Figure 5.2: Jaccard index

To sum up the graphical results and provide numbers as well, the average proximity with each metric for the clustering is shown in Table 5.6. Like we can see, documents within the same cluster are at least forty times more similar among each other than with respect to elements in other clusters, and lie at least two times closer among themselves. The most dramatic difference is given by the Jaccard index, where the average intra-similarity is a hundred times greater than the inter-similarity. Furthermore, cluster intra-similarity reached peaks of 0.8 for both cosine and Jaccard, while achieving a minimum distance of 0.0 with semantic relatedness.

Table 5.6: Internal evaluation average results

	Cosine sim.	**Jaccard index**	**Sem. rel.**	**Sem. rel.** (infinites)
Compactness (intra-cluster)	0.23	0.19	0.23	44.47
Separation (inter-cluster)	0.005	0.001	0.42	425.52
Difference ratio	*46:1*	*190:1*	*1:2*	*1:10*

Results for the relative strength ratio can be seen in Table 5.7; let us note that less than 1% of the clusters lied below the threshold. On the other hand, it is also interesting to observe that there is a weak correlation between the strength ratio and relative density, although this is partly expected, as both measurements have similar foundations.

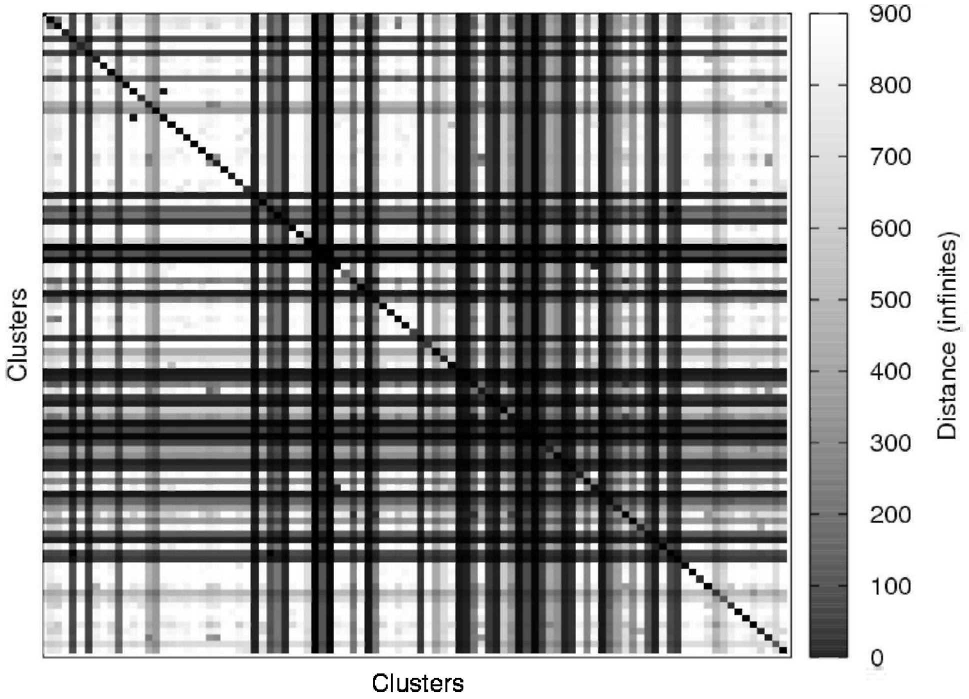


Figure 5.3: Semantic relatedness (infinites)

Table 5.7: Relative strength ratio results

Avg.	24.9
Min.	0.38
Max.	10,022.5
Elements below 1	52
Correlation between density and relative strength ratio: **0.21**	

To sum up the current discussion, it seems fair to state that our internal validation results reveal a well formed clustering. Regarding this aspect, the three chosen metrics for cluster compliance support this claim on an individual and collective basis; the former is accomplished by the consistent depiction of a compliance *pattern* and the latter is achieved by the *coverage* of different perspectives (e.g. similarity and dissimilarity, text and links, semantic and clustering levels etc.). The group quality (golden threshold) measurement, on its own, represents an alternative form of proving cohesion.

Nevertheless, it also seems fair to discuss the limitations of this kind of evaluation. Perhaps one of the most important concerns *scalability* (both for the validation approach itself and visualization tools); in that sense, the larger the sample, the more expensive the evaluation procedure became. Therefore, with this validation approach we can obtain only a *partial perspective* of quality. Similarly, the strength ratio solely provides an initial quality insight.

Seeing all of this, it becomes convenient to use a different kind of evaluation to have a wider perspective on the quality of our results.

5.3 External validation

Recalling from previous chapters, external evaluation consists of contrasting the resultant *clusters* against an established *model*, which can either involve a set of *reference classes* or the *judgment* of a human. In both cases, the validation is carried out at the *semantic level*: on one hand, for our case, the reference classes are known to have been gathered according to topicality; on the other hand, a human being understands that the logical relation among a group of articles is a common theme and is able to tell if this is the case or not. In that sense, the quality and topicness of our groups shall be assessed via these two methods.

5.3.1 Alignment with Wikipedia's category network

Even when Wikipedia is not a specialized corpus for experiments (this kind of corpora have tags, just as training and test sets, for performing experiments), it is also true that it contains a *category network* that can serve as a source for extracting reference classes. With this structure, which is similar to a hierarchy, an *alignment* between clusters and categories can be done. By alignment, we refer to a one-to-one relation, where a given cluster is the best representative of a category and viceversa; in other words, the idea is to match or "marry" clusters and categories to measure the existing correspondence between our grouping and the provided classification. As a consequence, our central assumption consists of the following:

If one of the discovered groups matches a Wikipedia category, it can be considered as a topic.

With regard to the aforementioned network, categories are arranged into a *loose hierarchical structure* that contains both general and specific classes (categories); by "loose", we mean that the hierarchy resembles a tree, but is not actually one (we will go over this with more detail later). Now, using this network comprises several advantages. First, it is reliable to a considerable extent, since it has been created and revised by Wikipedia editors (therefore constituting a "wisdom of crowds" resource); also, its importance cannot be underestimated, as its use has been reported in literature as well. Furthermore, the network covers basically the whole corpus (see Table 5.8 for more details), and, because a single document may appear in several categories, it also proves useful for evaluating a non-exclusive clustering. Finally, information from this category network is available and condensed in Wikiprep files, so it is easier to extract.

However, using the network also imposes several challenges, where the main one concerns handling its *quasi*-hierarchical structure. Before going deeper into this aspect, it seems more convenient to first discuss another fundamental matter: how to evaluate a flat clustering with a hierarchy of classes. An accepted solution for this disparity consists of finding the reference class that best resembles a given cluster; this is usually achieved by means of the *F-score*.

Let us explain the previous statement with more detail: if we take in consideration that at the very top of the category tree we have very general classes, our clusters will have a very high precision (1.0 even). However, recall will be very poor. On the contrary, if we take a very specific class, recall will reach high values, while precision is to drop dramatically. So, in order to have a fair evaluation, first we need to *place the cluster in the class of most similar size*; that way, our evaluation will not be inflated or deflated by size differences. An option for placement, as mentioned earlier, is to employ F-score, since it aims to balance precision and recall. Therefore, we can opt to take precision and recall at the best F-score value. This is illustrated by Figure 5.4. Because we do not know if our clusters will tend to have better precision or better recall, we shall use the normal formula (F_1) and not the weighted variants.

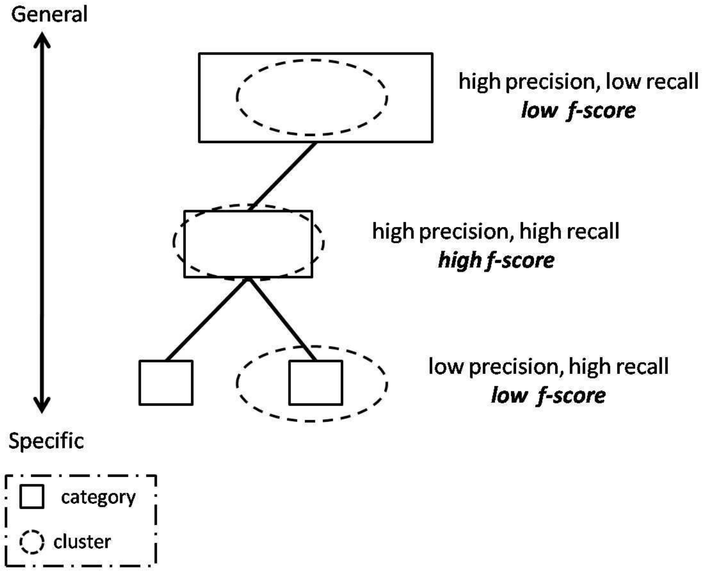

General
high precision, low recall
low f-score
high precision, high recall
high f-score
low precision, high recall
low f-score
Specific
category
cluster

Figure 5.4: F-score as indicator of good correspondence.

Another important consideration is *category expansion*. Under this rubric, we can view each category in a *collapsed* (shallow) or *expanded* (deep) mode. The first view only considers as members those documents that are present within the current level, and the second one considers these documents *and* the ones that belong to sub-categories; for example, at the first level of the hierarchy, a shallow view would only take as members those articles directly referenced by the root, and a deep view would take all articles as belonging to the root. Because a cluster may resemble either one of these modes, we should have both of them available.

Consequently, for evaluation we need to be capable of going up and down the hierarchy and expanding categories without getting cycled. Because in the *quasi*-hierarchical structure categories are allowed to have several parents or appear in more than one level (which creates cycles), a proper *category tree* has to be constructed from this structure. The construction involves several steps:

- *Information source detection.* Wikiprep provides a list of every parent category and its immediate (first level) descendants; additionally, the article XML file contains a `<category>` tag, which indicates for each document the (immediate) categories it belongs to. Both sources suffice to construct our expandable tree of categories.

- *Root and leaf detection.* With the descendant list previously mentioned, we can recognize as roots those categories that are not listed as children of others, and, as the leaves of our tree, those categories that do not have children. Formally:

  – Parents $= \{p|\, p, ch \in V \wedge (p, ch) \in E\}$

  – Children $= \{ch|\, p, ch \in V \wedge (p, ch) \in E\}$

  – Roots $= \{r|\, r, p \in V \wedge (p, r) \notin E\}$

  – Leaves $= \{l|\, l, ch \in V \wedge (l, ch) \notin E\}$

- *Level assignment.* To enforce a strict hierarchy, we can mark every category with a corresponding unique level. To know which is this level, a breadth-first strategy (BFS) can be applied: mark all roots with a level equal to 0, then mark their (direct) children with level 1, and so on. If a category attempts to be marked when it already has been assigned to another level, the link between the category and this "second parent" (or third, fourth, etc. depending on how many parents the category has) is broken to eliminate a possible cycle. Of course, this *normalization* procedure (detailed in Algorithm 12) implies losing some information from the category network, but at the end a tree structure is produced. Results from this assignment are shown in Table 5.9.

- *Expansion.* By employing BFS, this task becomes trivial: we first add the documents that properly belong to the category and then the ones that belong to children categories. If one of the children also has descendants, the procedure is repeated once again, and this is done until the leaves of the tree are reached.

Another important consideration comes from *uncategorized elements*, which should be eliminated at the time of evaluating because they actually represent noise.

5.3.2 Results and discussion

The most relevant results for the cluster-category alignment are listed in Table 5.10. Because small-sized clusters and clusters that only count with a few categorized elements actually represent noise, at this point we considered it necessary to introduce an additional, uniform cut-off value. In that sense, aligned clusters with 30 or more elements were solely taken.

Algorithm 12 Category Tree Construction (normalization).

Description: Receives a set of roots and sweeps the tree-like structure in a breadth-first fashion for marking every node with its corresponding level. Therefore, once a parent node (starting by the roots) has been marked, its descendants are added to the element queue for being marked later as well. If a node has already been visited and there is an attempt to mark it again, a multi-parent relation is detected and the edge between that node and its "second" parent is deleted.

```
function ASSIGN-LEVELS(Roots)
    currentLevel = 0
    for all $root \in Roots$ do                                   ▷ First mark all roots.
        mark root with currentLevel
        add root to Queue
    end for

    increment currentLevel by 1
    while !empty(Queue) do
        parentCategory=GET-FRONT-ELEMENT(Queue)
        Children=GET-CHILDREN(parentCategory)

        for all $child \in Children$ do
            if marked(child) then
                delete (parentCategory, child)
            else
                mark child with currentLevel
                add child to Queue
            end if
        end for

        increment currentLevel by 1
    end while
end function
```

Table 5.8: Basic information from Wikipedia's category network

Feature	Description
Categories	77,972
Corpus coverage	$\approx 80\%$
Roots	96
Leaves	48,825
Levels	16
Level mode	7

A first important result is given by the moderate *correlation* found between F-score and relative density; such outcome is determinant because it *confirms that structural cohesion is an indicator for topicness*.

With respect to quality, Figure 5.5 presents the typical precision vs. recall curve and Table 5.6 shows a similar cumulative view of F-score with respect to clusters; as we can see from both figures, quality does not drop significantly while reviewing a larger amount of clusters, and this is an encouraging result. Furthermore, although quality on average neither is excellent nor poor, it is important to highlight that 23% of the clusters accomplished a perfect or nearly-perfect F-score. In fact, another interesting finding is that approximately 80% of those groups whose density was greater than or equal to 0.8 were awarded a score that was superior to 0.5 (in plenty of cases, even superior to 0.8). Consequently, the correlation between ρ and F_1 seems to be stronger on this density range.

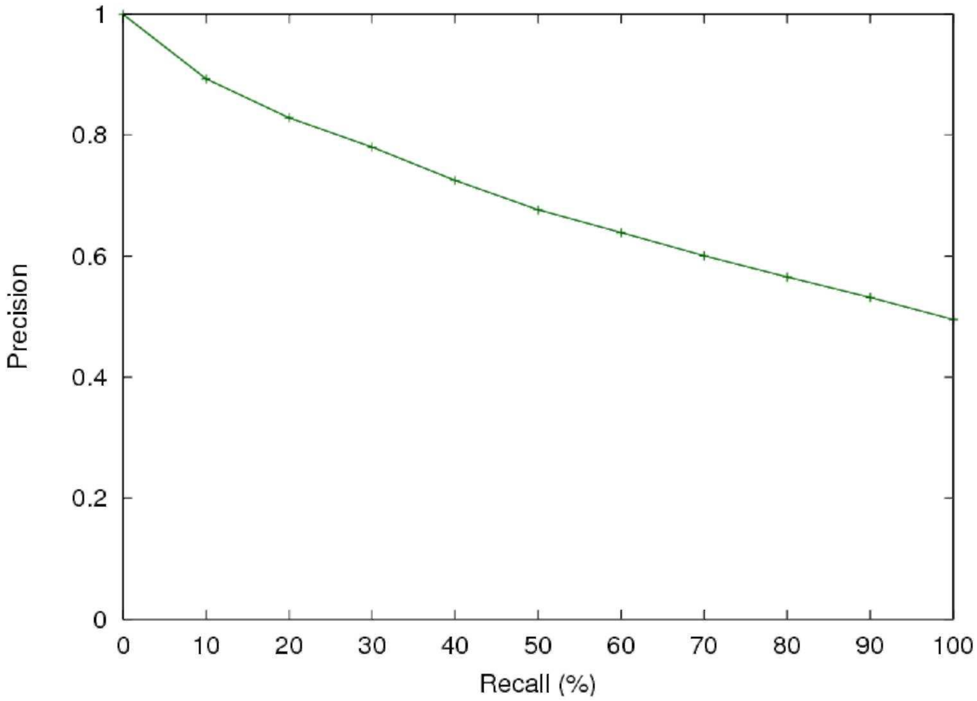


Figure 5.5: Precision vs. recall results

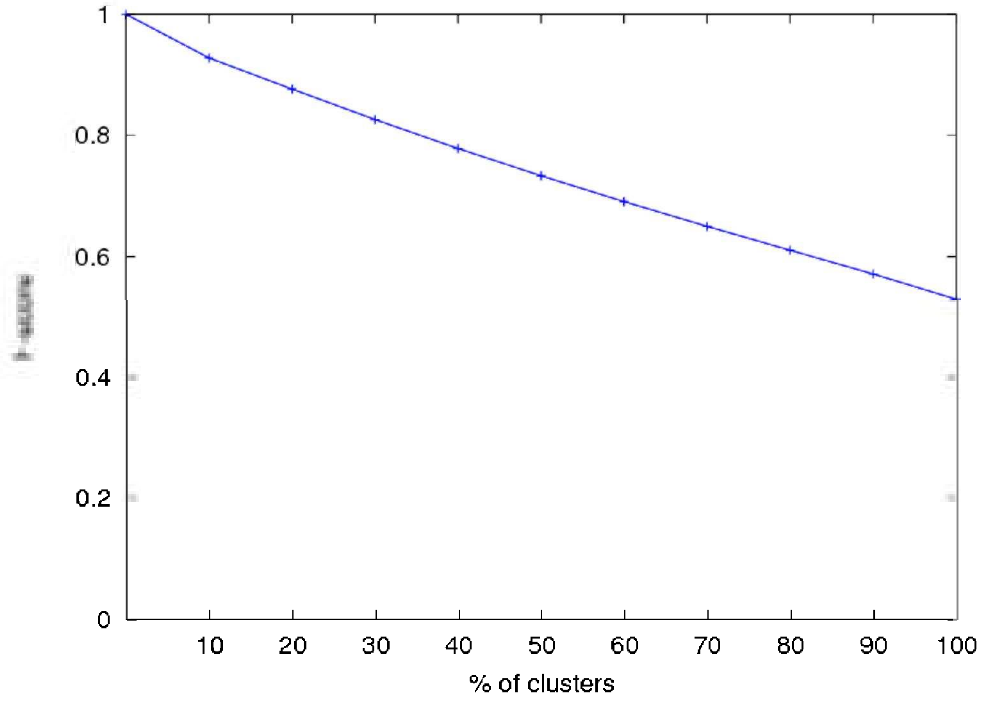


Figure 5.6: F-score results (sorted in descending order)

Regarding coverage information, it results interesting to realize that the 2,000 aligned clusters represent a 30% of the corpus, taking into account that 300,000 (the total amount) barely doubles this coverage (78%); we believe the former might be a consequence of overlapping, redundancy, and cluster size. In any case, recalling that our filtered results comprised a core already, it would be fair to consider the aligned groups a *meta-core* of our clustering. Now, with regard to coverage in general, it is definitely low (more with respect to topics than to documents). Even when this strongly suggests enhancing our method, there are also three aspects that we need to additionally consider. First, there might be a certain number of topics that have not been included in the reference class set; in that sense, these would have been *genuinely discovered* by the method. Second, approximately 40% of the aligned topics corresponded to *expanded* categories; therefore, they are actually covering more than one reference class. Third, it is important to remind that our primary priority concerns quality; once achieving good results on the former, it is easier to focus only on coverage.

Table 5.11 shows some representative clusters arranged according to density. From this reduced set, some typical features found in our topics are present, though. For example, even when relative density is bounded by nodal degree, we have also observed that, parting from a stable cluster size (30), combinations of different densities and sizes can be found among our groupings. So, it is possible to find small clusters tightly connected (such as the "Paralympics" group) or large clusters that are more loose (e.g. the "Algebra" group). Another phenomenon we have observed is that, specially with academic disciplines, the achieved clusters are "too big" for a sub-discipline (e.g. "Algebra"), but also "too small" for the general discipline (e.g. "Mathematics"); it would, therefore, seem that our construction process either got half-way to discovering the whole theme or failed to stop on time. We believe this behavior could be due to several reasons, i.e. the quantum imposed and/or the graph structure itself. Now, an even more interesting trait that we have observed concerns clusters with a very high density; like we can see from the last example, these structures clearly look (and we know for a fact they are) like *communities*: topic communities. As a matter of fact, if we could measure homogeneity by counting the words or phrases that are repeated on the set of titles, it would not be surprising for such topic communities to be *highly homogeneous*. On the other side, recalling from previous chapters, homogeneity has been a characteristic feature coined to cohesive subgroups in *social networks*. Then, the presence of topic communities provides evidence that this feature can be found as well in *information networks*. Another trait that we are able to notice when observing the matching categories of the extracted clusters is that topic communities (that is, the most dense groups) usually represent specific themes, e.g. counties of a certain state in the U.S., football clubs of a specific nationality, worldwide events, etc. regardless of their size; less cohesive, large-sized clusters, on the other hand, tend to represent broader subjects, such as academic disciplines.

Table 5.9: Categories per level

Level	Categories	Percentage
1	96	0.1%
2	302	0.4%
3	217	0.3%
4	949	1.2%
5	11,377	14.6%
6	20,014	25.7%
7	23,225	29.8%
8	18,035	23.1%
9	2,572	3.3%
10	623	0.8%
11	337	0.4%
12	127	0.2%
13	87	0.1%
14	9	0.012%
15	1	0.001%
16	1	0.001%

Table 5.10: Cluster - category alignment results

Aligned clusters	2,053
Density	
Avg.	0.62
Min.	0.5
Max.	1.0
F-score	
Avg.	0.53
Min.	0.05
Max.	1.0
Precision	
Avg.	0.51
Min.	0.03
Max.	1.0
Recall	
Avg.	0.64
Min.	0.05
Max.	1.0
Cluster size	
Avg.	177
Min.	30
Max.	2,754
Coverage	
Documents	247,019 31%
Topics	2,053 11%
Correlation between F-score and relative density: **0.34**	

Table 5.11: Aligned clusters with varying densities

beatles; lennon; mccartney; song; album *Category:* The Beatles *Cluster size:* 351 F_1: 0.71 ρ: 0.51 The Beatles The Beatles discography John Lennon Paul McCartney George Harrison Ringo Starr George Martin Paul Is Dead The Beatles' influence History of the Beatles Apple Records Fifth Beatle	**harry; potter; voldemort; hogwarts; dumbledore** *Category:* Harry Potter *Cluster size:* 274 F_1: 0.84 ρ: 0.67 Harry Potter Hogwarts Harry Potter and the Order of the Phoenix Harry Potter and the Half-Blood Prince Lord Voldemort Harry Potter (character) Harry Potter and the Goblet of Fire Albus Dumbledore J. K. Rowling Ron Weasley	**acid; reaction; chemical; carbon; hydrogen** *Category:* Chemistry *Cluster size:* 2,662 F_1: 0.4 ρ: 0.73 Chemical compound Chemical formula Organic chemistry Chemistry Biochemistry CAS registry number Melting point Hydrogen Oxygen Carbon
b; space; algebra; vector; matrix *Category:* Algebra *Cluster size:* 2,754 F_1: 0.5 ρ: 0.82 Mathematics Real number Complex number Topological space Function (mathematics) Vector space Field (mathematics) Group (mathematics) Topology Set	**artery; vein; anatomy; blood; iliac** *Category:* Arteries *Cluster size:* 94 F_1: 0.62 ρ: 0.9 Aorta Pulmonary artery Pulmonary vein Venae cavae Superior vena cava Femoral vein Femoral artery Inferior vena cava Portal vein External iliac vein	**paralympics; olympics; summer; winter; games** *Category:* Paralympics *Cluster size:* 32 F_1: 0.92 ρ: 1.0 Paralympic Games 2004 Summer Paralympics 1988 Summer Paralympics 1980 Summer Paralympics International Paralympic Committee 1976 Winter Paralympics 1964 Summer Paralympics 1972 Summer Paralympics 1992 Summer Paralympics 1984 Summer Paralympics

Two reference points for situating the result achieved quality in comparison to other approaches are given by the works of Garza et al. [47] and Wartena and Brussee [144]. The former corresponds to a proper comparison among GLC, k-means, and Principal Direction Divisive Partitioning (a.k.a. PDDP, this method is a hierarchical divisive algorithm based on Principal Component Analysis) done on one of our sub-collections (Quantumpedia); while GLC obtains only a partial coverage of the corpus (on the contrary of the other two, which guarantee to place every document into a cluster without exception), with regard to cluster compliance, the approach not only showed to be competitive, but also the best in two of the used metrics. With respect to the latter work, it is a topic identification approach whose aim is to cluster keywords by using a similarity metric based on probabilistic model similarity; the resulting clusters were also aligned with Wikipedia categories (eight in total) and the F-score was of approx. 0.6 on average; our alignment results show similar results, although considering that our amount of clusters amply surpasses the ones of this approach.

5.3.3 User Tests

Evaluation by means of Wikipedia's category hierarchy provides a very *fine-grained* perspective of the quality in our topics, and is practically equivalent to an *expert's point of view*. Despite this assessment is quite useful, its exhibited drawbacks (rigid structure, loss of information due to regularization) motivate us to additionally validate our approach in a more panoramic fashion. For this reason, it seems convenient to complement our validation scheme with *user tests*.

The aim of performing tests with users is to use human judgment directly for showing, in an overall form, that our detected groups are actually topics. In that sense, our principal interest is not to see if our document groups strictly match a reference topic, but to discover if they make sense to a person whose expertise on the subject is variable (neophyte to nearly expert). Consequently, we are assuming the following:

If a document group is a topic, a human being should be able to confirm this fact.

Even when the former sounds simple, this validation endeavor implies several punctual aspects, such as test set creation, user gathering, and experimental design. Perhaps a clearer explanation can be given if we start by this last aspect.

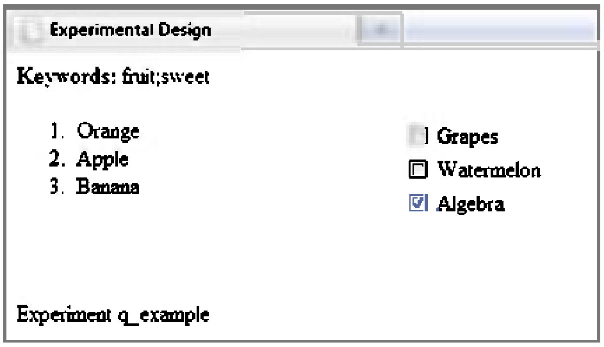
A user-based task that allows to express evaluation results in terms of accuracy (recall, precision) and has been used recently on the topic mining domain consists of *outlier detection*. For example, in the topic modeling work by Boyd et al., users are tested with two different intrusion tasks: a) selecting the outlier term from a related group of words (*word intrusion*) and b) selecting the topic that does not correspond to a certain document by seeing the document's title and summary (*topic intrusion*) [20]. Taking this as a starting point, we might be able to design an experimental framework according to our needs.

From the former outlier detection example, there are three characteristics regarding test presentation that we consider important to take into account: *summarization*, *relevance*, and overall *understanding*. These key features, we think, are general enough to be applicable for any task that is based on intruder detection; let us explain this further.

With respect to summarization, it is important to realize that users cannot be overloaded with information. This would only lead to dissatisfaction, lack of people who want to do the test, and, ultimately, negative results. Therefore, it seems logical to present each experiment (specifically, each topic to be evaluated) in a *concise* way. Seeing that this is basically mandatory, it is convenient as well to provide the *essence* of the topics we want to validate, since it captures a clear view of them. Furthermore, this enforces the third aspect, which consists of providing a group that is collectively understandable, either because it seems logical or because it is *known*. This characteristic is fundamental if we want users to properly detect outliers.

Aside from selecting the topic information to present, the other aspect that contributes to a robust experimental design regards *choosing the outliers*; however, this issue is easy to tackle, as we can take documents at random from other groups also taken at random.

Bundling everything up, our general approach would then consist of presenting cluster *properties* (tag and list of representative document titles) as the "positive example" for each topic and, on a separate block, showing a certain number of documents that may or may not be part of the topic in question (this document list represents the test, properly speaking); the test list would be composed of representative documents as well, both for the true members and the outliers. This design requires two actions on the part of the user: firstly, to *understand in coarse terms* what the topic is about, and, secondly, to be *able to tell* its elements apart. Seen from a didactic point of view, such final design resembles the exercise of identifying *semantic fields*[2]. Please refer to Figure 5.7 for an example of our rough sketch; this example is artificial in the sense that it does not contain data from the obtained clusters, but it was actually used to illustrate users when testing.



Experimental Design

Keywords: fruit;sweet

1. Orange
2. Apple
3. Banana

☐ Grapes
☐ Watermelon
☑ Algebra

Experiment q_example

Figure 5.7: Experimental design (general)

Setup

Besides experimental design, we were also initially concerned about another pair of relevant aspects for our evaluation: creation of the test set and user gathering. While the former is covered by specifying how every experiment is configured, the latter is addressed by describing the platforms used for testing.

With respect to the test set, it is important to first state how the clusters for experimenting were chosen. In that sense, the candidate cluster pool was made up of the aligned groups, mainly because these constitute the core of our clustering. Furthermore, since these clusters

[2] Semantic fields comprise terms that talk about the same general phenomenon. According to theory on this area, the meaning of a word partly depends on its relation to other words in the same conceptual area.

cover a wide spectrum of densities (all above 0.5, of course) and F-scores, both high and low quality clusters (where quality is measured according to the category hierarchy) have probabilities of being selected. We decided to take 200 groups, as this number is already significant.

A very important aspect that we need to discuss at this point is the *degree of background knowledge* that an individual cluster requires. Because users are not expected to be experts at all on the theme represented by the document set of a given cluster, we need to assure that the groups we select do not demand a high level of expertise to be understood; in other words, they need to be *self-explaining*. Such issue is one of the most delicate facets on this kind of tests, specially since topics in Wikipedia can be very narrow, very broad, or something in between. Even though this could be a dead end for carrying out topic-related user tests (or at least without a gigantic amount of effort), fortunately we have three advantages on our side, which are exemplified in Table 5.12:

A variety of our extracted topics become self-explaining when visualizing their titles together. This is actually a great benefit that comes from of finding communitarian-structure clusters.

Several of the topics contain self-descriptive titles (i.e., for disambiguation purposes). On one side, this is totally independent from our approach, because our construction algorithm never employs titles for clustering. On the other side, this feature keeps us from having to introduce additional information for the users to get an idea of what an article in particular is about.

Some topics refer to general knowledge areas, such as elementary education subjects or popular entities (artists, TV shows, famous events, etc.).

Table 5.12: Self-explanation in Wikipedia topics

Collective explanation	**Descriptive titles**	**General knowledge**
BMW	Digamma	Plant
BMW 3 Series	Nu (letter)	Flowering plant
BMW 5 Series	Mu (letter)	Dicotyledon
BMW E30	Eta (letter)	Tree
BMW 7 Series	Alpha (letter)	Leaf
BMW 6 Series	Sigma (letter)	Flower
BMW Z3	Tau	Fruit
BMW E46	Chi (letter)	Seed
BMW 8 Series	Psi (letter)	Rosales
BMW E39	Epsilon	Botany
⋮	Lambda	⋮
	⋮	

Therefore, the presence of these cluster features turns our user experiments into something realizable, as we can assure that a considerable number of topics require a very basic level of knowledge. However, unluckily, having self-explaining clusters also implies that we have groups that do not count with sufficient self-contained information and/or are so specific that only a domain expert could effectively distinguish their elements. In case these clusters are selected for the test set (which is highly probable), we basically have three options: 1) provide extra article information for the users to access (the articles' text or a hyperlink to their Wikipedia entry), 2) expect the users to be proactive enough to look for this reference information (which is totally valid), or 3) dismiss these clusters. Because the first option is not practical—or even feasible—for a physical setup (e.g. paper tests), it does not seem fair to put an extra burden on users (who probably share more of a passive attitude), and, summing up, we rather opt for a "stand-alone" experimental framework, the most viable choice seems to be discarding non-explicative clusters (unknown topics). Nevertheless, we are perfectly aware that going for this option inserts bias to some extent. In order to prevent it and keep the test set as neutral as possible, an alternative is to typify the kinds of clusters to eliminate from consideration. By observing several document sets, we can notice that the most conflictive class concerns groups purely conformed by English proper names, particularly groups of *regional subdivisions* (counties, suburbs, etc.) and *person names* (e.g. senators, governors, first ladies, ...).

So, clusters were chosen at random, discarding the typified non-explicative ones; group id's continued to be selected by chance until the test set of 200 elements was complete. Another relevant datum concerns the number of representative documents; for achieving a significant sample, the top 30 articles were selected for each cluster. From these, 25 were left for presenting actual topic members (left column) and the rest (5) for the test list (right column); the latter were chosen by chance. With regard to the outlier set for each experiment, the procedure was highly randomized as well. First, two clusters were selected at random; if the id of the first one was an even number, two documents were chosen (also by chance) from this cluster and three from the second cluster. On the contrary, three were selected from the first and two from the second. Logically, we validated that neither of the outlier-source clusters was the same as the one being tested. However, there was no restriction for these two clusters being equal with respect to each other (in an odd case, all outliers could have been obtained from the same cluster).

An additional detail that we have to care about in our setup concerns *user reliability*; assuming that all users will have good intentions is simply naive (specially with the Internet platform that we will describe later), and this orients us towards configuring a mechanism for detecting careless answers—for instance, selecting all the elements of the test list. Actually, using this kind of mechanisms is not uncommon in fields such as marketing; surveys, for example, contain redundant questions (written with a different style to go unnoticed, of course), and if a person replies with different or contradictory answers on those questions, that survey is dismissed. In our case, a simple way of perceiving this behavior is by copying a couple of topic members into the test list; so, if a user is being clumsy, it is likely that he will fall on this trap (in other words, the probability of selecting everything but those two copied members while not paying attention is very low). Another advantage of using this mechanism, is that it allows us to be able to discard those tests where the user seems to be analyzing just a part of the topic members or the tag alone (which should act more as an aid). Above this, another way of validating that the user is committed to the task is by only accepting tests that count with a minimum number of correct answers, as we are assuming that the exercises are not extremely difficult. To avoid biasing our results towards choosing only "good" evaluations, however, the minimum number of correct answers was set to *one*.

In that sense, for every experiment (note that we use "experiment", "test", and "evaluation" as interchangeable terms), the test list consisted of 12 elements: 7 actual members of the topic (2 of them being "tricks" for the user) and 5 outliers. Logically, the test list was scrambled and the order of the elements was unpredictable; to scramble the list, a series of random numbers was obtained and put in front of the elements. The numbers were then sorted, and the elements were placed according to their corresponding number; for some experiments, the sort was in ascending order, and for others, in descending order. For a certain amount of tests, instead of following this procedure, the test list was sorted in alphabetical order, although we tried to do this rarely, as in several topics all the articles start with the same letter and this could be inconvenient for the final configuration of the test list.

After having completed the experimental setup with regard to test design, let us describe how users were gathered for performing the evaluations. An initial remark consists of the *utilized media*; with regard to this aspect, we prepared physical tests (on paper) and electronic tests. However, it may seem more correct to categorize the tests by the way users were approached, as a first (complete) set was separated for *directly* contacted users, and a second one (equal to the first one) was designated for users addressed via *Amazon's Mechanical Turk*.

Direct contact users

For the first test set, the subjects were mainly students from the Tec de Monterrey University and acquaintances from abroad (USA, specifically, as we were searching for English speakers). The students were mostly graduates (engineers, medical doctors, and PhD colleagues), and some of them went into the same class; these users were handed paper tests. With respect to people abroad, the tests were sent by e-mail. The format for these tests, shown in Table 5.13 was slightly different for the on-line version.

Table 5.13: Format for paper and e-mail tests

q139: nobility; titles; peerage; count; earl	
1. Nobility	a. Series (mathematics)
2. Count	b. Aristocracy
3. Baron	c. Viscount
4. Viscount	d. Rationalis
5. Marquess	e. Baron
6. Earl	f. Conservation designation
7. Gentry	g. Radius of convergence
8. Courtesy title	h. Peerage
9. Graf	i. Area of Outstanding Natural Beauty
⋮	j. Chinese nobility
	k. National Nature Reserve
	l. Comes

The instructions provided to the users were the following:

(1) *These tests have been designed to be answered even by non-expert users.*

(2) *The idea is to identify those elements from the RIGHT column (letters) that DO NOT belong to the left one (numbers); that is, to find the "intruders" or "outliers".*

(3) *Highlighted with a different color are five keywords that attempt to describe the topic of the left column. They may be used as an aid.*

(4a) *The outliers can be indicated by coloring their cell.* (Electronic format.)

(4b) *The outliers can be indicated by circling, underlining, or putting an "x" beside.* (Paper format.)

Even when the instructions were designed to explain the task in such a way that the subjects would not require additional assistance, users were not banned from expressing their doubts (although this situation was unusual), specially when having a face-to-face contact. Another important aspect is that users were usually assigned seven tests, on average, and did not have a specific time limit to fulfill these tests. However, the whole assignment generally was completed within 20 minutes or less. Assessing this kind of facts actually leads us to discuss the other aim, besides evaluating results, of paper and e-mail tests: acting as a preliminary step before launching the test set on-line (which was specially important because we had to pay for the Amazon tests). Concerning this, we wanted to observe if valid evaluations were generated with people whose academic and English level is known, and who could be considered as a bit more trustworthy than users we basically do not have information about. This would let us notice if major changes or enhancements had to be made to the design in order to have a better response (fortunately, this was not the case).

As a closing remark, it is important to highlight that even when several of our direct contacts were acquaintances or colleagues, it was not possible to "cheat" with their answers in some way, simply because these persons did not have in any moment the form of accessing our answer key, and we did not have enough human memory to retain it either.

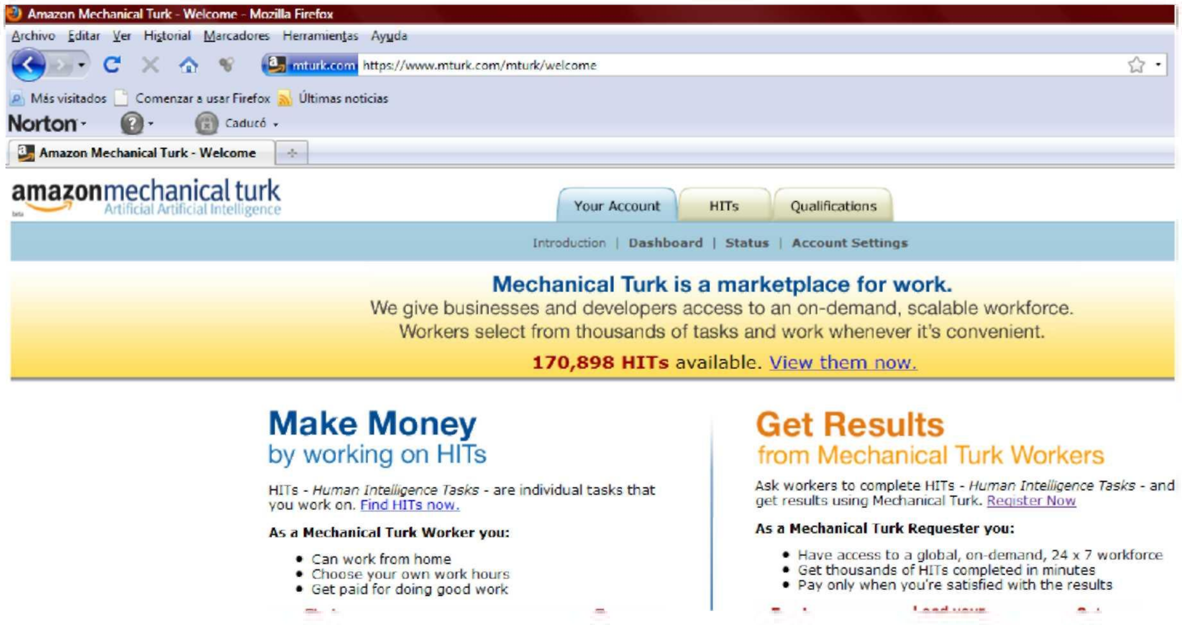
AMAZON'S MECHANICAL TURK (MTURK)

To counter-balance evaluation results gathered with the first kind of users, we considered applying our test set via *Amazon's Mechanical Turk*[3]. This platform simulates an "on-line marketplace for work" that connects job *requesters* with *workers* by letting the former to upload certain tasks (usually denominated "HIT's" for *Human Intelligence Tasks*) that are executed by the latter. Of course, both the workers and the broker (Amazon) are paid for their services; their earning is imposed by the requester, who also has the right of rejecting a task instance when considering it was not well done.

It seems convenient to go deeper into several advantages of Mechanical Turk:

Trustable platform. Being powered by Amazon gives *MTurk* a high degree of reliability, since this company is a serious business that performs a considerable number of on-line transactions per day. In that sense, MTurk concerns also a robust outsourcing option. Moreover, it has already been used for research purposes.

[3] `https://www.mturk.com/mturk/welcome`


Amazon Mechanical Turk - Welcome - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
mturk.com https://www.mturk.com/mturk/welcome
Más visitados Comenzar a usar Firefox Últimas noticias
Norton Caducó
Amazon Mechanical Turk - Welcome
amazonmechanical turk
Artificial Artificial Intelligence
Your Account
HITs
Qualifications
Introduction | Dashboard | Status | Account Settings

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.
170,898 HITs available. View them now.

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Get Results
from Mechanical Turk Workers

Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. Register Now

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Figure 5.8: Amazon's Mechanical Turk

Facilitates implementation. Because MTurk provides all the back-end (concurrency, host services, result collection, exceptions, etc.), a requester only has to worry about designing the HIT (in fact, a number of pre-defined templates is already provided), specifying the attributes of the task (description, how much time it will be available, how much will be paid for it, etc.), and providing the data in the correct format.

Batch works are allowed. Probably this is one of the greatest advantages, since MTurk is prepared to deal with tasks where the requester wants to create a series of webpages that all have the same design template but contain different data (which is our case). These batch works can be automatically generated by only specifying the data source they will feed from (a comma-separated file).

Results are manageable. Not only is it possible to configure the result webpage (filter, move field order), but also to download the answers to a batch of HIT's as a comma-separated file (*.csv*). This is extremely convenient for automatically analyzing results.

Worker reputation is handled. Every worker has a history according to the quality he has achieved with his HIT's, and is "graded" according to this history. This lets requesters to be able to specify a lower limit on the desired *worker approval rate* (e.g. "only allow workers with more than 95% of approval to answer the HIT").

However, "there's no such thing as a free lunch". Let us briefly enumerate several disadvantages:

The service has to be paid. Actually, this issue is more complicated than it seems. On one hand, it is difficult to calculate a fair price for a certain HIT, specially at the beginning; if the price is too low, workers will have no interest on answering the HIT, but if the price is high, the batch is more expensive and the requester becomes vulnerable for "predators" (people who do not care about the quality of their job, but only to earn money easily). Another delicate point is making sure that the uploaded batch is correct; if a requester makes a mistake on the design or specifications of the HIT, there are no refunds, as everything is pre-paid.

Tedious work rejection. Rejecting a job implies stating a genuine reason for denying that worker his pay; therefore, requesters must have adequate, objective validation mechanisms. Furthermore, there are time limits for approving or rejecting a task instance; if surpassed, the job is automatically paid.

Data has to comply with a specific format. Because MTurk is a third-party tool, experimental designs must cope with the way the platform works.

Our design for the MTurk's HIT can be appreciated in Figures 5.10 and 5.9 (this last one actually shows how the HIT appeared to workers). Moreover, Table 5.14 presents the settings for the batch of HIT's; let us note that the rubrics of *Title*, *Description*, and *Keywords* were also available for workers to, respectively, read and search the HIT batch. Regarding *Name*, this was an internal descriptor of the used template; on the other hand, the *Assignments per HIT* field is meant for specifying how many workers can be devoted to a single HIT instance, and the *Assignment expiration* field is used for indicating the total time the batch is to be available on-line.

Table 5.14: MTurk setup

Name	Outliers
Title	Select non-members
Description	Given a set of keywords and a list of members from a certain topic, pick from another list the ones that do not belong. For research purposes.
Keywords	topic, members, related, outliers
Batch size	200
Approval rate	greater than 95%
Time limit per HIT	25 min.
Assignments per HIT	1
Assignment expiration	8 days
Reward per assignment	$0.20

Results and discussion

A summary for the applied user tests is presented in Table 5.15. Almost all the tests were valid, and results were similar for both the direct and MTurk sets. As we can notice also in the graphs of precision vs. recall and cumulative F-score (Figures 5.11 and 5.12, respectively), the average quality is considerably higher.

Because the majority of the tests achieved a good individual score, it is not trivial to assess which kinds of topics were the most difficult to recognize. However, by examining

Table 5.15: User test results

PAPER AND E-MAIL EXPERIMENTS

Answered - 166		Valid - 151 (91%)	
	Precision	**Recall**	**F-score**
Min.	0.25	0.2	0.22
Max.	1.0	1.0	1.0
Avg.	0.92	0.94	***0.93***

MTURK EXPERIMENTS

Answered - 200		Valid - 175 (88%)	
	Precision	**Recall**	**F-score**
Min.	0.2	0.2	0.2
Max.	1.0	1.0	1.0
Avg.	0.93	0.9	***0.91***

OVERALL RESULTS

Answered - 366		Valid - 327 (89%)	
	Precision	**Recall**	**F-score**
Min.	0.2	0.2	0.2
Max.	1.0	1.0	1.0
Avg.	0.92	0.93	***0.92***

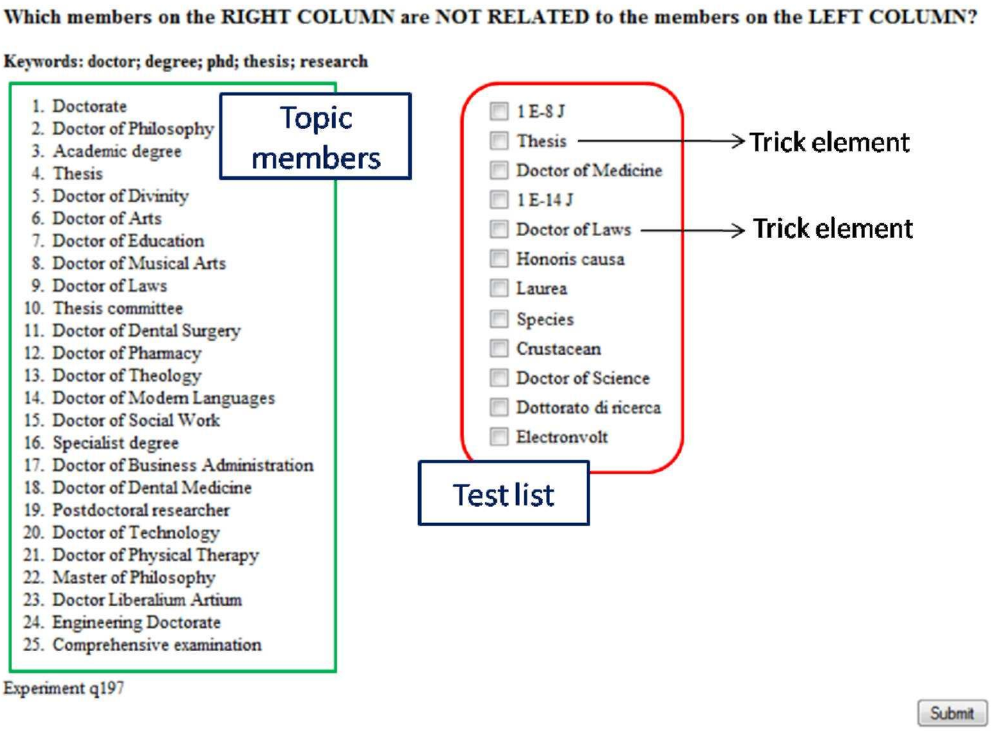
the lowest scores, we noticed that both direct-contact and Amazon users failed on test #27 (depicted in Table 5.16), as the obtained F-score was of 0.2. The former was due to a very simple reason: by chance, titles from a similar cluster were placed as intruders. In that sense, it becomes hard for a person who is not expert on the theme to discriminate its members without mistake. In fact, topics involving biological species—such as the one just mentioned—were slightly harder for users; probably this is because more knowledge is required on these themes. Other experiments that apparently placed more difficulty on users as well included themes that lie farther from common knowledge, e.g. knot theory and foreign countries.

Table 5.16: The hardest test for users

q27 **species; rana; style; birds; frogs**

1. Animal
2. Chordate
3. Bird
4. Binomial nomenclature
5. Amphibian
6. Ciconiiformes
7. Reptile
8. Fish
9. Ardeidae
10. Frog
11. Species
12. Carolus Linnaeus
13. Genus
14. Vertebrate
15. Egg (biology)
16. Ibis
17. Family (biology)
18. Neobatrachia
19. Invertebrate
20. International Code of Botanical Nomenclature
21. Feather
22. Threskiornithidae
23. Taxon
24. Flamingo
25. Pelecaniformes

a. Amphibian
b. Stork
c. Heron (disambiguation)
d. Euphorbioideae [X]
e. International Code of Zoological Nomenclature
f. Family (biology) [X]
g. Scientific classification
h. Crater [X]
i. Anura
j. Rille [X]
k. Ferdinand Albin Pax [X]
l. Fish



Which members on the RIGHT COLUMN are NOT RELATED to the members on the LEFT COLUMN?

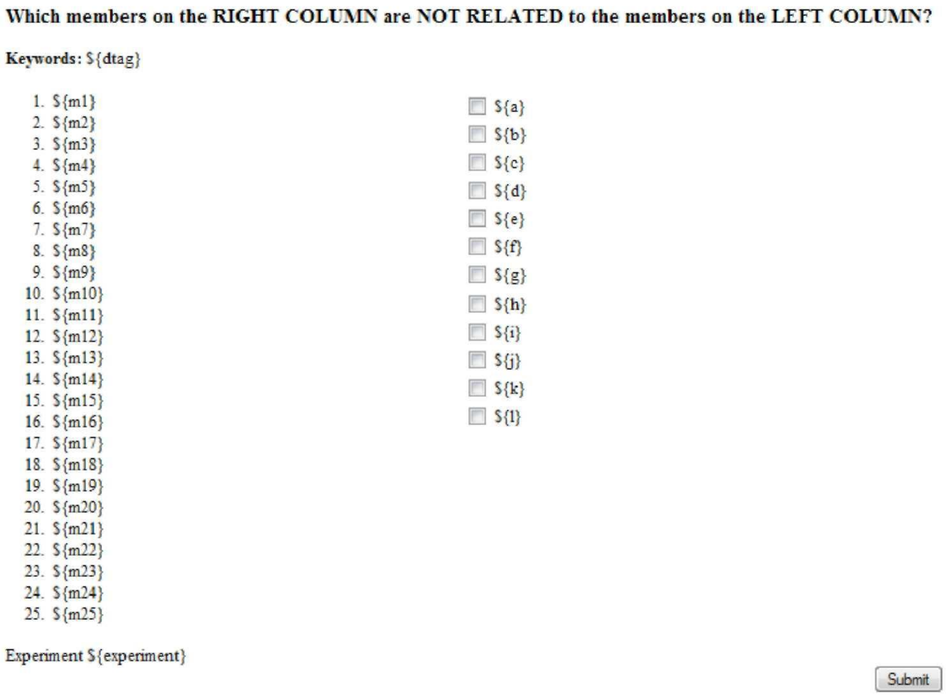
Keywords: doctor; degree; phd; thesis; research

1. Doctorate
2. Doctor of Philosophy
3. Academic degree
4. Thesis
5. Doctor of Divinity
6. Doctor of Arts
7. Doctor of Education
8. Doctor of Musical Arts
9. Doctor of Laws
10. Thesis committee
11. Doctor of Dental Surgery
12. Doctor of Pharmacy
13. Doctor of Theology
14. Doctor of Modern Languages
15. Doctor of Social Work
16. Specialist degree
17. Doctor of Business Administration
18. Doctor of Dental Medicine
19. Postdoctoral researcher
20. Doctor of Technology
21. Doctor of Physical Therapy
22. Master of Philosophy
23. Doctor Liberalium Artium
24. Engineering Doctorate
25. Comprehensive examination

□ 1 E-8 J
□ Thesis
□ Doctor of Medicine
□ 1 E-14 J
□ Doctor of Laws
□ Honoris causa
□ Laurea
□ Species
□ Crustacean
□ Doctor of Science
□ Dottorato di ricerca
□ Electronvolt

Experiment q197

Figure 5.9: HIT example



Which members on the RIGHT COLUMN are NOT RELATED to the members on the LEFT COLUMN?

Keywords: ${dtag}

1. ${m1}
2. ${m2}
3. ${m3}
4. ${m4}
5. ${m5}
6. ${m6}
7. ${m7}
8. ${m8}
9. ${m9}
10. ${m10}
11. ${m11}
12. ${m12}
13. ${m13}
14. ${m14}
15. ${m15}
16. ${m16}
17. ${m17}
18. ${m18}
19. ${m19}
20. ${m20}
21. ${m21}
22. ${m22}
23. ${m23}
24. ${m24}
25. ${m25}

□ ${a}
□ ${b}
□ ${c}
□ ${d}
□ ${e}
□ ${f}
□ ${g}
□ ${h}
□ ${i}
□ ${j}
□ ${k}
□ ${l}

Experiment ${experiment}

Figure 5.10: HIT design

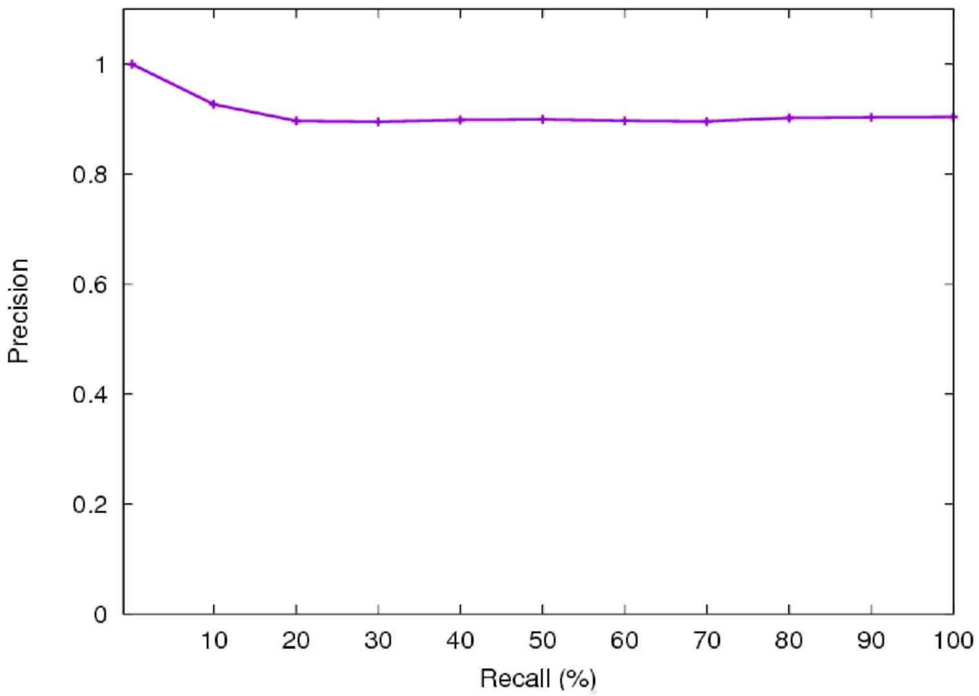


Figure 5.11: Precision vs. recall curve for user tests

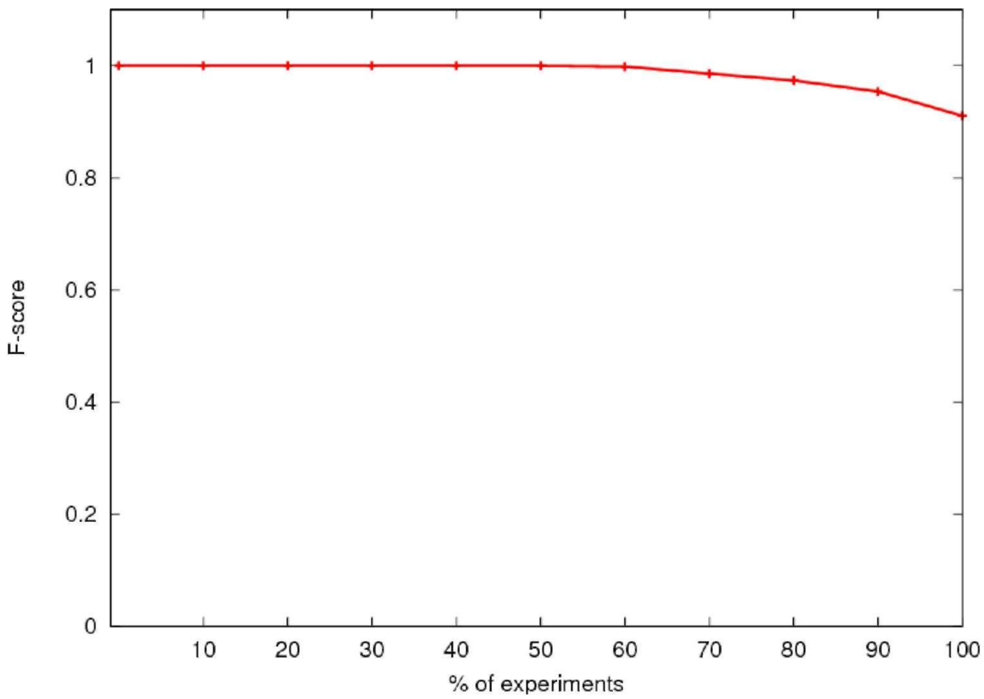


Figure 5.12: F-score curve for user tests (sorted in descending order)

Another point of discussion is provided by *topic properties*; even though these were not concretely evaluated, the quality of the obtained results subtly suggests that representing topics by means of their properties was not prejudicial. Nevertheless, in order to assess the actual summarization capability of topic properties, a respective evaluation of these would have to be carried out, perhaps with more specific tests. This kind of validation so far lies out of our current scope.

Overall results from user tests come to "round up" our series of validation experiments by providing strong evidence of the topical coherence of our discovered document groups. In that sense, we believe human judgment is a valuable tool that can certainly be utilized on this domain, as persons count with general background knowledge and a common sense that is able of clearly identifying the semantics of a given document conglomeration.

5.4 General discussion and sub-task summary

Although the cohesion revealed by internal validation metrics is very notorious (specially with the Jaccard index and the relative strength ratio), we believe that result quality is better appreciated with external evaluation, and more with user tests. Probably the most outstanding outcome of the alignment with the category tree is the perfect correspondence that a number of our clusters show with respect to categories; furthermore, it is interesting to note how in some cases, (different) clusters match a category and also one (or more) of its sub-categories. We have discovered that this happens because several groups contain others (which in turn gives insight about the possibility of finding a hierarchy). On the other hand, the user scores unmistakably confirm the semantics of our discovered groups; an additional interesting fact is that, despite background—and perhaps cultural—differences, results were very similar both for paper and electronic user tests.

Because Table 5.17 already presents a brief summary of our validation schemes, there are only a couple of points left for further discussion. First, our main finding is that, *from different perspectives (intrinsic features, external references, judgment), our extracted groups consist of topics*. Second, without failing to recognize that coverage is indeed important, we have been more committed towards highlighting *quality* in our results. Finally, with a large-size collection, we have observed that complexity an inherent issue—even for the validation sub-task.

5.5 Chapter summary

With respect to topic validation, the construction approach was finally carried out over the Wikipedia corpus to produce a clustering, which was evaluated internally and externally. Internal validation, on one hand, assessed intra-cluster *compactness* vs. inter-cluster *separation* on our filtered results; also, an *SNA cohesion metric* whose satisfactory threshold is known has been applied. Regarding external evaluation, an *alignment* against Wikipedia's category tree was executed, and accuracy-based measurements (precision, recall, F-score) were obtained to assess quality, principally. Moreover, *intrusion detection* tests were applied to users and quantified with accuracy-related metrics as well. *Our results showed to support the main hypothesis: the discovered groups are topics.*

Because validation closes the “cycle” of topic extraction and contains the final elements of the present work, we are ready to give concluding remarks and discuss future work. However, it would be important to first discuss more thoroughly related work in order to make clear the differences and original components of our approach.

Table 5.17: Main aspects of the validation sub-task.

<table>
<tr><td colspan="2"><b>Validation.-</b> <i>Consists of proving that the extracted topics are coherent.</i></td></tr>
<tr><td><b>Key question</b></td><td><b>Related aspect</b></td></tr>
<tr><td><i>What type of evaluation is going to be carried out?</i></td><td><i>Internal validation</i>: cluster compliance and golden threshold
<i>External validation</i>: alignment to Wikipedia's category hierarchy and user tests (human judgement)</td></tr>
<tr><td><i>How is the topical coherence going to be proved with the chosen evaluation type?</i></td><td><b>Internal evaluation</b> Orthogonal (text-based) cluster compliance metrics show a semantic relationship among elements, and this establishes a topical bondage. The other metrics (parallel and group quality) prove group coherence in general and <i>suggest</i> topical coherence.
<b>Alignment</b> Topicality is shown by measuring the resemblance between a given cluster and its matching category (external reference).
<b>User tests</b> Topical coherence is judged by a common person.</td></tr>
<tr><td><i>What metrics are going to be used?</i></td><td>1. Cluster compliance
a. <i>Similarity</i>: cosine similarity and Jaccard index (document pairs)
b. <i>Dissimilarity</i>: semantic relatedness
2. Golden threshold
a. Relative strength ratio (SNA)
3. Alignment
a. <i>Quality</i>: precision, recall, F-score
b. <i>Coverage</i>: percentage of clustered documents, percentage of matching categories
4. User tests
a. Precision, recall, and F-score</td></tr>
</table>

Chapter 6

Related Work

The main focus of the current chapter is to *discuss related work*. With this target in mind, it is necessary to first situate our approach within the state of the art. After doing such classification, related approaches—placed according to their degree of resemblance—are to be explained (with fairly more detail than in Chapter 2) in order to highlight similarities and differences with respect to our method.

6.1 Classification of our approach

To be able to have a better perspective on how our work differs from similar approaches found on literature, it is precise to classify it according to the taxonomies we have defined for the state of the art. Let us present this classification (also summarized in Table 6.1).

With respect to Web structure mining, our approach is cut-based (link count) because it works around this notion and specifically tackles the construction problem by keeping track of internal and external links. Furthermore, because it employs density, the discovered groups are cohesive; as a matter of fact, this group type seems to represent a special kind of topic (this is to be retaken in the general conclusions). In addition, since we are using a graph-theoretic method with local search to construct our clusters, the mining sub-task corresponds to network clustering (note that this is depicted even from the name of the algorithm, GLC) and complexity is addressed by problem decomposition.

Regarding topic mining, the used representation is object-oriented, as our topics consist of two document lists (the actual document cluster and the subset of representative documents) and a label (tag); consequently, the approach comes to be hybrid (flow), as it carries out enumeration followed by labeling and a soft form of distillation.

On the Wikipedia rubric, our paradigm is soft because we do not recur to the use of ontologies; moreover, although text is employed for a part of description, the method can be seen as mainly structural. Finally, Wikipedia is considered both as a source and a destination for our results (we will discuss applications on the next chapter).

Table 6.1: Approach classification

WEB STRUCTURE MINING	
Approach type	Cut-based, link count approach
Group type	Cohesive groups
Complexity management	Problem decomposition (mainly local search)
Sub-task	Network (graph clustering)
TOPIC MINING	
Representation	Object-oriented (enumeration + label)
Approach type	Hybrid
Sub-task	Flow: Enumeration → labeling, distillation
WIKIPEDIA MINING	
Semantic Web paradigm	Soft
Information source	Structure
Use	Source and destination

6.2 Related approaches

Now that we have described our work in terms of the classification used for the state of the art, it is possible to establish comparisons between our approach and other existing methods in literature. Consequently, the current section is devoted to explaining with more detail those works that we consider as the most related ones to ours with the intent of highlighting similarities and differences. The ultimate aim is, then, to elucidate the originality of our approach and how it occupies a particular place within the state of the art. With respect to related work, to facilitate comprehension, we have divided it into two main categories: specifically related and generally related. The first category attempts to show the closest works (either in domain or technical aspects), while the second mainly intends to enumerate various complementary methods and seminal works that we believe should not be left out.

6.2.1 Specifically related

The following works are being considered as the ones that have a higher resemblance to ours. In that sense, they either carry out the same topic task and solve it with hyperlink information or use the same approach in a different domain.

Topic + Web Structure Mining

CLUSTERING HYPERTEXT WITH APPLICATIONS TO WEB SEARCHING – Modha and Spangler, 2003

This method comprises hypertext clustering based on a combination of textual and structural information, and could ultimately be treated as a topic mining (enumeration, labeling, distillation) approach, although the topic domain remains implicit. Three are the pillars of such approach: a hybrid similarity metric, a novel variant of k-means, and the inclusion of properties (information "nuggets") into the clustering process. It first starts by introducing a *query* into a search engine (AltaVista) and collecting the 200 best ranked webpages; each document is then treated as a *triple* that consists of three feature vectors: words, in-links, and out-links. For the in-links, the top 20 documents that *cite* the query-result set are taken into account. Similarity between pairs of triplets is, thus, calculated by a *weighted sum* of the three features, each feature being affected by an *importance factor* or weight (this last part is important and will be discussed afterwards). Furthermore, because documents are viewed as lying in a space of three spheres (a torus), a *toric* k-means algorithm is proposed and used for clustering; this algorithm actually follows the same approach of k-means, while adapting to the defined geometric space. The algorithm begins by taking random *concept vector triplets*, which act as the centroids of simple k-means, and aims to maximize a *coherence* measurement over the clustering, which could analogously be seen as the squared error (SSE) that tries to be minimized on the conventional version. Similarly, at each iteration all documents are placed in the cluster of the closest concept triplet, and this is followed by the re-calculation of these triplets; these steps are repeated until the change in coherence is below a threshold value.

Another fundamental part of the approach is cluster *annotation* (property calculation), which is motivated as the clusters are considered of "little use" in their raw form. As a result, six information nuggets are proposed (two according to text, two according to in-links, and the other two according to out-links): *summary, review, breakthrough, keywords, references, and citations*. While the summary, breakthrough, and review properties are *descriptive* and consist of the most representative word, in-link, and out-link feature vectors (in that order), the keywords, citations, and references properties are *discriminative* and present the individual words, out-links, and in-links with the largest weights on the cluster, respectively. To ensure good nuggets, the importance factors of the similarity weighted sum are tuned in order to maximize the combined feature quality.

The approach was tested with four queries of multiple senses, acknowledging that an appropriate clustering algorithm should be able to create different clusters for each of the distinct senses; the queries were "latex", "abduction", "guinea", and "abortion". An intuitive validation ("proof in the pudding") is done by presenting these nuggets for the reader to judge. Future directions suggest *iterative hill-climbing* for carrying out maximization of coherence and optimal weights for the features, trying other clustering approaches (e.g. hierarchical or graph-based), and recognize computational complexity issues, basically.

Although the conception of gathering and describing groups of Web documents is very aligned to our topic extraction process, there are some major key differences between the approach of Modha and Spangler and ours. To start with, their method performs data (similarity) clustering with a modified version of a classical algorithm; furthermore, the similarity metric employed by the clustering technique is hybrid, while our clustering technique is pure, in the sense that it only uses hyperlinks to conform the groups. On the other hand, our method does not part from a broad query, which finally serves to create a sub-collection of manageable size; even though our seeds accomplish a similar function by letting us to concentrate only on a certain region of the collection at a time, the whole corpus is eventually explored with the intent of extracting all the cohesive topics. Another notable difference is given by properties; besides being more than the ones we have defined so far (note, however, that both theirs and ours include text and hyperlink metadata), a very interesting aspect of Modha's approach is that the properties are actually binded to the clustering process by the importance factors of the similarity measure; as for our case, we treat properties independently from cluster construction. Furthermore, even when this related approach does not intend to provide a formal framework, documents and cluster centroids are formally defined.

AUTOMATIC TOPIC IDENTIFICATION USING WEBPAGE CLUSTERING – He et al., 2001

Resembling some aspects of the previous work, this approach does topic extraction (enumeration and distillation) by clustering webpages with a spectral method that employs a hybrid similarity metric; the central aim is to enhance information organization by presenting a list of the authoritative webpages given a user information request. Consequently, the process starts by collecting and expanding the top 100 results from a broad query to subsequently place them into a similarity matrix, where similarity is a weighted sum between *adjacency* and *co-citation* information; however, the adjacency component is affected by *cosine similarity*, a textual factor. So, if two pages share a link but do not resemble each other regarding text, their overall similarity is smaller. This matrix is then passed as input to a hierarchical divisive clustering algorithm that consists of a bisecting spectral graph partition method based on normalized cuts (the method is adopted from the image segmentation context); the algorithm splits the initial group in half and recursively divides the resulting halves until a stopping criterion is met. After gathering the groups, the HITS algorithm is applied on the clusters to calculate hub and authority scores; the top authorities are presented as result.

The approach was tested with a set of three one-word queries (“star”, “amazon”, “apple”), which were chosen precisely for their multiple senses. For each query, 2000˜3,500 documents were retrieved, and seven clusters on average were obtained.

Because this work is indeed very similar to the one discussed before, let us note that several of the major differences mentioned above still hold (e.g. parting from a broad query, a hybrid similarity metric). Also, this method—which is explicitly recognized as topic identification—is not concerned with the labeling task, and distillation is specifically done using HITS. Regarding valuable aspects that may well be mentioned, on one hand, although a divisive spectral approach is used, a partial clustering is considered by arguing that some documents may not fit into a cluster at all. On the other hand, the clustering technique was taken and adapted from a different domain.

Web structure mining with GLC

Clustering the Chilean Web – Virtanen, 2003

This work, whose main goal is to validate graph generative models and gain insight on the cluster formation process of the Web, is the one that introduces the GLC method [140]. The general motivation for such approach is placed upon pointing out that global approaches do not scale well for large graphs, mainly because calculations over the adjacency matrix of such structures is computationally expensive (if it were possible to build the matrix on the first place). In that sense, a graph local clustering approach that combines simulated annealing with a fitness function composed of local and relative density is presented and applied over the largest connected component of the Chilean Web to produce a flat, partitional grouping of pages. Specifically, the algorithm starts with a single-vertex seed and considers a neighborhood of size 10; at each construction step a node may be either added or removed from the cluster (250 modifications being allowed per cluster). The procedure is repeated 30 times per every initial node, and the best (densest) cluster is selected for the final partition. Those pages that already belong to one of the groups are prevented from being clustered again; furthermore, the removal strategy does not delete the seed vertex in any case. It is interesting to note that a closing remark consists of extending the approach to community discovery and topic extraction.

The main difference between this work and ours lies on the orientation of the approach; in that sense, it is important to acknowledge Virtanen’s work as the pioneer for the GLC method and the approach in general, while stressing that we employ such approach to the topic extraction domain. Consequently, our approach is bent towards topic discovery and focuses on particular aspects of this discipline, such as definition, description, and proper validation. This domain difference marks several contrasts (technical, basically) between the specific clustering algorithms, such as particular enhancements and selection of component combination, i.e., the local search strategy and fitness function.

6.2.2 Generally related

These approaches can be considered as standing on the periphery of the area where our approach specifically lies; in that sense, they either perform a topic task that does not involve enumeration, use a different type of information, and/or represent seminal works of the knowledge area (structure mining) that it is valuable to discuss.

Topic + Wikipedia mining

TOPIC DETECTION BY CLUSTERING KEYWORDS – Wartena and Brussee, 2008

Topic modeling by grouping keywords with a novel distance metric based on the Jensen-Shannon divergence (which originally calculates the similarity between a pair of probability distributions) is the main contribution of this work. As normal with data clustering methods, the approach consists of a similarity metric and an algorithm that creates groups based on this metric. Similarity is actually stated in terms of distance, since this complementary measurement is calculated between distributions that are related to keyword co-occurrence in documents; on the other hand, a variant of bisecting k-means is used for creating a binary tree of clusters, whose centroids (that represent the average distribution of the co-occurrence distributions) are considered as the topics of the collection.

To test the approach, a set of 758 articles from eight categories of the Dutch Wikipedia were used for grouping. From the more than 100,000 words that these documents contained, the 160 most important ones were selected by using the Kullback-Liebler divergence and preprocessing operations, such as lemmatization, stemming, relevant part-of-speech discrimination (nouns, verbs). With this word base, four different metrics were used for clustering: cosine with document distributions, Jensen-Shannon divergence with term distributions, Jensen-Shannon divergence with document distributions, and cosine with tf-idf. To validate, the F-score obtained by comparing the resulting clusters against their corresponding Wikipedia categories was measured; the proposed metric (Jensen-Shannon divergence with term distributions) showed to be superior. Notable remarks about expensive computations, however, are stated, since the information vectors are very large (not surprising to believe, as Wikipedia contains a vast vocabulary even for a small amount of documents) and sparse.

Despite the fact that this work is related to Wikipedia, this work—we believe—does not lie as close as the previously discussed ones for several reasons. On one hand, the topic subtask here concerns modeling, which is not so similar to enumeration (an accute dissimilarity is also that they cluster cluster keywords, while we cluster documents); furthermore, on the contrary of our method, the work of Wartena and Brussee is completely based on text. Of course, while this work is also based on data clustering, more differences are added (e.g. the use of a similarity measure and the identification of common-trait groups). Nevertheless, it is important to keep in mind that, finally, this approach and ours have the common goal of extracting the topics of a collection, and this enforces the consideration of specific circumstances (e.g. how to represent the topic). Moreover, both methods are developed under the same corpus, and this makes them comparable to some extent.

TOPIC IDENTIFICATION USING WIKIPEDIA'S CATEGORY NETWORK – Schönhofen, 2006

This work corresponds to topic labeling with the aid of the on-line encyclopedia's base of categories; in that sense, the aim is to assign labels from a fixed set to a group of documents. For this case, the fixed label set is represented by categories from Wikipedia, as this corpus is considered as a taxonomy of wide coverage. To carry out such assignment, the text from a document is mapped first to a Wikipedia title; this title logically belongs to one or more categories, and the best matching category is finally selected according certain factors, like element (article, word, category, title) relative importance and the degree to which a title supports each of its categories.

To test the approach, the Wikipedia 2006 version corpus ($\approx$ 900,000 articles) was preprocessed and used for constituting the label set. Two types of validation experiments were performed; the first one consisted of applying the method directly with articles of the famous encyclopedic knowledge collection (because the text of Wiki articles does not form part of the approach, such tests are valid) and then compare the obtained categories against the "official" categories of the articles. These results were measured in terms of terms of coverage, that is, number of matching categories; for almost half of the pages, all the official categories were listed by the method. Moreover, by additionally taking into account super and sub categories at most two levels deep, these results slightly improved. The other experiment consisted of separately clustering (with CLUTO[1] default parameters) and categorizing (with naive Bayes) 20,000 documents from newsgroups and the Reuters RCV1 corpus; the groupings were made using four different representations, namely, 1) full-text, 2) top 20 words according to tf-idf, 3) resultant Wikipedia categories, and 4) a combination of the last two (tf-idf + categories). While the Wikipedia category representation actually earned some of the lowest scores, the combination surpassed all representations.

Even when the title could be misleading, this approach (such as the one previously discussed) shares some important differences when compared to ours. First, here the topic sub-task is labeling alone; therefore, the method on the inside is completely distinct, specially as it relies totally on text. A third difference is that Wikipedia seems to be used solely as an information source; although validation was carried out using this corpus also, the ultimate application appears to be on other corpora. However, the final intention of the approach is to map documents to topics, and this is also what we are looking for; handling Wikipedia implies common steps as well (e.g. treating redirections and managing the category network, whose *quasi*-hierarchical structure is also noted by the authors).

Seminal work

AUTHORITATIVE SOURCES IN A HYPERLINKED ENVIRONMENT – Kleinberg, 1999

This work by Kleinberg is one of the main references for Web structure mining and topic distillation; as we have seen, it consists of providing the most important pages (authorities) and lists that contain them (hubs) given a user query [73]. Regarding the query, it is of the broad type, and, thus, represents a subject wide enough to be capable of returning a considerable number (thousands or even millions) of webpages when introduced into a search engine; therefore, this potential amount of results—assumed to be logically tiring for the user—requires a prioritization. Such process is carried out by the HITS algorithm. This method starts by taking the top n (where n is typically 200) results of the broad query in question to conform a *root set*; such set is then expanded further by including linked and linking pages, although the latter are treated under several constraints to prevent an excessive growth,

[1] Clustering Toolkit for high dimensional data.
Available at `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview`

as this ultimately incurs into computational costs. into the set The expanded root set is known as the *base set*. Once having this group of webpages, hub and authority scores are calculated; the basic idea behind this calculation is that a good hub points at many good authorities and good authorities are pointed by many good hubs. Even when this is a circular reference that apparently cannot be solved, the scores can be obtained by finding the principal eigenvectors of the pages adjacency matrix with the power iteration method. After a certain amount of iterations, hubs and authorities start to appear, as several webpages begin to show a tendency towards being one of these. A closing remark about the approach is that it is not only limited to topic distillation, but may also be used for other tasks, such as community discovery.

Undoubtedly, this work has opened a wide perspective on structure mining and has led to a range of other approaches. Furthermore, it presents several issues and procedures that are common when mining the Web. For example, the idea of starting with a seed and then expanding it (which is also done by Flake) has been widely adopted, and problems with topic drift and complexity are first introduced by HITS as well. Moreover, the approach is also related to the topic extraction domain. Nevertheless, in specific terms, this approach is fairly different from ours; in that sense, it performs other structure and topic mining sub-tasks (resource discovery and distillation, respectively) and is devoted to finding distinctive-feature groups, which differ from our clusters.

EFFICIENT IDENTIFICATION OF WEB COMMUNITIES – Flake et al., 2000

This method establishes the general definition of a community based on density and proposes a discovery method that relies on the theorem of maximum-flow / minimum-cut. Regarding the first aspect, a community is identified as a subgraph whose internal link density exceeds the density of connection to nodes outside it to some margin. With respect to the approach, it assumes that if a graph contains a is linked by source node and links to a target node, then each edge can be seen as a water pipe that has a certain maximum flow capacity. When the maximum rate of flow from the source to the target that does not exceed capacity is found, actually this is equivalent to calculating the minimum-capacity cut that separates the source from the target (minimum capacity that needs to be removed from the network so no flow can pass from one side to the other). To carry out community discovery on this form, first a seed set of URL's (community member examples) is provided; each element of this set is linked to the source with edges of infinite capacity. The rest of the vertices of the graph (i.e., excluding seed, source, target nodes) are then connected to the target with a capacity of 1. Every other edge is made undirected and assigned a capacity of k (heuristically determined). After forming this configuration, the max-flow algorithm is called and the nodes that remain on the side of the source are considered as the discovered community (let us note that all seed nodes are guaranteed to end up as part of this cohesive group). However, it is not trivial to have all the WWW at hand, so only nodes found at a certain depth level from the seed (e.g. two hops) are crawled. Furthermore, to prevent finding only a few community members as result, after the first iteration is complete, the nodes of largest degree can be used as seed for the second one, and so on until the community either becomes stable or reaches a given size.

To test the approach, communities that revolve around individuals and institutes were discovered. As a form of validation, text features from each community were extracted with the help of the Kullback-Leibler metric, and utilized separately as classifiers to observe if the same community (or a similar one) could be recovered.

The approach by Flake et al. has established an essential precedent on structure mining for dense group extraction; it, therefore, serves as a model for this kind of endeavor. With respect to our method, it is also founded on the community notion, and this constitutes a major point of resemblance; needless to say, both methods are graph-theoretic and find groups of cohesive, related webpages as well. However, the specific method (flow) is different and implies some modifications to the actual document graph; moreover, the domain of application is distinct, as these authors are more oriented towards social communities.

6.3 General discussion

Logically, every work—like the ones just described—tends to be unique. In fact, if every classification rubric (information type, task, complexity, management, etc.) would be considered as a dimension for a given space, it would not be difficult to find out that each approach represents a distinct point on this space. With respect to our work, it can be seen as situated within three general dimensions: structure, topic, and Wikipedia. This combination does not seem to be common. Moreover, certain particularities that we are able to observe more clearly by analyzing related work consist are the following:

- The whole collection is taken into account.
- Topic definition is an integral part of the approach; in general, we acknowledge topic extraction as a process that consists of four essential sub-tasks.
- Starting points are not necessarily part of the final clusters (we will go over this on the next chapter).

6.4 Chapter summary

Our work has been classified within the state of the art according to three discussed central axes: Web-based meaningful group detection, topic mining, and semantic information extraction in Wikipedia. Following the meaningful group detection taxonomy, the work can be considered as a *cut-based, link count approach* that finds cohesive groups via network clustering and uses problem decomposition (specifically, local search) to manage complexity. With respect to the topic mining taxonomy, our approach can be regarded as object-oriented (topics consist of a document enumeration and a label); consequently, it might be considered as *hybrid*, since it obeys a flow composed of *enumeration* followed by *labeling* and a soft form of *distillation*. With respect to Wikipedia mining, the approach is based on structure and a *soft* Semantic Web paradigm; also, it employs Wikipedia both as source and destination. As recently stated, the main differences in comparison to other works are three: taking the whole collection into account, conceiving topic extraction as an integral process, and a using a *corpus-based* method.

By having discussed the most important related approaches, we become able to close the present work with concluding remarks and future directions (next chapter).

Chapter 7

Conclusions and Future Work

The present chapter aims to give closure to the exposed approach. First, a *summary* of the topic extraction process is to be presented, along with a list of our main contributions. Afterwards, a set of concluding *remarks*, followed by several *lessons learned*, shall be discussed; similarly, we will provide an *answer* for each of the research questions stated at the introduction. Finally, directions for *future work* and possible applications are to be highlighted.

7.1 Overall summary

A process for extracting groups of topically related documents in encyclopedic knowledge Web collections by means of a pure hyperlink-based clustering approach has been presented throughout the previous chapters; Wikipedia was selected as an appropriate case study corpus for this environment. This process was designed and developed as to comply with four main axes or *topic sub-tasks*: definition, construction, description, and validation. These tasks were aligned to an overall *topic extraction conceptual model* (TCM) that views topics as document clusters whose semantics reveal a thematic bondage; the model is inspired by layered architectures and consists of four abstraction levels: data representation (physical and logical), clustering, semantics, and applications. Definition and a part of construction are carried out at the representation level, while at the clustering level construction is finally accomplished; validation is also accomplished at two levels, namely, clustering and semantics. Finally, description is done at the semantic level.

Regarding topic definition, an extraction *formal framework* that comprises the basic elements (document, corpus, topic) and operations (construction/basic function, clustering/extended function, property methods) of the process was introduced. This framework is graph-theoretic, since it is suited for a hyperlinked environment such as the Web.

With respect to topic construction, this sub-task was regarded as the most important one of the extraction process. A first aspect of construction is the *link information extraction* endeavor, as it prepares data for the clustering procedure; for this aspect, several options were compared and evaluated according to specific criteria. Our final choice was to use Wikiprep as our source: it enables an easy access to articles and links, anchor text (titles) is available, most irrelevant material is already discarded, and the text is neat.

It has also been mentioned that the bulk of construction—and the core of the whole extraction process–falls upon hyperlink-based document clustering. This clustering approach, in abstract terms, was stated to be heavily related to *community detection*, because it searches for highly inter-linked overlapping groups. Consequently, the method is *structure-based* (graph theoretic) and assumes that topics will tend to concentrate into *maximum-cohesion subgroups*, which can be visualized as the local optima (peaks) on a multidimensional surface. Furthermore, the corner stone of the approach was stated to rely on the *Graph Local Clustering* (GLC) approach; not only was this working approach selected because it complies with our general idea for tackling topic construction, but also for other additional advantages: an inherent ability to deal with complexity, conceptual clarity, adaptation to our (graph-theoretic) domain, and its theoretical foundations.

Another important point regarding construction that was described comprises the concrete method, which consists of two fundamental algorithms (consistent with our formal framework), namely the *construction and clustering functions*: the former extracts a single cluster from a seed document set and is used repeatedly by the latter to produce a grouping of the collection. Recalling that the two general parameters of the construction function imply a local search strategy and a cohesion fitness function, we chose to employ first choice *hill-climbing* and *relative density*, respectively. Such selection was based on the Ockham's razor principle: start with a simple base and add sophistication as needed. In that sense, we decided to explore seven possible *enhancements*: a removal strategy (finally included), candidate ordering (descending by prospective internal links), neighborhood type (partial), seed selection (at random), seed expansion (not included), secondary clusters (considered), and quantum addition. Inclusion and tuning of these refinements was carried out by observing the algorithms' behavior with (Wikipedia) sub-collections of smaller size.

For topic description, two main properties were considered: a representative document subset and a tag. *Degree centrality* was employed for generating a ranking of the cluster documents and being able to select the k most important ones. On the other hand, a *weighting scheme* based on text frequency was used for ranking words and generating the topic tag.

With respect to topic validation, the construction approach was finally carried out over the Wikipedia corpus to produce a clustering and evaluated internally and externally. Internal validation consisted, on one hand, of assessing intra-cluster *compactness* vs. inter-cluster *separation* on our filtered results; also, an *SNA cohesion metric* whose satisfactory threshold is known was applied. Regarding external evaluation, an *alignment* against Wikipedia's category tree was executed, and accuracy-based measurements (precision, recall, F-score) were obtained to assess quality, principally. Moreover, *intrusion detection* tests were applied to users and quantified with accuracy-related metrics as well. *Our results showed to support the main hypothesis: the discovered groups are topics.*

Finally, our work was classified within the state of the art according to three discussed central axes: Web-based meaningful group detection, topic mining, and semantic information extraction in Wikipedia. Following the meaningful group detection taxonomy, the work can be considered as a *cut-based, link count approach* that finds cohesive groups via network clustering and uses problem decomposition (specifically, local search) to manage complexity. With respect to the topic mining taxonomy, our approach can be regarded as object-oriented (topics consist of a document enumeration and a label); consequently, it might be considered as *hybrid*, since it obey a flow composed of *enumeration* followed by *labeling* and a soft form of *distillation*. With respect to Wikipedia semantic information extraction, the approach was declared to be based on structure and a *soft* Semantic Web paradigm; also, it was described as employing Wikipedia both as source and destination.

7.1.1 List of contributions

General contribution

The overall contribution, already stated at the beginning of our summary, consists of the following:

> *A process for extracting groups of topically related documents in encyclopedic knowledge Web collections by means of a pure hyperlink-based clustering approach.*

Particular contributions

From the general contribution, a number of more specific contributions can be listed:

- ▶ Concepts
    - ◇ Layered topic conceptual model
    - ◇ Topic extraction formal framework
    - ◇ Statement and characterization of four topic extraction sub-tasks: definition, construction, description, and validation
- ▶ Methods
    - ◇ Specific construction (GLC) approach
        - ▷ Removal strategy
        - ▷ Candidate ordering strategy
        - ▷ Scheduling strategy (seed ordering + quantum + dynamic seed list)
    - ◇ Set of options for topic evaluation and description
    - ◇ Evidence of the method's quality
- ▶ Products
    - ◇ Wikipedia set of cohesive topics
    - ◇ Reusable Wikipedia items: sub-collections, expanded category tree

7.2 Concluding remarks

After having reviewed our work briefly, it seems convenient to discuss some important remarks that may not be visible at a simple glance. With regard to these, the extraction mechanism actually:

Discovers meta-topics. If we consider that in an encyclopedic knowledge corpus every article by itself corresponds to a topic, we are then clustering similar topics into groups that belong to a more general thematic.

Discovers a specific class of topics. Although we part from the initial idea of being able to find all the topics of a collection regardless of whether they share or not certain traits (e.g., being dominant or weak), the truth is that our mechanism is instead focused on detecting a specific kind of topics: *communitarian topics*. First, like we had seen on previous chapters, not every group is necessarily cohesive, and cohesive to the degree of representing a community. Then, it is natural for an algorithm that searches for dense structures to find basically this class of topics; moreover, the algorithm not only aims to detect these structures, but also prefers them at certain points, e.g., at the time of removal (where the least cohesive part of a cluster is dismissed). Seeing all of this, we come to realize two things; on one hand, our work is similar to those topic mining approaches that consider complementary topic classes and concentrate on identifying a one of them ("hot", "prominent", "dominant", or "emergent" topics, for example). On the other hand, we come to acknowledge that, being this the case, coverage is truly a secondary aspect. Therefore, to properly approach it, the inherent behavior of the mechanism would have to be modified or complemented with another method.

Is corpus-centered. Summarizing the prior point, our specific extraction task can be defined as *detecting the cohesive topics given a collection*. Therefore, even when we employ certain features for trying to place every document inside a cluster (with an exhaustive seed list, for example), we are really not enforcing this condition to be met, because at the end documents can be removed from a group even if they were initially a part of its seed. As a result, our main intention would be not to find the topics of a *document*, but more precisely to find the topics of a *corpus*—being this entity our primary element. The counterpart, a *document-centered approach* is to be discussed with more detail later.

Is highly sensitive to structure. Because communitarian topics are the specialty of the mechanism, sensitiveness to structure is logical to some extent. However, we might add that the more community-like a cluster is, the less difficult it becomes for the mechanism to find; as a result, even a rough version of the algorithm would suffice to detect these topics. On the contrary, to correctly extract less cohesive clusters, a fine tuning is required.

Furthermore, throughout the development of the present work, we earned some observations and lessons learned. Even though some of these aspects are not entirely new, we consider it relevant to discuss them, specially in terms of our domain.

Size does matter. For our case, the crucial aspect is not size *per sé*, but rather the *intricateness* of the document graph; this obviously impacts variables such as the length of neighborhoods and can complicate several operations, such as the search procedure that in cluster construction. Moreover, not only is the construction sub-task affected by of this issue; data representation, validation, and description have to cope to an increasing complexity as well. Likewise, when dealing with massive data, we have experienced that it is not trivial to predict the behavior with the whole dataset by using smaller cases, as differences begin to intensify with rising orders of magnitude; with regard to this, it was not the same to carry out the process with 10,000 documents than with 30,000, 100,000, and 900,000.

Competition between coverage and quality can be stretch. Although this situation tends to be frequent for optimization and real-life problems, we could confirm that, for this domain, the rivalry between these two goals is fierce. This was mainly seen with the clustering algorithm, in which several of the enhancements were oriented towards doing a "cheaper" construction for the sake of completion.

Not everything is important. Even when this point is partially subjective—and for this reason we decided to consider all articles in our clustering—, it is also true that there are certain documents that can be accounted, in general, as irrelevant: bad words, pornography-related, and nonsense articles serve as examples. Despite the amount of these, considering such documents places an extra load on the method, and finally the results (not very encouraging for the usual), have to be filtered. Therefore, it seems advantageous to envisage a mechanism (either human or automatic) for detecting which elements are worth to cluster; doing so could avoid wasting time and effort, and probably yield a better quality. This leads us to our next observation: the importance of pre-processing.

Pre-processing should not be underestimated. This first step of the data mining process might not seem critical, but actually the costs of neglecting data preparation tend to replicate at each subsequent stage of the process. In that sense, we found it beneficial to invest time on this endeavor; on the other hand, we also experienced the negative aspects on our first attempts. Now, from the hand of pre-processing comes also a previous knowledge of the corpus; from our experience, this can also result advantageous, because the approach can be tailored according to specific features. As a consequence, spending time to gather information—either from literature or exploratory experiments—is more of an intelligent investment.

A supplementary remark is given by our *overall working approach*, which was mainly empirical. In that sense, the extraction process was developed on a trial-and-error basis, where observation and intense experimental exploration played a fundamental role; logically, theoretical foundations were not left apart, but were used more soberly. Another key feature of our approach comprises preference for breadth; with regard to this trait, our main concern was to provide a comprehensive end-to-end process, which could be capable of covering the most critical facets related to topic extraction. Obviously we realize that going towards this direction makes it difficult to treat each and every part of the process in depth; however, an effort was done to tackle topic construction with more detail, as it is the backbone of extraction. Therefore, a deeper examination of the remaining aspects may be left in general as future work.

7.3 Answers to research questions

What is the relation between structure and thematic in Web knowledge corpora (our study domain)?

Throughout the present work, we found out that a high relative density in node groups indicates that these nodes (documents), tend to share a common thematic in WikiWikiWeb encyclopedic knowledge corpora. Furthermore, this was shown on an experimental basis—mainly by comparing these groups (clusters) against a set of reference classes (categories) that belong to the case study corpus (Wikipedia) and with the aid of human judgment.

Is our approach able to detect (construct) topics solely based on structure? How?

Yes, topics can be detected (constructed) with an approach solely based on structure by searching for highly interlinked (dense) sub-graphs on the corpus; to identify such sub-graphs, we only require to have the links among documents. Our approach, in particular, constructs topical clusters by iteratively adding elements (documents) in the vicinity of a given starting point.

Is our approach able to extract topics while considering the whole collection? How?

Yes; the approach is able to consider the whole collection by focusing on specific regions or (local) neighborhoods at a given time. As a result, the corpus is not viewed all at once, but by parts. That way, all the search space is gradually covered when attempting to place the different documents into one or more topical clusters.

7.4 Future Work

Because the context in which the current work was developed is very broad, there is a considerable number of interesting and convenient future directions. Because the scope and generality of these directions is variable, the most extensive ones shall be discussed first; afterwards, specific continuation works are to be enumerated. At the end, several concrete applications shall be detailed.

Regarding future work in general, several fundamental issues still remain to be covered with respect to topic extraction in Web collections. Besides describing each of these issues, we will suggest how they can possibly be addressed:

Document-centered extraction. So far, we have been initially concerned on solving the problem of obtaining the cohesive topics of a collection. However, a variant of this task is to extract the topic that comprises a specific document; although this endeavor is similar to our approach, there are a couple of key differences. On one hand, more than being used as a means for discovering a dominant theme, the document actually sits at the center of the topic we aim to extract. The former implies, necessarily, that the document must remain on-topic all the time; therefore, it cannot be removed from the cluster. Also, this case suggests a restriction in scope, assuming that the members of the document's topic should not lie very far from the initial point (in fact, this conjecture was also proven by Menczer in his study of link semantics [92]). This consideration is basically equivalent to avoiding topic drift at distillation, since the output of this task (authorities and hubs) should be related to the given broad query; in that sense, the query here is the document for which the corresponding topic wants to be found. Furthermore, this raises the issue of knowing how to distinguish a loose topic from a poorly constructed cluster, as the final output could be of little cohesion; perhaps a general solution could be to complement the structural approach with other methods (e.g. based on text or usage) to combine evidence and make it stronger. As a consequence, the present work can be seen as a precedent with regard to structural methods on this domain, and can serve to establish the strengths and weaknesses of such pure approaches for topic extraction.

Temporal aspects. This is an essential aspect for corpora like Wikipedia; therefore, a considerable number of points has to be brought up for discussion.

How dynamic is Wikipedia? In our case, we started by concentrating only on a snapshot of the collection, but we know that for its popularity and openness to modifications, it has a great potential for being highly dynamic. While we believe that structure will tend to be more stable over time than contents (text), this fact still remains to be proved; furthermore, by having an idea of how dynamic and evolving Wikipedia is, one can gain an insight on the way to treat time-related aspects and on the attention they deserve.

Updates and tendencies. Associated to the previous is the issue of updating the topics of the collection. Here, we can distinguish two main cases: adding a new article to a topic and re-calculating the topics every certain period. With respect to newly created articles, we believe a viable option is to use a *classification* procedure for placing the article into its corresponding clusters; perhaps this could be done by using topic properties and the document's characteristic features. However, this might be not the only possible approach; in that sense, this issue is left open and may be addressed in a similar or a completely different way than the one commented here. For the second point, it seems natural that, after some time and continuous changes to the document repository, refreshing the topic base (regrouping) becomes necessary. Undoubtedly, this task is more complex and requires to handle several key areas; the first of these is how to determine the point after which the update is to be done. A second problem is to define the type of update to carry out; acknowledging the dimensions of Wikipedia-like corpora, it seems beneficial to explore an incremental alternative. This leads us to additionally consider the aspect of tendencies and stability, since this could help to determine what parts of the collection it is crucial to re-cluster—for instance, if we already know that a certain topic is very stable along time, it is not useful to construct this cluster again. The former, on its own, perhaps is also related to analyzing controversies and popular articles.

Hierarchical schemes. An overlapping scheme is able to represent the fact that a document may belong to different themes; however, another highly desirable clustering feature is to present topics in a stratified manner (like a directory, for instance). Conventional hierarchical clustering by itself is not scalable, but other techniques could be explored and/or proposed to produce this kind of scheme.

There are other future directions that could be taken as a proper continuation of the current work. Some of these consist of:

Statistical significance.- An immediate work that remains to be done concerns running the algorithm a sufficient number of times (e.g. 30 or more) for achieving significance, which in turn helps to guarantee robustness and repeatability, and to increase certainty in our results.

Refinements and variations.- Here, there is a wide range of extensions that could be carried out. For example, using weighted graphs, employing hybrid information sources (text, links, usage data, etc.), trying out different parameter combinations and values, etc. For example, as stated previously, instead of using a fixed quantum, we could dynamically extend the time for a given cluster if a constant progress in its construction is shown.

Scalability and efficiency.- Even when these issues have been addressed on the present work (as they are inherent to our problem), it is always possible to improve on these couple of areas. A straightforward extension, for instance, comprises parallelization of the clustering algorithm.

Exploration of topic properties.- Along the present work, an initial exploration of possible topic properties and their management was done. However, a possible extension could consist of going deeper into this aspect; for instance, a concrete validation scheme for assessing property quality could be proposed, alternative methods for obtaining our two discussed properties could be explored, or even new properties could be suggested (e.g. a text summary for the cluster).

7.4.1 Applications

As we mentioned earlier, applications that could benefit from topic extraction include semantic information retrieval (browsing and searching), visualization, automatic linking, and automatic resource construction (e.g. building or suggesting elements for Wikipedia article sections such as "Related to" or "See also"). Nevertheless, in order to concretely show how the outputs of the extraction process can be utilized, let us further detail two specific applications that can be developed: *link highlighting* and *topic clouds*. Both of them are being thought inside the context of Wikipedia, but are not necessarily limited to this domain.

Link highlighting consists of outstanding, in an article, those hyperlinks that denote documents lying on the same topic cluster; therefore, the aim here is to provide an overview of the articles that are related to a specific subject matter while the user is examining one of these documents. This application results convenient mainly for users that are beginning to get acquainted with a certain subject, because the most relevant pieces of information related to the article they are reading are distinguished from the rest, which are mostly tangential relationships. Furthermore, highlighting allows to automatically determine which links are really important, instead of leaving this task completely on the hands of editors. Another advantage is that highlighted links would by themselves constitute a complement to the "See also" and "Related to" sections in Wikipedia.

Table 7.1 shows how link highlighting in Wikipedia pages would look like. Of course, this is a modest version, and more sophisticated variations could eventually take place. For example, instead a single hue, a range of colors could be used to indicate how important the linked document is within the context; this could be specially useful for didactic purposes or children education.

Table 7.1: Link highlighting example

The Beatles were an English rock band, formed in Liverpool in 1960 and one of the most commercially successful and critically acclaimed acts in the history of popular music. From 1962 the group consisted of **John Lennon** (rhythm guitar, vocals), **Paul McCartney** (bass guitar, vocals), **George Harrison** (lead guitar, vocals) and **Ringo Starr** (drums, vocals). Rooted in skiffle and 1950s **rock and roll**, the group later worked in many genres ranging from folk rock to psychedelic pop, often incorporating classical and other elements in innovative ways. The nature of their enormous popularity, which first emerged as the **"Beatlemania"** fad, transformed as their songwriting grew in sophistication. The group came to be perceived as the embodiment of progressive ideals, seeing their influence extend into the social and cultural revolutions of the 1960s.

With an early five-piece line-up of Lennon, McCartney, Harrison, **Stuart Sutcliffe** (bass) and **Pete Best** (drums), The Beatles built their reputation in Liverpool and Hamburg clubs over a three-year period from 1960. Sutcliffe left the group in 1961, and Best was replaced by Starr the following year. Moulded into a professional outfit by music store owner Brian Epstein after he offered to act as the group's manager, and with their musical potential enhanced by the hands-on creativity of producer George Martin, The Beatles achieved UK mainstream success in late 1962 with their first single, **"Love Me Do"**. Gaining international popularity over the course of the next year, they toured extensively until 1966, then retreated to the recording studio until their breakup in 1970. Each then found success in an independent musical career. McCartney and Starr remain active; Lennon was shot and killed in 1980, and Harrison died of cancer in 2001.

During their studio years, The Beatles produced what critics consider some of their finest material, including the album **Sgt. Pepper's Lonely Hearts Club Band** (1967), widely regarded as a masterpiece. Four decades after their breakup, The Beatles' music continues to be popular. The Beatles have had more number one albums on the UK charts, and held down the top spot longer, than any other musical act. According to RIAA certifications, they have sold more albums in the US than any other artist. In 2008, Billboard magazine released a list of the all-time top-selling Hot 100 artists to celebrate the US singles chart's fiftieth anniversary, with The Beatles at number one. They have been honoured with 7 Grammy Awards, and they have received 15 Ivor Novello Awards from the British Academy of Songwriters, Composers and Authors. The Beatles were collectively included in Time magazine's compilation of the 20th century's 100 most important and influential people.

Regarding topic clouds, they would constitute a resource similar to Wikipedia Portals and Lists; however, just as conventional tag clouds, these structures would be oriented towards providing an overview of a particular topic and would present article titles in a more graphical fashion. In that sense, a topic cloud could be conformed of the k most outstanding documents from a given cluster (k could be left for editors or users to set), each document being represented by its title and in a font size and position that corresponds to its centrality; therefore, the most important topic documents would be larger and placed towards the middle of the page (see Figure 7.1). Additionally, these clouds could be searched by means of topic tags; nonetheless, these are merely suggestions.

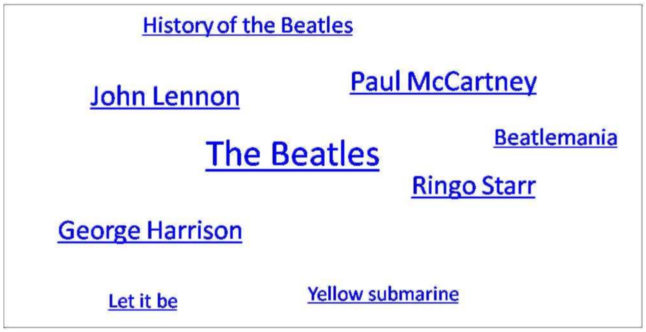


Figure 7.1: Topic cloud example

So far, we have centered on a pair of didactic applications; nevertheless, the topic extraction process could also be used for commercial purposes, such as focused advertising. For instance, let us consider once more the Beatles topical cluster: if Wikipedia's management is aware of which articles make up such cluster, and a new compilation of the band's hits is about to go out into the market, the corresponding promotional banners could be placed on these articles. Considering that the case study Web collection is visited by millions, such strategy could become advantageous.

However, the application scope is not limited only to Wikipedia—although describing applications within this specific collection flows more naturally. In that sense, the approach could be extended to other Web collections or the Web itself; for this last case, several examples could be stated. For instance, the method could be integrated to a commercial search engine (like Google) in order to group query results by topic—or "sub-topic", if we take into account that the webpage result set of an unambiguous query is already a theme. Of course, working at the scale of the Web (obviously larger than any collection derived from it) places higher demands on efficiency, whether the mechanism is executed on-line or off-line. For the on-line version, perhaps parallelization and answer approximation could be a pair of options to consider; on the other hand, for the off-line version, a viable alternative could be to cluster based on the most frequent/popular queries posed to the engine (e.g., "britney spears") and store these results for future requests. Information about these popular queries can be obtained via log data, and there would have to be updates on the clusters if the related webpages show to be dynamic. Topic properties could also be used to fetch and present results, but, once again, all of these details are merely coarse-grained ideas.

Moreover, although our approach was designed thinking about the Web, it could even be applied over data outside of this context. For example, an interesting domain is the organization by topics of the items stored on a computer (e.g., the files found on the desktop). This organization could be achieved by, for instance, making use of metadata from the file system, as this information may contain links among documents. In general, the construction approach is able to work with any kind of linked data (as a matter of fact, community discovery techniques can be and have been applied on a variety of networks and domains); nevertheless, even when there was a lack of “physical” links, we could still build a graph based on object similarity[1] and cluster this structure. Regarding this aspect, to avoid high costs on massive data, the similarity graph could be built “on demand” by locally constructing those regions that are required at a time. Finally, as we have been mentioning, these and other fine details correspond to future work.

[1] As we may recall from the state of the art (Chapter 2), network clustering techniques can actually be employed over similarity graphs, where (weighted) edges depict resemblance between pairs of objects. In fact, different thresholds can be used for making the graph more dense or sparse by allowing only edges with a certain weight to exist.

Bibliography

[1] ADAFRE, S. F., AND DE RIJKE, M. Discovering missing links in wikipedia. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery* (New York, NY, USA, 2005), ACM, pp. 90–97.

[2] AHUJA, R., ERGUN, O., ORLIN, J., AND PUNNEN, A. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics 123*, 1-3 (2002), 75–102.

[3] ALBERT, R., AND BARABÁSI, A. Statistical mechanics of complex networks. *Reviews of modern physics 74*, 1 (2002), 47–97.

[4] ALLAN, J. Automatic hypertext link typing. In *Proceedings of the the seventh ACM conference on Hypertext* (1996), ACM New York, NY, USA, pp. 42–52.

[5] ALLAN, J., Ed. *Introduction to topic detection and tracking.* Kluwer Academic Publishers, 2002.

[6] AMITAY, E., AND PARIS, C. Automatically summarising web sites: is there a way around it? In *Proceedings of the ninth international conference on Information and knowledge management* (2000), ACM New York, NY, USA, pp. 173–179.

[7] ANDREWS, N., AND FOX, E. Recent developments in document clustering. Tech. rep., Virginia Tech, 2007.

[8] AUER, S., AND LEHMANN, J. What have Innsbruck and Leipzig in common? Extracting semantics from Wiki content. *Lecture Notes in Computer Science 4519* (2007), 503.

[9] BAEZA-YATES, R. *Modern Information Retrieval.* Addison-Wesley, New York, 1999.

[10] BASS, L., CLEMENTS, P., AND KAZMAN, R. *Software Architecture in Practice.* Addison-Wesley Longman Publishing Co., Inc., 1998.

[11] BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. The Semantic Web. *Scientific American 284*, 5 (2001), 28–37.

[12] BERRY, M., AND LINOFF, G. *Mastering data mining: The art and science of customer relationship management.* John Wiley & Sons, Inc. New York, NY, USA, 1999.

[13] BHARAT, K., AND HENZINGER, M. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), ACM New York, NY, USA, pp. 104–111.

[14] BLOEHDORN, S., AND BLOHM, S. A self organizing map for relation extraction from Wikipedia using structured data representations. In *Proceedings of the International Workshop on Intelligent Information Access (IIIA-2006)* (2006), Citeseer.

[15] BOCCALETTI, S., LATORA, V., MORENO, Y., CHAVEZ, M., AND HWANG, D. Complex networks: Structure and dynamics. *Physics Reports 424*, 4-5 (2006), 175–308.

[16] BONATO, A. A survey of models of the web graph. *Lecture notes in computer science 3405* (2005), 159.

[17] BOTAFOGO, R. Cluster analysis for hypertext systems. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (1993), ACM Press New York, NY, USA, pp. 116–125.

[18] BOTAFOGO, R., RIVLIN, E., AND SHNEIDERMAN, B. Structural analysis of hypertexts: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems (TOIS) 10*, 2 (1992), 142–180.

[19] BOTAFOGO, R., AND SHNEIDERMAN, B. Identifying aggregates in hypertext structures. In *Proceedings of the third annual ACM conference on Hypertext* (1991), ACM New York, NY, USA, pp. 63–74.

[20] BOYD-GRABER, J., CHANG, J., GERRISH, S., WANG, C., AND BLEI, D. Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems (NIPS) 31* (2009).

[21] BRIN, S., AND PAGE, L. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems 30*, 1-7 (1998), 107–117.

[22] BRODER, A., KUMAR, R., MAGHOUL, F., RAGHAVAN, P., RAJAGOPALAN, S., STATA, R., TOMKINS, A., AND WIENER, J. Graph structure in the Web. *Computer Networks 33*, 1-6 (2000), 309–320.

[23] BROOKS, C., AND MONTANEZ, N. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th international conference on World Wide Web* (2006), ACM, p. 632.

[24] CHAKRABARTI, S. *Mining the Web: Discovering knowledge from hypertext data.* Morgan Kaufmann, 2003.

[25] CHAKRABARTI, S., DOM, B., GIBSON, D., KLEINBERG, J., KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. Mining the link structure of the world wide web. *IEEE Computer 32*, 8 (1999), 60–67.

[26] CHAKRABARTI, S., DOM, B., GIBSON, D., KUMAR, S., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. Experiments in topic distillation. In *ACM SIGIR Workshop on Hypertext Information Retrieval on the Web* (1998), Melbourne, Australia.

[27] CHAWNER, B., AND LEWIS, P. WikiWikiWebs: New ways of interacting in a Web environment. Online-Publikation: http: www. ala. org/ala/lita/litaevents/2004Forum/CS_WikiWikiWebs. pdf (18.02. 2005), 2004.

[28] CHIRITA, P., COSTACHE, S., NEJDL, W., AND HANDSCHUH, S. P-tag: large scale automatic generation of personalized annotation tags for the web. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, p. 854.

[29] CHUNG, F. Lectures on spectral graph theory. http://yaroslavvb.com/papers/chung-lectures.pdf. Lecture presentation.

[30] CHUNG, F. *Spectral graph theory.* American Mathematical Society, 1997.

[31] CILIBRASI, R., AND VITÁNYI, P. The Google Similarity Distance. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING 19* (2007), 370–383.

[32] CLAUSET, A., NEWMAN, M., AND MOORE, C. Finding community structure in very large networks. *Physical Review E 70*, 6 (2004), 66111.

[33] COOK, D., AND HOLDER, L. *Mining graph data.* Wiley-Interscience, 2006.

[34] DEERWESTER, S., DUMAIS, S., FURNAS, G., LANDAUER, T., AND HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American society for information science 41*, 6 (1990), 391–407.

[35] DHILLON, I., GUAN, Y., AND KULIS, B. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), ACM New York, NY, USA, pp. 551–556.

[36] DOLAN, S. Six Degrees of Wikipedia. `http://www.netsoc.tcd.ie/~mu/wiki/`. (Retrieved on May 10, 2010).

[37] EBERSBACH, A., GLASER, M., AND HEIGL, R. *Wiki: Web collaboration.* Springer-Verlag New York Inc, 2008.

[38] ERTÖZ, L., STEINBACH, M., AND KUMAR, V. Finding topics in collections of documents: A shared nearest neighbor approach. *Clustering and Information Retrieval 1* (2003), 83–104.

[39] FELDMAN, R., AND SANGER, J. *The text mining handbook.* Cambridge University Press New York:, 2007.

[40] FELLBAUM, C. *WordNet: An electronic lexical database.* MIT press Cambridge, MA, 1998.

[41] FLAKE, G., LAWRENCE, S., AND GILES, C. Efficient identification of Web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (2000), ACM New York, NY, USA, pp. 150–160.

[42] FLAKE, G., LAWRENCE, S., GILES, C., AND COETZEE, F. Self-organization and identification of Web communities. *Computer 35*, 3 (2002), 66–70.

[43] FLAKE, G., PENNOCK, D., AND FAIN, D. The self-organized Web: The yin to the Semantic Webs yang. *IEEE Intelligent Systems 18*, 4 (2003), 75–77.

[44] FURNAS, G., DEERWESTER, S., DUMAIS, S., LANDAUER, T., HARSHMAN, R., STREETER, L., AND LOCHBAUM, K. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval* (1988), ACM New York, NY, USA, pp. 465–480.

[45] GARFIELD, E. Citation analysis as a tool in journal evaluation. *Essays of an Information Scientist 1* (1962-1973), 527–544.

[46] GARZA, S., AND BRENA, R. Graph Local Clustering for Topic Detection in Web Collections. In *2009 Latin American Web Congress* (2009), IEEE, pp. 207–213.

[47] GARZA, S., ELIZALDE, L., AND CANSECO, A. Clustering Hyperlinks for Topic Extraction: An Exploratory Analysis. In *2009 Eighth Mexican International Conference on Artificial Intelligence* (2009), IEEE, pp. 128–133.

[48] GETOOR, L., AND DIEHL, C. Link mining: a survey. *ACM SIGKDD Explorations Newsletter 7*, 2 (2005), 3–12.

[49] GIBSON, D., KLEINBERG, J., AND RAGHAVAN, P. Inferring Web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems* (1998), ACM New York, NY, USA, pp. 225–234.

[50] GIBSON, D., KUMAR, R., AND TOMKINS, A. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st international conference on Very large data bases* (2005), VLDB Endowment, pp. 721–732.

[51] GIRVAN, M., AND NEWMAN, M. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences 99*, 12 (2002), 7821.

[52] GRIFFITHS, T. L., AND STEYVERS, M. Finding scientific topics. *Proc Natl Acad Sci U S A 101 Suppl 1* (April 2004), 5228–5235.

[53] GRIFFITHS, T. L., STEYVERS, M., AND TENENBAUM, J. B. Topics in semantic representation. *Psychological Review 114* (2007), 211–244.

[54] GROSSMAN, D., AND FRIEDER, O. *Information retrieval: Algorithms and heuristics.* Springer, 2004.

[55] HALKIDI, M., BATISTAKIS, Y., AND VAZIRGIANNIS, M. On clustering validation techniques. *Journal of Intelligent Information Systems 17*, 2 (2001), 107–145.

[56] HASSAN-MONTERO, Y., AND HERRERO-SOLANA, V. Improving tag-clouds as visual information retrieval interfaces. In *International Conference on Multidisciplinary Information Sciences and Technologies* (2006), Citeseer, pp. 25–28.

[57] HE, X., DING, C., ZHA, H., AND SIMON, H. Automatic topic identification using webpage clustering. In *Proc. IEEE Intl Conf. Data Mining. San Jose, CA* (2001), pp. 195–202.

[58] HERR, B., AND HOLLOWAY, T. Visualizing the Power Struggle in Wikipedia.

[59] HOBBS, J. Topic drift. *Conversational organization and its development 38* (1990), 3–22.

[60] HUANG, J., ZHU, T., AND SCHUURMANS, D. Web communities identification from random walks. *LECTURE NOTES IN COMPUTER SCIENCE 4213* (2006), 187.

[61] HUANG, X., AND LAI, W. Identification of clusters in the Web graph based on link topology. In *Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International* (2003), pp. 123–128.

[62] IDE, N., AND VÉRONIS, J. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics 24*, 1 (1998), 2–40.

[63] JACOBSON, I., GRISS, M., AND JONSSON, P. *Software reuse: architecture, process and organization for business success.* ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1997.

[64] JAIN, A., AND DUBES, R. *Algorithms for Clustering Data.* Prentice-Hall Advanced Reference Series, 1988.

[65] JAIN, A., MURTY, M., AND FLYNN, P. Data clustering: a review. *ACM computing surveys 31*, 3 (1999), 264–323.

[66] JANSON, S., ŁUCZAK, T., AND RUCIŃSKI, A. *Random graphs.* John Wiley New York, 2000.

[67] JARVIS, R., AND PATRICK, E. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers 22*, 11 (1973), 1025–1034.

[68] JO, Y., LAGOZE, C., AND GILES, C. Detecting research topics via the correlation between graphs and texts. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (2007), ACM, p. 379.

[69] KARYPIS, G., HAN, E., AND KUMAR, V. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE computer 32*, 8 (1999), 68–75.

[70] KARYPIS, G., AND KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing 20*, 1 (1999), 359.

[71] KESSLER, M. Bibliographic coupling between scientific papers. *American documentation 14*, 1 (1963), 10–25.

[72] KIM, M., AND RAMAKRISHNA, R. New indices for cluster validity assessment. *Pattern Recognition Letters 26*, 15 (2005), 2353–2363.

[73] KLEINBERG, J. Authoritative sources in a hyperlinked environment. *Journal of the ACM 46*, 5 (1999), 604–632.

[74] KLEINBERG, J., KUMAR, S., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. The Web as a Graph: Measurements, Models and Methods. *LECTURE NOTES IN COMPUTER SCIENCE 1627* (1999), 1–17.

[75] KOSALA, R., AND BLOCKEEL, H. Web mining research: A survey. *SIGKDD Explorations 2* (2000), 1–15.

[76] KUBICA, J., MOORE, A., SCHNEIDER, J., AND YANG, Y. Stochastic link and group detection. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE* (2002), Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, pp. 798–806.

[77] KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., SIVAKUMAR, D., TOMPKINS, A., AND UPFAL, E. The Web as a graph. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (2000), ACM New York, NY, USA, pp. 1–10.

[78] KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. Trawling the web for emerging cyber-communities. *COMPUTER NETWORKS 31*, 11 (1999), 1481–1493.

[79] LATTIN, J., CARROLL, J., AND GREEN, P. *Analyzing multivariate data.* Thomson Brooks/Cole, 2003.

[80] LAUDON, K., AND LAUDON, J. *Management Information Systems: Managing the Digital Firm*, eight edition ed. Pearson, New Jersey, 2003.

[81] LESKOVEC, J., LANG, K. J., DASGUPTA, A., AND MAHONEY, M. W. Statistical properties of community structure in large social and information networks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web* (New York, NY, USA, 2008), ACM, pp. 695–704.

[82] LI, H., AND YAMANISHI, K. Topic analysis using a finite mixture model. *Information processing and management 39*, 4 (2003), 521–541.

[83] LIU, B. *Web data mining: exploring hyperlinks, contents, and usage data.* Springer, 2007.

[84] LIU, Y., NICULESCU-MIZIL, A., AND GRYC, W. Topic-link LDA: joint models of topic and author community. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM New York, NY, USA.

[85] MACHNIK, Ł. Documents clustering method based on ants algorithms. In *Proceedings of the International Multiconference on ISSN* (2006), p. 7094.

[86] MANNING, C., RAGHAVAN, P., AND SCHTZE, H. *Introduction to information retrieval.* Cambridge University Press New York, NY, USA, 2008.

[87] MARDIA, K., KENT, J., AND BIBBY, J. *Multivariate Analysis.* Academic Press, London, 1979.

[88] MARLOW, C., NAAMAN, M., BOYD, D., AND DAVIS, M. Position paper, tagging, taxonomy, flickr, article, toread. In *Collaborative web tagging workshop at WWW* (2006), vol. 6, Citeseer.

[89] MATTEUCCI, M. A Tutorial on Clustering Algorithms. `http://home.dei.polimi.it/matteucc/Clustering/tutorial\_html/index.html`. (Retrieved on May 10, 2010).

[90] MCCALLUM, A., WANG, X., AND CORRADA-EMMANUEL, A. Topic and role discovery in social networks with experiments on Enron and academic email. *Journal of Artificial Intelligence Research 30*, 1 (2007), 249–272.

[91] MEI, Q., CAI, D., ZHANG, D., AND ZHAI, C. Topic modeling with network regularization. In *WWW '08: Proceeding of the 17th international conference on World Wide Web* (New York, NY, USA, 2008), ACM, pp. 101–110.

[92] MENCZER, F. Links tell us about lexical and semantic Web content. *Arxiv preprint cs/0108004* (2001).

[93] MENESES, E. Vectors and Graphs: Two Representations to Cluster Web Sites Using Hyperstructure. In *Proceedings of the Fourth Latin American Web Congress* (2006), IEEE Computer Society Washington, DC, USA, pp. 172–178.

[94] MENESES, E. Effective representations for web document clustering. Master's thesis, Instituto Tecnológico de Costa Rica, 2007.

[95] MICHALSKI, R., STEPP, R., AND DIDAY, E. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE TRANS. PATTERN ANAL. MACH. INTELLIG. 5*, 4 (1983), 396–409.

[96] MIHALCEA, R., AND CSOMAI, A. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (2007), ACM New York, NY, USA, pp. 233–242.

[97] MILNE, D., AND WITTEN, I. Learning to link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (2008), ACM New York, NY, USA, pp. 509–518.

[98] MODHA, D., AND SPANGLER, W. Clustering hypertext with applications to Web searching, Sept. 11 2003. US Patent App. 10/660,242.

[99] MOON, T. The expectation-maximization algorithm. *IEEE Signal processing magazine 13*, 6 (1996), 47–60.

[100] NAKAYAMA, K., HARA, T., AND NISHIO, S. Wikipedia Link Structure and Text Mining for Semantic Relation Extraction. *Semantic Search 334* (2008), 59.

[101] NAVIGLI, R. Word sense disambiguation: a survey. *ACM Computing Surveys (CSUR) 41*, 2 (2009), 10.

[102] NEWMAN, M. The structure and function of complex networks. *SIAM Review 45* (2003), 167–256.

[103] NEWMAN, M. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences 103*, 23 (2006), 8577.

[104] NEWMAN, M., AND GIRVAN, M. Finding and evaluating community structure in networks. *PHYSICAL REVIEW E Phys Rev E 69* (2004), 026113.

[105] OREILLY, T. What is Web 2.0: Design patterns and business models for the next generation of software. *COMMUNICATIONS & STRATEGIES 65* (2007), 17–37.

[106] ORPONEN, P., AND SCHAEFFER, S. Local clustering of large graphs by approximate fiedler vectors. *Lecture Notes in Computer Science 3503* (2005), 524–533.

[107] PALIOURAS, G., PAPATHEODOROU, C., KARKALETSIS, V., TZITZIRAS, P., AND SPYROPOULOS, C. Large-scale mining of usage data on web sites. In *AAAI 2000 Spring Symposium on Adaptive User Interfaces* (2000).

[108] PAPADOPOULOS, S., KOMPATSIARIS, Y., AND VAKALI, A. Leveraging Collective Intelligence through Community Detection in Tag Networks. In *First International Workshop on Collective Knowledge Capturing and Representation - CKCaR'09* (2009).

[109] PARSONS, L., HAQUE, E., AND LIU, H. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter 6*, 1 (2004), 90–105.

[110] PETERSON, J., AND SILBERSCHATZ, A. *Operating system concepts*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1985.

[111] PIRLOT, M. General local search methods. *European journal of operational research 92*, 3 (1996), 493–511.

[112] PIROLLI, P., PITKOW, J., AND RAO, R. Silk from a sow's ear: extracting usable structures from the Web. In *Proceedings of the SIGCHI Conference on Human factors in computing systems: common ground* (1996), ACM New York, NY, USA, pp. 118–125.

[113] Puig-Centelles, A., Ripolles, O., and Chover, M. Identifying communities in social networks: a survey. In *Proceedings of the IADIS International Conference Web Based Communities* (2007), pp. 350–354.

[114] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences 101*, 9 (2004), 2658–2663.

[115] Ramirez, E., and Brena, R. An information-theoretic approach for unsupervised topic mining in large text collections. In *IEEE/WIC/ACM International Conference on Web Intelligence, WI 2009, Milan, Italy.* (2009).

[116] Rand, W. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association 66* (1971), 846–850.

[117] Rosell, M. Presentation. Document Clustering.

[118] Russell, S. J., and Norvig, P. *Artificial Intelligence: a modern approach.* Prentice Hall, 1995.

[119] Sait, S., and Youssef, H. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems.* IEEE Computer Society Press Los Alamitos, CA, USA, 1999.

[120] Salton, G. *The SMART Retrieval System—Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[121] Salton, G., and Buckley, C. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management* (1988), pp. 513–523.

[122] Salton, G., Wong, A., and Yang, C. A vector space model for automatic indexing. *Commun. ACM 18* (1975), 613–620.

[123] Satoshi, S. N., Oyama, S., Hayamizu, T., and Ishida, T. Analysis and Improvement of HITS Algorithm for Detecting Web Communities. In *Communities, The 2002 International Symposium on Applications and the Internet* (2002), IEEE Computer Society, pp. 132–140.

[124] Schaeffer, S. Stochastic local clustering for massive graphs. *Advances in Knowledge Discovery and Data Mining 3518* (2005), 354–360.

[125] Schaeffer, S. Graph clustering. *Computer Science Review 1*, 1 (2007), 27–64.

[126] Schaeffer, S., Marinoni, S., Särelä, M., Nikander, P., and Jorvas, F. Dynamic local clustering for hierarchical ad hoc networks. *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on 2* (2006), 667–672.

[127] SCHÖNHOFEN, P. Identifying document topics using the Wikipedia category network. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (2006), IEEE Computer Society Washington, DC, USA, pp. 456–462.

[128] SCHWARTZ, R., SISTA, S., AND LEEK, T. Unsupervised topic discovery. In *Proceedings of workshop on language modeling and information retrieval* (2001), pp. 72–77.

[129] SCOTT, J. *Social Network Analysis: a handbook*, second edition ed. SAGE Publications, 2000.

[130] SHADBOLT, N., HALL, W., AND BERNERS-LEE, T. The semantic web revisited. *IEEE Intelligent Systems 21*, 3 (2006), 96–101.

[131] SISTA, S., SCHWARTZ, R., LEEK, T. R., AND MAKHOUL, J. An algorithm for unsupervised topic discovery from broadcast news stories. In *Proceedings of the second international conference on Human Language Technology Research* (San Francisco, CA, USA, 2002), Morgan Kaufmann Publishers Inc., pp. 110–114.

[132] SMALL, H. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science 24*, 4 (1973), 265–269.

[133] SMITH, G. *Tagging: People-powered Metadata for the Social Web.* New Rider Pr, 2008.

[134] SRIVASTAVA, J., COOLEY, R., DESHPANDE, M., AND TAN, P. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD explorations 1*, 2 (2000), 12–23.

[135] STEIN, B., AND ZU EISSEN, S. Topic identification: Framework and application. In *Proc. International Conference on Knowledge Management* (2004), vol. 399, pp. 522–531.

[136] SUCHANEK, F., KASNECI, G., AND WEIKUM, G. Yago: A large ontology from Wikipedia and Wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web 6*, 3 (2008), 203–217.

[137] TAN, P.-N., STEINBACH, M., AND KUMAR, V. *Introduction to Data Mining.* Addison Wesley, 2006.

[138] THURAISINGHAM, B. *Web data mining and applications in business intelligence and counter-terrorism.* CRC, 2003.

[139] TREESE, W. Web 2.0: is it really different? *netWorker Volume 10 , Issue 2* (2006), 15 – 17.

[140] VIRTANEN, S. Clustering the Chilean Web. In *Web Congress, 2003. Proceedings. First Latin American* (2003), pp. 229–231.

[141] VÖLKEL, M., KRÖTZSCH, M., VRANDECIC, D., HALLER, H., AND STUDER, R. Semantic wikipedia. In *WWW '06: Proceedings of the 15th international conference on World Wide Web* (New York, NY, USA, 2006), ACM, pp. 585–594.

[142] VON LUXBURG, U. A tutorial on spectral clustering. *Statistics and Computing 17*, 4 (2007), 395–416.

[143] WARREN, R., AIROLDI, E., AND BANKS, D. Network Analysis of Wikipedia. *Statistical Methods in E-Commerce Research 1* (2008), 81–102.

[144] WARTENA, C., AND BRUSSEE, R. Topic detection by clustering keywords. In *DEXA 2008: 19TH INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, PROCEEDINGS* (2008).

[145] WASSERMAN, S., AND FAUST, K. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[146] WEISE, T. Global Optimization Algorithms–Theory and Application, 2008.

[147] WU, F., AND WELD, D. Autonomously semantifying Wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (2007), ACM New York, NY, USA, pp. 41–50.

[148] WU, K.-J., CHEN, M.-C., AND SUN, Y. Automatic topics discovery from hyperlinked documents. *Information Processing & Management 40*, 2 (March 2004), 239–255.

[149] XU, X., YURUK, N., FENG, Z., AND SCHWEIGER, T. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (2007), ACM, p. 833.

[150] ZADEH, L. A. Soft computing and fuzzy logic. *Software, IEEE 11* (1994), 48–56.

[151] ZHU, S., YU, K., CHI, Y., AND GONG, Y. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), ACM New York, NY, USA, pp. 487–494.

[152] ZLATIĆ, V., BOŽIČEVIĆ, M., ŠTEFANČIĆ, H., AND DOMAZET, M. Wikipedias: Collaborative Web-based encyclopedias as complex networks. *SIAM Rev Phys Rev E 74* (2003), 016115.