Hierarchical semi-supervised confidence-based active clustering and its application to the extraction of topic hierarchies from document collections

Bruno Magalhães Nogueira

Hierarchical semi-supervised confidence-based active clustering and its application to the extraction of topic hierarchies from document collections

Bruno Magalhães Nogueira

Advisors: **Profa. Dra. Solange Oliveira Rezende**
Prof. Dr. Alípio Mário Guedes Jorge

Doctoral Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP in partial fulfillment of the requirements for the Doctoral degree in Computer Science and Computacional Mathematics. This dissertation has also been presented to the Doctoral Program in Computer Science of the Faculdade de Ciências, Universidade do Porto, as part of the double degree agreement between ICMC-USP and FC-UP. EXAMINATION BOARD PRESENTATION COPY.

USP – São Carlos
November 2013

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

N778h

Nogueira, Bruno Magalhães
Hierarchical semi-supervised confidence-based active clustering and its application to the extraction of topic hierarchies from document collections / Bruno Magalhães Nogueira; orientadora Solange Oliveira Rezende; co-orientador Alípio Mário Guedes Jorge. -- São Carlos, 2013.
123 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2013.

1. Agrupamento semissupervisionado. 2. Aprendizado ativo. 3. Hierarquias de tópicos. I. Rezende, Solange Oliveira, orient. II. Jorge, Alípio Mário Guedes, co-orient. III. Título.

Agrupamento hierárquico semissupervisionado ativo baseado em confiança e sua aplicação para a extração de hierarquias de tópicos a partir de coleções de documentos

Bruno Magalhães Nogueira

Orientadores: **Profa. Dra. Solange Oliveira Rezende**
Prof. Dr. Alípio Mário Guedes Jorge

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. Esta tese também foi apresentada ao Programa de Doutoramento em Ciência de Computadores da Faculdade de Ciências da Universidade do Porto, como parte do acordo de dupla titulação entre o ICMC-USP e a FC-UP. EXEMPLAR DE DEFESA.

USP – São Carlos
Novembro de 2013

Bruno Magalhães Nogueira

Hierarchical semi-supervised confidence-based active clustering and its application to the extraction of topic hierarchies from document collections

Tese submetida à Faculdade de Ciências da Universidade do Porto
para obtenção do grau de Doutor em Ciência de Computadores

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Novembro de 2013

Dedication

To my parents,
Rita and José Geraldo.
To my beloved fiancée,
Vanessa.

Acknowledgements

First, I would like to thank God for protecting and blessing me along this hard journey.

I would like to thank my parents, Rita and José Geraldo, for all their love and unrestricted support in all moments of my life. I cannot express in words my feelings for you. If I got here it is because you inspired and motivated me. It is an honour to be a son of such special people. Also, I would like to thank my sister, Thayse, and my brother, Túlio, for all their love and companionship. My sincere thanks to my grandmother, Eva, the sweetest person I have ever known. I will be forever grateful for all you have done to me.

My special thanks to my fiancée, Vanessa, for being by my side during all this time. Your endless love and support guided me during all this period. Your loving embrace and patience in difficult times and your sweet smile in the good ones were essential during this journey.

I would like to express my deep gratitude to my advisors, Prof. Solange Rezende and Prof. Alípio Jorge. I was very lucky for having such dedicated people guiding me and filling me with encouragement. More than advisors, you were true friends, providing me academic and personal advices. Thanks for all the good will, patience and trust.

My sincere thanks to all my family, for the support in all moments in my life and the comprehension with my absences in holidays and family gatherings. Also, my deep acknowledgements to my friends from Belo Horizonte for being so present despite the distance: Bruno Costoli, Felipe Gonçalves, Leandro Cardoso, Maria Fernanda Fonseca and Rodrigo Costa.

For all professors, colleagues and friends from LABIC, in São Carlos, I owe special thanks for the good moments and the collaboration in this work. In special, I would like to thank Diego Silva, Everton Cherman, Fabiano Santos, Ígor Braga, Maria Fernanda Moura, Merley Conrado, Rafael Rossi, Ricardo Marcacini and Vinícius Souza for all the support and companionship.

Special thanks also to all colleagues and friends from LIAAD, in Porto, for the good times during my stay there. In particular, I would like to thank Prof. Pavel Brazdil, Prof.

João Gama and Prof. Carlos Soares for their collaboration and for being so receptive. I am also specially grateful to André Rossi, Cláudio Sá, Fernando Corrêa, Márcia Oliveira, Melissa Rodrigues, Odair Tavares, Pedro Almeida, Petr Kosina and Robson Motta for their friendship.

I would also like to thank Anandsing Dwarkasing for all the reviews in the writing of our papers and this thesis.

Thanks also to ICMC/USP and the University of Porto, for providing academic structures that enabled the development of this work in cotutelle. Also, I would like to acknowledge the financial support from the Coordination for the Improvement of Higher Education Personnel - CAPES (Brazil) and from the Erasmus Mundus Euro Brazilian Windows II - EBWII (EU).

Finally, I would like to thank to all those who directly or indirectly contributed to the development of this research.

Abstract

Topic hierarchies are efficient ways of organizing document collections. These structures help users to manage the knowledge contained in textual data. These hierarchies are usually obtained through unsupervised hierarchical clustering algorithms. By not considering the context of the user in the formation of the hierarchical groups, unsupervised topic hierarchies may not attend the user's expectations in some cases. One possible solution for this problem is to employ semi-supervised clustering algorithms. These algorithms incorporate the user's knowledge through the usage of constraints to the clustering process. However, in the context of semi-supervised hierarchical clustering, the works in the literature do not efficient explore the selection of cases (instances or cluster) to add constraints, neither the interaction of the user with the clustering process. In this sense, in this work we introduce two semi-supervised hierarchical clustering algorithms: HCAC (Hierarchical Confidence-based Active Clustering) and HCAC-LC (Hierarchical Confidence-based Active Clustering with Limited Constraints). These algorithms employ an active learning approach based in the confidence of cluster merges. When a low confidence merge is detected, the user is invited to decide, from a pool of candidate pairs of clusters, the best cluster merge in that point. In this work, we employ HCAC and HCAC-LC in the extraction of topic hierarchies through the SMITH framework, which is also proposed in this thesis. This framework provides a series of well defined activities that allow the user's interaction in the generation of topic hierarchies. The active learning approach used in the HCAC-based algorithms, the kind of queries employed in these algorithms, as well as the SMITH framework for the generation of semi-supervised topic hierarchies are innovations to the state of the art proposed in this thesis. Our experimental results indicate that HCAC and HCAC-LC outperform other semi-supervised hierarchical clustering algorithms in diverse scenarios. The results also indicate that semi-supervised topic hierarchies obtained through the SMITH framework are more intuitive and easier to navigate than unsupervised topic hierarchies.

Resumo

Hierarquias de tópicos são formas eficientes de organização de coleções de documentos, auxiliando usuários a gerir o conhecimento materializado nessas publicações textuais. Tais hierarquias são usualmente construídas por meio de algoritmos de agrupamento hierárquico não supervisionado. Entretanto, por não considerarem o contexto do usuário na formação dos grupos, hierarquias de tópicos não supervisionadas nem sempre conseguem atender às suas expectativas. Uma solução para este problema é o emprego de algoritmos de agrupamento semissupervisionado, os quais incorporam o conhecimento de domínio do usuário por meio de restrições. Entretanto, para o contexto de agrupamento hierárquico semissupervisionado, não são eficientemente explorados na literatura métodos de seleção de casos (instâncias ou grupos) para receber restrições, bem como não há formas eficientes de interação do usuário com o processo de agrupamento hierárquico. Dessa maneira, neste trabalho, dois algoritmos de agrupamento hierárquico semissupervisionado são propostos: HCAC (*Hierarchical Confidence-based Active Clustering*) e HCAC-LC (*Hierarchical Confidence-based Active Clustering with Limited Constraints*). Estes algoritmos empregam uma abordagem de aprendizado ativo baseado na confiança de uma junção de clusters. Quando uma junção de baixa confiança é detectada, o usuário é convidado a decidir, em um conjunto de pares de grupos candidatos, a melhor junção naquele ponto. Estes algoritmos são aqui utilizados na extração de hierarquias de tópicos por meio do *framework* SMITH, também proposto nesse trabalho. Este framework fornece uma série de atividades bem definidas que possibilitam a interação do usuário para a obtenção de hierarquias de tópicos. A abordagem de aprendizado ativo utilizado nos algoritmos HCAC e HCAC-LC, o tipo de restrição utilizada nestes algoritmos, bem como o framework SMITH para obtenção de hierarquias de tópicos semissupervisionadas são inovações ao estado da arte propostos neste trabalho. Os resultados obtidos indicam que os algoritmos HCAC e HCAC-LC superam o desempenho de outros algoritmos hierárquicos semissupervisionados em diversos cenários. Os resultados também indicam que hierarquias de tópico semissupervisionadas obtidas por meio do *framework* SMITH são mais intuitivas e fáceis de navegar do que aquelas não supervisionadas.

This thesis was prepared with the text formatter LaTeX. We used the style developed by Ronaldo Cristiano Prati. The bibliographical citations follow the *Apalike* pattern from the BibTeXsystem.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AFCC Active Fuzzy Constrained Clustering algorithm

AL^{2}FIC Active Learning to Frequent Itemset-based Text Clustering

Average Average-link clustering algorithm

CCL Constrained Complete-Link algorithm

CLIKM Cluster-Level Interactive KMeans algorithm

CVQE Constraints Vector Quantization Error algorithm

COP-COBWEB Constraints-Partitioning COBWEB algorithm

COP-KMeans Constraints-Partitioning KMeans algorithm

CRISP-DM Cross Industry Standard Process for Data Mining

DF Document Frequency

EM Expectation Maximization

FHV Fuzzy HyperVolume measure

FS-KMeans Farthest-Seeded KMeans

HCAC Hierarchical Confidence-based Active Clustering

HCAC-LC Hierarchical Confidence-based Active Clustering with Limited Constraints

HMRF-KMeans Hidden Markov Random Fields KMeans algorithm

IDF Inverse Document Frequency

LABIC Laboratory of Computational Intelligence (*Laboratório de Inteligência Computacional*)

LCVQE Linear Constraints Vector Quantization Error algorithm

LLMA Locally Linear Metric Adaptation

LSA Latent Semantic Analysis

PCA Principal Component Analysis

PreText Text Preprocessing Tool

RLUM Robust Labeling Up Method

SCKMM Semi-supervised Clustering Kernel method based in Metric Learning

SeCLAR Selecting Candidate Labels using Association Rules

SMITH SeMI-supervised Topic Hierarchies

SNN Similarity Neural Networks

SS-KMeans Splitting-Seeded KMeans

SVaD Spatial Variant Dissimilarities

SVD Singular Value Decomposition

SVM Support Vector Machines

TF Term Frequency

TFIDF Term Frequency-Inverse Document Frequency

TopTax Methodology for automatic extraction of Topic Taxonomies

Torch TOpic HierarCHies tool

UCI University of California Irvine Machine Learning Repository

XML Extensible Markup Language

Chapter

1

Introduction

The advance and the popularization of the technologies have allowed the adoption of data collection and storage systems by diverse users and corporations. The size of the datasets increases fast and continuously. According to Gantz and Reinsel (2012), in 2020 the digital universe will contain 40 trillion of gigabytes. It is estimated that about 33% of this huge amount of data in the digital universe will have valuable information for analysis. Moreover, it is estimated that 95% of the data in the digital form are unstructured data (Gantz and Reinsel, 2009). The most common unstructured data are images and texts.

Managing and exploring this huge amount of data, specially unstructured data, has become a great challenge. In general, the analysis and comprehension of these data extrapolates the human capabilities. In other words, the application of a human-based analysis over these data would probably result in the loss of useful information given the complexity of these data.

Considering this scenario, it is of great importance the development of computational techniques that aid the humans to adequately manage the informations. Text Mining computational processes are helpful in this context. These processes provide efficient methods to transform the textual information into useful and, most of the time, innovative knowledge. For example, Text Mining has been used in applications such as bioinformatics, documents indexing and retrieving, competencies management and marketing. In this work, we focus on Text Mining applications that facilitate the organization and exploration of information in document collections, enabling an efficient knowledge management.

One of the tendencies in information management through *Text Mining* applications is topic detection in document collections (Neto et al., 2000; Lawrie et al., 2001; Kashyap

et al., 2004; Punera et al., 2005; Dupret and Piwowarski, 2005; Pons-Porrata et al., 2007; Gil-García and Pons-Porrata, 2008; Tang et al., 2008; Moura et al., 2008a; He et al., 2010; Zavitsanos et al., 2011; Paukkeri et al., 2012). By detecting the topics in the collections, the documents are organized under significant groups. In most of the applications, the number of groups present in a dataset is unknown a priori. Thus, the detection of the groups in these applications requires the usage of clustering algorithm over the document collections. Using the clustering process, similar documents are allocated in the same groups. The organization of documents in similar groups facilitate the exploration of the contents, since if the user is interested in one specific document, he/she may be also interested in similar documents (Chakrabarti, 2003).

Most of the applications that aim at detecting topics in document collections employ hierarchical clustering algorithms. Using these algorithms, the document collection is hierarchically organized, according to the contents of the documents. In this hierarchy, each concept is described in one node, so that parent nodes represent more general concepts, while child nodes represent more specific concepts (Paukkeri et al., 2012). This hierarchical structure is a very intuitive form of organizing a data collection, since it provides a visualization of the data in different levels of abstraction (Gil-García and Pons-Porrata, 2008), which facilitates the comprehension and the navigation over the document collection.

In order to facilitate the navigability and comprehensibility of this hierarchical organization of the documents, each cluster is labeled with its main descriptors. The set of descriptors of each cluster is a list of terms that is unique and more discriminative for the content of that cluster. The junction of the hierarchical structure with the cluster descriptors forms a ***topic hierarchy***, also known as a topic taxonomy (Moura et al., 2008a). Under this representation, the document collection can be navigated through a set of topics and subtopics, which are hierarchically disposed. In this sense, topic hierarchies allow different users and applications to share the same terminological description about a domain described by a document collection (Paukkeri et al., 2012).

In the literature of the organization of document collections under hierarchical structures such as topic hierarchies, most of the work employ unsupervised hierarchical clustering algorithms. As examples of traditional clustering algorithms normally used for document clustering we have the single link (Sneath and Sokal, 1973), the complete link (King, 1967), the average link (UPGMA) (Jain and Dubes, 1988) and the bisecting k-means, which consists of the iterative application of the k-means algorithm (MacQueen, 1967). According to Zhao et al. (2005), the average link and the bisecting k-means algorithms tend to obtain interesting results when statistically evaluating the obtained clusters.

However, in some applications, unsupervised clustering algorithms may not attend the user's clustering preferences in the dataset (Huang and Lam, 2009). Unsupervised

algorithms search to form groups by optimizing statistical objective functions and do not allow a supervisor interaction of the user. Thus, these algorithms do not consider the user's knowledge domain and his/her expectations over the organization of the document collection. On the other hand, as discussed above, for huge document collections, it is infeasible the supervision of the user of the entire dataset. The manual indication of the groups and the group membership to each document would demand an excessive human effort.

In this scenario, *semi-supervised clustering* algorithms emerge as an interesting alternative. These methods allow the insertion of the knowledge about the domain to the clustering process through a limited user interaction with this process. By "limited intervention" we mean the user supervision to a small proportion of the dataset. In semi-supervised clustering algorithms the user can interact with the clustering process through the addition of constraints. These constraints can be directly provided by the user or derived from labeled data and will guide the formation of the clusters, in order to adapt the clustering results to the expectations of the user (Dasgupta and Ng, 2010).

1.1 Motivation

To the best of our knowledge, there is no application of semi-supervised clustering algorithms in the extraction of topic hierarchies. However, there are several applications involving the semi-supervised document clustering, both flat and hierarchical. The performance of these algorithms are most of the time superior to the performance of unsupervised clustering algorithms, as reported in the work of Wagstaff and Cardie (2000); Basu et al. (2002) and Davidson and Ravi (2009). The results are even better when the semi-supervised clustering algorithms employ an active learning approach to select the proper cases (documents or clusters) to the user interaction (Basu et al., 2004a; Huang and Lam, 2009).

Considering this scenario, involving semi-supervised clustering and the organization of document collections, it is possible to find some gaps in the current state-of-the-art research that would be explored. The first and main research gap detected is related to the semi-supervised hierarchical clustering algorithms available. Most of the semi-supervised clustering algorithms aim at obtaining a flat structure of the data. There have been some efforts for developing solutions for semi-supervised hierarchical clustering (Talavera and Béjar, 1999; Klein et al., 2002; Kestler et al., 2006; Daniels and Giraud-Carrier, 2006; Bade et al., 2007; Böhm and Plant, 2008; Davidson and Ravi, 2009; Miyamoto and Terami, 2011; Zheng and Li, 2011). However, there are no convincing proposals for the appropriate addition of information nor for the selection of good cases to add constraints. Most of the work on active approaches and query variations are designed for flat clustering algorithms. One other important aspect is that most studies consider binary clustering

problems only and do not assess the behavior of the methods in multi-cluster domains. Most of real-world applications are multi-cluster problems, in special application that aim at organizing document collections.

The second research gap is related to the attainment of semi-supervised topic hierarchies. Topic hierarchies have proven to be an efficient way of organizing document collections, but there are no applications involving semi-supervised clustering algorithms to build such structures. The usage of unsupervised learning algorithms result in hierarchies that organize the document collection according to statistical measures and document similarities. However, they do not consider the context of the user and his/her expectations. So, topic hierarchies that attend user-based specifications may require the manual construction of such structures, which is considerably costly (Paukkeri et al., 2012). In this scenario, the usage of a semi-supervised approach for construct topic hierarchies would fill this gap and bring significant contribution to knowledge organizing and sharing.

In this work, both of the research gaps mentioned above are investigated. We introduce a new approach for semi-supervised hierarchical clustering that employs innovative procedures for user querying and the selection of informative cases for user interaction. Two algorithms are presented following this approach: *Hierarchical Confidence-based Active Clustering (HCAC)* and *Hierarchical Confidence-based Active Clustering with Limited Constraints (HCAC-LC)*. These are general purpose clustering algorithms and present a dominant performance in clustering textual data.

We also present SMITH (SeMI-supervised Topic Hierarchies), a framework for constructing topic hierarchies using semi-supervised hierarchical clustering algorithms. This framework allows the iterative knowledge insertion by the user during the clustering process. In the SMITH framework, topic hierarchies are obtained by the semi-supervised clustering of the document collections and the further labeling of the obtained clusters. The SMITH framework extends the TopTax *methodology* (Moura et al., 2008a), which presents significant results in organizing document collections and helping domain specialists in knowledge management. However, in some cases, the topic hierarchies extracted through TopTax were not in accordance with the user's expectations. Also, SMITH instantiates TopTax by proposing a more defined set of activities in each step.

1.2 Objectives

The main objective of this work is to propose a complete solution for generating **topic hierarchies** about **restricted domains** that incorporate the **user's knowledge** during the process. This knowledge must be provided by the user in an iterative way.

Given the research gaps mentioned above, we can break this main objective in two objectives, as follows:

- **Objective 1**: Providing a **semi-supervised hierarchical clustering algorithm** which is efficient in clustering textual data. By efficient, we refer to a clustering algorithm that: (i) minimizes the human effort by **selecting the most informative cases** for user intervention; and (ii) provides interpretable **user queries** that introduce enough information to correctly guide the hierarchical clustering process. By considering these two aspects, this semi-supervised clustering algorithm must present better performance in clustering results, according to objective and subjective criteria, than other unsupervised and semi-supervised clustering algorithms.

- **Objective 2**: Providing a framework to create **semi-supervised topic hierarchies** for restricted domains. These hierarchies should be in accordance to the user's expectations over the organization of the document collections. This framework extend the TopTax methodology by allowing the incorporation of user's knowledge through **semi-supervised hierarchical clustering** algorithms.

The topic hierarchies to be generated must efficiently **aid the management of textual datasets** and, consequently, the knowledge contained in these collections. In this work, the efficiency of a topic hierarchy is related to its representativeness to the domain, intuitiveness and easy navigation.

The user interaction must be possible through a **limited supervision** to the semi-supervised hierarchical clustering process. By limited, we mean the supervisory action of the user over a small part of the dataset, providing information about documents or clusters, depending on the constraint to be considered. In order to make this limited supervision efficient, methods for optimizing it, as ***active learning approaches***, should be investigated. These methods should search for selecting the cluster or documents that are more willing to receive constraints, i.e., the most informative cases.

1.3 Hypothesis

The first research hypothesis of this work is that topic hierarchies are an efficient way of representing the knowledge expressed in document collections. Thus, topic hierarchies are useful in organizing the information contained in these documents.

The second hypothesis states that the incorporation of the user's knowledge during the construction of topic hierarchies provides more intuitive topic hierarchies. This would lead to constructing topic hierarchies that better fit the user's expectations over the problem domain. Consequently, the management of the documents collections is eased.

Our third hypothesis considered during our investigation is that it is possible to efficiently incorporate the user's knowledge to a clustering process through constraints.

1.4 Organization

This work is divided in five more parts. In Chapter 2 we present a description about the Text Mining process and the construction of topic hierarchies. In this same chapter, we discuss the TopTax framework, which is extended in this work. In Chapter 3, we discuss the semi-supervised clustering state-of-the-art, presenting a classification of the methods, the main works in the literature and pointing some research gaps and directions. Then, in Chapter 4 we introduce two innovative semi-supervised hierarchical clustering algorithms that were developed during our investigation: Hierarchical Confidence-based Active Clustering (HCAC) and Hierarchical Confidence-based Active Clustering with Limited Constraints (HCAC-LC). These two algorithms fill some research gaps present in the literature and are used in SMITH, our methodology to extract semi-supervised hierarchical topic hierarchies. This methodology is presented in Chapter 5, along with a case study. Finally, in Chapter 6 we present the conclusions of our investigation and point out the directions for some future work.

Chapter

2

Text Mining and Topic Hierarchies Extraction

In a scenario where great part of the data is available in a textual format, the Text Mining process emerges as a powerful tool in knowledge management. Text Mining can be defined as a set of techniques and processes that discover innovative knowledge in documents (Hearst, 1999; Ebecken et al., 2003). In this sense, the objective of Text Mining processes is to search for patterns, tendencies and regularities in documents written in natural language.

We can say that *Text Mining* is a specialization of the *Data Mining* process. The main difference between these two processes is that while conventional Data Mining processes deal exclusively with structured data, Text Mining processes inherently deal with unstructured data (Weiss et al., 2005). In this work, we consider unstructured data the one that does not follow any format pattern, while structured data follows some format, such as attribute-value matrices.

The Text Mining process is typically developed in a cycle. In the end of the process, the user obtains the knowledge about the analysed data. This process can be instantiated according to the requirements of each application. For example, applications that aim at obtaining an efficient organization of textual information usually employ methods to obtain clusters, as well as methods to obtain descriptors for the clusters in the pattern extraction step. On the other hand, processes that aim at obtaining an automatic classification of documents obtain efficient classification models that relate new documents to a set of classes previously known. In this work, the Text Mining process is instantiated with the objective of extracting topic hierarchies from document collections, in a spirit similar to the TopTax methodology (Moura et al., 2008a).

In this chapter, we first describe the Text Mining process. Then, we highlight how

each of its steps are instantiated in the TopTax methodology, which is extended in this work.

2.1 The Text Mining Process

The Text Mining process, as a specialization of the Data Mining process (Fayyad et al., 1996), can be divided in five steps: (i) Problem Identification; (ii) Pre-processing; (iii) Patterns Extraction; (iv) Post-processing; and (v) Knowledge Usage. The cycle formed by these steps can be observed in Figure 2.1. Each of these steps are discussed in the next sections of this chapter.

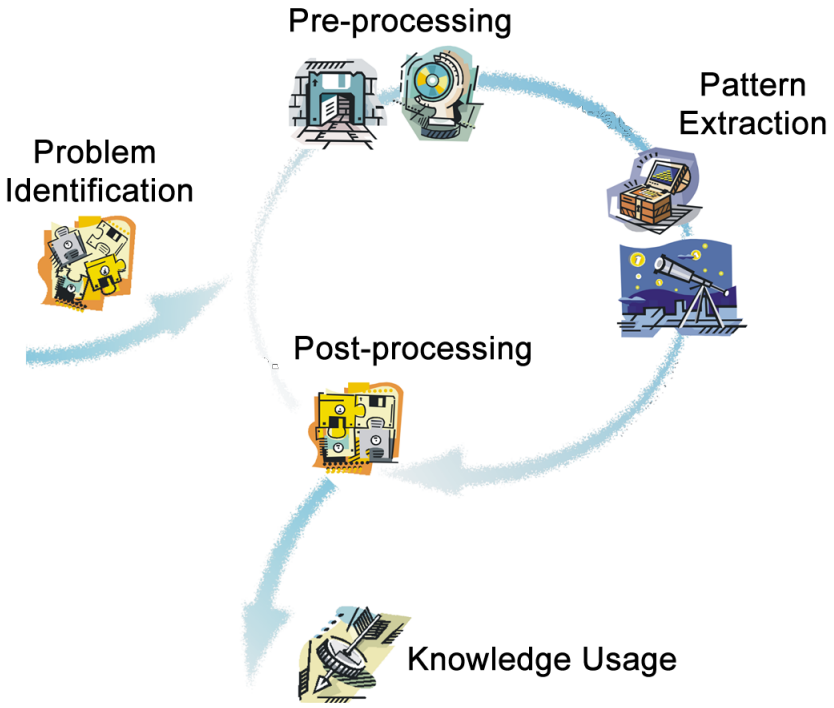

Figure 2.1: Steps of the Text Mining process. Adapted from: Rezende et al. (2003)

2.1.1 Problem Identification

In this step, the user must delimit the scope of the problem and define the objective of the application of the Text Mining process. Basically, the analyst defines the document collection that will be employed, what is expected from the data analysis and how the result of the analysis can be used. Rezende et al. (2003) define four questions to be answered in this step, which are adapted from the CRISP-DM process (Shearer, 2000):

- Which are the main goals of the process?
- Which performance criteria are important?

- Does the extracted knowledge need to be comprehensible by humans or a black-box model is appropriate?
- Which relation between simplicity and precision the extracted model needs to present?

The decisions taken in this step will guide the next steps in the process and may have impact in the performance of the application of the process. In this sense, an indispensable activity in this step is to study the problem domain. This activity must be carried out assisted by an specialist in the problem domain and helps the decision taking in the subsequent steps (Fayyad et al., 1996).

In this step we also define the document collection to be employed. The user must select documents that are relevant to the domain and to the application of the extracted knowledge. These documents may be collected from diverse sources, as electronic books and articles, as well as Web documents. This is a critical activity, since documents may not be available in an adequate format, as non-digitalized documents or unlabeled data in processes that involve activities of classification of documents.

2.1.2 Pre-processing

The Pre-processing step is one of the most time consuming steps during the process. In this part of the process, the data are structured in a format adequate to the knowledge extraction. This transformation depends on the methods that will be used, but typically involves activities such as data cleaning and volume reduction.

As with other types of data, the analyst must assure the reliability, non-redundancy and the balancing of the collection. In this sense, Moura (2009) cites a series of actions that may be taken assisted by an domain specialist, such as:

- Removing repeated documents;
- Balancing the document collection by resampling. This balancing should consider the topics covered by the documents;
- Reducing the number of documents, when the objectives of the process allow;
- Verifying the existence of a structure in the documents (such as sections in scientific papers and html pages), in order to use this information in the final structure of the data collection;
- Analysing the size of the document collection. We should verify the necessity of normalizing the weights attributed to the terms in function of the size of the texts.

The first task of the documents preprocessing is the documents standardization. Two main procedures have to be carried out in this task: (i) standardizing the documents

format and (ii) cleaning the contents. Since documents can come from diverse data sources, there can be documents in different formats, as *pdf* and hypertext. To assure that all documents are equally accessible and manipulable, we should transform all documents to a plain text format, with no formatting characters. Then, all unnecessary content from these documents should be removed. This content includes punctuation and mathematical symbols, in cases that they are not interesting. These removed tokens do not provide useful information and would imply in extra computational cost to the process (Manning et al., 2008).

After filtering and standardizing, the documents in the collection are represented in a structured way so that common data mining algorithms can be applied. The most common structure for textual datasets is the vector space model (Salton, 1989; Weiss et al., 2005). In this model, each document is modelled as a vector (d_i) and each position in this vector is a term of the document (t_i). The vectors for all documents form an attribute-value matrix, as presented in Table 2.1. The last column in the matrix corresponds to the class of the documents, if the process deals with labeled data. Besides being extremely simple, the attribute-value matrix has proven to be an efficient solution to documents representation, as reported by Bekkerman and Allan (2004).

Table 2.1: Example of attribute-value matrix

	t_1	t_2	...	t_M	*Class*
d_1	a_{11}	a_{12}	...	a_{1M}	c_1
d_2	a_{21}	a_{22}	...	a_{2M}	c_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
d_N	a_{N1}	a_{N2}	...	a_{NM}	c_N

In Text Mining, terms are terminological units that represent concepts in the document collection (Cabré et al., 2001). In this work, we consider the terms present in the documents as the attributes that describe the collections. However, not all words in the collection are considered relevant to represent the problem domain. So, we should select the most representative words. First, words that do not represent useful information to the process are discarded through the elimination of *stopwords*. These words are not relevant to the analysis of the documents and are usually prepositions, pronouns, articles, interjections, among other grammatical structures. In some domains, it is also possible to obtain *domain stoplists*. These stoplists consist in a set of words that, in that specific domain and according to the objectives of the Text Mining process, can be eliminated from the collection.

After this, we should identify similarities between the meaning of different words, such as morphological variations and synonyms (Manning and Schtze, 1999). In this sense, it is possible to reduce words to their root or stem through *stemming* processes (Krovetz, 1993); to their lemma through *lemmatization* (Arampatzis et al., 2000); to a generator substantive through *substantivation* (Gonzalez et al., 2006); or even using dictionaries or

thesaurus (Sparck-Jones and Willett, 1997; Conrado et al., 2012).

These approaches are sufficient to identify uni-grams, or simple terms. Considering these simple terms, it is possible to search the collection for compound *terms* or *n-grams*. Compound terms are the terms formed by more than two or more elements that appear consecutively in the collection but have one single semantic meaning (Manning and Schtze, 1999; Conrado and Rezende, 2008; Conrado et al., 2012). Compound terms introduce to the document representation the notion of context of occurrence, since it considers the order of appearance of the terms to form the compound terms. Several researches have argued that compound terms may improve the quality of the results in several applications involving documents, such as categorization (Carvalho and Cohen, 2006; Tesar et al., 2006), clustering (Beil et al., 2002; Fung et al., 2003) and information retrieval (Koster and Seutter, 2003).

In the attribute-value matrix, each cell is filled by a measure that relates a document and a term. Two measures are usually adopted to relate a document and a term. The simplest measure is the *term frequency (TF)*, which counts the number of occurrences of a term in a document. The other measure is the *tf-idf* measure (Salton and Buckley, 1987), which weights the term frequencies by their distribution along the document collection. In this sense, it is introduced the Inverse Document Frequency (IDF), which provides smaller weights to those terms that occur in a large number of documents. The tf-idf of a term i in a document j, considering a collection with N documents, is calculated according to the Equation 2.1:

$$TFIDF_{i,j} = f_{ij} * log\left(\frac{N}{DF_i}\right) \tag{2.1}$$

The attribute-value matrix is inherently sparse and presents high dimensionality (Forman, 2003). This may turn the Text Mining process extremely expensive in computational terms or even impracticable. Moreover, it may negatively affect the results of some machine learning algorithms. Thus, it is necessary to select the most relevant terms of the document collection, making the set of terms more concise but not less representative than the original set.

In order to reduce the dimensionality of simple terms, there are two possible approaches: ***attribute extraction*** and ***attribute selection***. These approaches are general purpose and can be applied in diverse types of data. The process of **attributes extraction** generates a new set of attributes, smaller than the original one, using a mapping function between the representations (Wyse et al., 1980). The main drawback of this approach is that the new attributes do not maintain a direct correlation with the problem domain, making the interpretation of the models more difficult (Dash and Liu, 1997). The main techniques in this approach are the Principal Component Analysis (PCA) (Jolliffe, 2002), the Latent Semantic Analysis (Landauer et al., 1998) and the word clustering

(Slonim and Tishby, 2000).

Attribute selection, on the other hand, is related to the selection of a subset of the original attributes set according to some criteria. Since the attributes are not modified and maintain a direct relation with the problem domain, the comprehension of the obtained model is better than using attribute extraction (Liu et al., 2005).

The attribute selection algorithm to be employed depends on the existence of labels in the data. In labeled datasets, supervised feature selection algorithms may be employed, such as: Information Gain, χ^2, odds ration and probability rate (Forman, 2003).

On the other hand, unsupervised feature selection algorithms may be employed in unlabeled datasets. Nogueira (2009) presents a comparison of some unsupervised feature selection algorithms for Text Mining. The most commonly used method is the Luhn's method (Luhn, 1958). This method is based on the Zipf's Law, also known as Principle of Least Effort (Zipf, 1949). According to the Zipf's Law, the frequency of occurrence of some events is related to an ordering function.

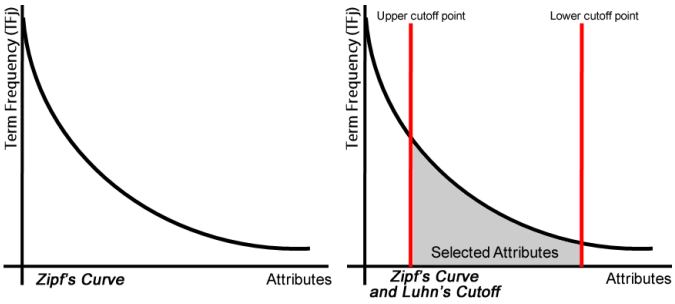

Figure 2.2: Luhn's cutoff selection.

In document collections, by counting the uni-gram frequencies and ordering the resulting histogram in decreasing order forms a Zipf's curve, as described in Figure 2.2. Over this curve, the Luhn's method suggests the detection of two cutoff points: an upper point and a lower point. High frequency terms are discarded as, in general, they tend to appear in most of the documents. Thus, high frequency terms do not help to differentiate the contents of the documents. On the other hand, low frequency terms are considered very rare and also do not provide discriminatory information. Besides being a simple method, in (Nogueira et al., 2008a) the authors claim that the Luhn's method performs as well as other more complex methods. Other examples of unsupervised feature selection algorithms are Salton's method (Salton and Buckley, 1987), Term Variance (Liu et al., 2005) and Zone Scored Term Frequency (Nogueira and Rezende, 2009).

To reduce the number of compound terms, Conrado and Rezende (2008) state that there are several statistical methods that can efficiently discover representative compound terms in the collection. These methods analyse the representativeness of the joint occurrence of the terms that compose the n-gram. For example, in the work of Tan et al. (2002), a bi-gram is generated if at least one of the terms that compose it is frequent in a docu-

ment; in the work of Rossi et al. (2012), the compound term is generated if its frequency and a measure of relation are higher than threshold values; and in the work of Moura et al. (2008c) the number of compound terms is reduced by analysing the redundancy among the frequencies of the words that compose the terms.

After reducing the dimensionality of attributes, the data is, then, ready to be presented to pattern extraction algorithms. This procedure is discussed in the next section.

2.1.3 Pattern Extraction

Since the problem is delimited and the data is an adequate format, the process moves to the Pattern Extraction step. The tasks to be performed in this step are defined according to the objective of the process. As in the Data Mining process, it is possible to divide the main pattern extraction tasks in Text Mining processes in two main groups: *predictive* and *descriptive*. *Predictive tasks* generate a model derived from the dataset in order to predict one or more features of interest (labels). *Descriptive tasks*, on the other hand, produce new knowledge based on the textual datasets through patterns that can be interpreted by humans (Kantardzic, 2003).

Predictive tasks require *supervised* machine learning algorithms. These algorithms, also known as inductors, require a set of training examples with known label values (Mitchell, 1997). This kind of algorithms is divided in two subcategories: classification algorithms and regression algorithms. Classification refers to the process where the class attribute has discrete values, while regression is the process where the class attribute has continuous values. The main application of classification in Text Mining processes is the automatic documents categorization (Sebastiani, 2002).

Descriptive tasks, on the other hand, employ *unsupervised* algorithms and deal with unlabeled datasets. The main tasks in this branch of Text Mining are association rules, data clustering and text summarization.

Association rules are logical relationships inferred between correlated data from one or more datasets (Agrawal and Srikant, 1994; Han et al., 2007). Association rules indicate relationships between two disjoint set of attributes L and R. These relationships are represented in the form $L \rightarrow R$, where L is called the antecedent ant R is the consequent of the rule. In Text Mining, the number of associations obtained tends to be very large, since every word in a document is a candidate attribute. As one document may have a great number of words, a great number of relationships may be found. Thus, the rules must be post-processed. This task is carried out in the Post-processing step of the Text Mining process, which is further discussed in this chapter.

Data clustering, or simply clustering, aims at grouping data in significant groups according to some similarity measure (Jain et al., 1999). Basically, this process searches for groups such that data inside one group are as similar as possible, while objects in

different groups present the maximum possible dissimilarity. The result of a clustering algorithm can be hierarchical, with multiple levels of aggregation, or flat, with one single level. Flat clusters are isolated, i.e., they form disjoint groups of elements. On the other hand, hierarchical clustering algorithms form groups with a hierarchical structure. Clustering algorithms may be interesting in applications that aim at text mining, exploration or summarization (Peltonen et al., 2002; Jain, 2010). Besides, in information retrieval applications, document clustering is employed since the user, while recovering one document, may also be interested in similar documents, which would be in the same cluster (Chakrabarti, 2003).

Data summarization is a task that aims at obtaining a compact description for a dataset (Chandola and Kumar, 2007; Jorge and Pardo, 2010). In Text Mining applications, the main application of summarization is to generate an automatic summary of documents. In this sense, documents are first clustered according to the topics they are related to. Then, the main information of these documents are extracted, forming one representative document. The aim is to reduce the content of a document collection without losing the main information.

As we have seen above, predictive tasks require supervised algorithms, while descriptive tasks require unsupervised algorithms. Besides these categories of algorithms, there is the category of semi-supervised algorithms, which combines both supervised and unsupervised approaches. It is possible to find semi-supervised algorithms for both predictive and descriptive tasks. These algorithms are employed in scenarios where there is information about classes or clusters to only a part of the data. This information can be provided in form of explicit labels or through constraints about the data distribution in groups or classes. In general, in knowledge extraction processes, there are more unlabeled data than labeled data, since unlabeled data are less expensive and easy to obtain (Zhu, 2005; Huang and Lam, 2009).

The great motivation to use semi-supervised algorithms is to incorporate, in a single learning algorithm, the information from unlabeled and labeled data. This allows more efficiency in the process and a smaller loss of important information about the domain. To predictive tasks, unlabeled elements are used during the training process of the classifiers. Some well known algorithms in this paradigm of classification are Co-Training (Blum and Mitchell, 1998) and the transductive SVM (Joachims, 1999). With respect to descriptive tasks, diverse initiatives have emerged in the last years to explore semi-supervised clustering algorithms. These algorithms are employed in this work and are discussed in details in Chapter 3.

2.1.4 Post-processing and Knowledge Usage

The patterns extracted should be analysed and interpreted according to the problem domain. It is taken into account the representativeness of the knowledge, the novelty contained in the results and the usage of the knowledge.

Regarding the validity of the extracted knowledge, the user verifies if the obtained patterns are in accordance with the problem domain configurations and are applicable in the context of the initial objective. For example, in applications that involve predictive tasks, the user can calculate measures related to the precision in predictions involving new data, such as error rate, precision and recall. The evaluation of descriptive tasks is more complex, since the evaluation criteria varies according to the objective of the process. Thus, sometimes there may not exist objective measures to evaluate the result of the process and the evaluation is done by a domain specialist - subjective evaluation.

Another important aspect is related to the knowledge comprehensibility. The algorithms of pattern extraction may generate a very large quantity of patterns. In this scenario, the comprehension of the patterns by the user is difficult and requires the application of mechanisms that select the most interesting patterns for the user (Silberschatz and Tuzhilin, 1995). In Text Mining applications, this is a recurrent problem due to the typical high dimensionality and, thus, demands a special attention (Carvalho et al., 2007).

One activity that may consistently help the user in this step is data **visualization**. Card et al. (1999) define this activity as a computer-based interactive visual representation of the data to improve the cognition. Various types of graphs and diagrams may be explored by the analyst along the process, facilitating the comprehension of the results and assisting the decision taking process. The visual analysis may, for example, indicate the failure of the decisions taken in some of the previous steps and the need for taking different decisions. Moreover, some of the final forms of the extracted knowledge are easily represented by some methods of information visualization. For example, the results of a hierarchical clustering algorithm generate a tree that can be visualized, for example, using directory trees or hyperbolic trees (Lamping et al., 1995; Marcacini, 2008; Alencar et al., 2012).

At the end of the Text Mining process, if the user detects problems in the extracted knowledge, then he/she should return to the step to be corrected and reapply the process. Otherwise, the knowledge is ready to be deployed.

2.2 Extraction of Topic Hierarchies and the TopTax methodology

Hierarchical topic structures obtained from textual collection are known as topic hierarchies or topic taxonomies. By definition, topic hierarchies are collections with controlled vocabulary organized with ancestral relationships between its members (Garshol, 2004).

In this work, we consider a topic hierarchy as a hierarchical classification formed by a set of descriptors extracted from a document collection.

Some work in the literature explore the formation of topic hierarchies for the management of document collections. One of the first initiatives in this area is the work of Miiller and Dorre (1999), which introduced the TaxGen environment. In TaxGen, topic hierarchies were created from the comparison of documents through their linguistic characteristics. A hierarchical agglomerative clustering algorithm was employed to construct the hierarchical cluster structure. After that, the hierarchy was pruned in its seventh level, in order to enable the visualization of the entire hierarchy.

In another work, Lawrie and Croft (2000) perform a flat clustering process for a set of documents from a restricted domain. For each cluster generated, a hierarchical clustering algorithm was applied, extracting hierarchies from homogeneous groups of documents. This work was extended by Lawrie et al. (2001), who proposed the usage of a probabilistic model of the vocabulary. This model was based on the problem of dominant sets in graphs, in order to detect terms that better describe the topics.

More recently, Dupret and Piwowarski (2005) explore the induction of topic hierarchies from the term similarity matrix through singular value decomposition (SVD). This decomposition leads to the identification of the concepts present in the document collection. The authors argue that this set of concepts is smaller than the set of terms but is a sufficient representation of the collection. Gil-García and Pons-Porrata (2008) explore the concept that documents may belong to more than one topic. In their approach, cluster overlapping is allowed and a graph-based clustering algorithm is employed. Moreover, their method allows the dynamic insertion of documents, which is desirable in scenarios where the document collection is constantly evolving. The dynamical aspect of document collections is also treated in the work of Tang et al. (2008). Their approach uses an initial topic hierarchy, which is refined in order to obtain hierarchies that better fit the data. The main application is in the determination of dynamic group profiling.

In this thesis, we extend the TopTax methodology (Moura et al., 2008a) to construct topic hierarchies. This methodology was developed in the Laboratory of Computational Intelligence (LABIC) of the University of Sao Paulo[1] in collaboration with the author of this work.

The TopTax methodology is an instantiation of the Text Mining process to support the organization and management of the information available in document collections about restricted domains, which is also the objective of this work. In this methodology, topic hierarchies are extracted from document collections through a semi-automatic process.

TopTax employs unsupervised hierarchical clustering algorithms to obtain a hierarchical structure of groups of documents. Then, each cluster is labeled with the topics that

[1] http://labic.icmc.usp.br/

describe its contents through a descriptors extraction process. The main objective of the TopTax methodology is to support the domain specialist in organizing the documents that describe the knowledge domain under a topic hierarchy, as well as to support the construction of this topic hierarchy. In Figure 2.3, we present the steps of the TopTax methodology.

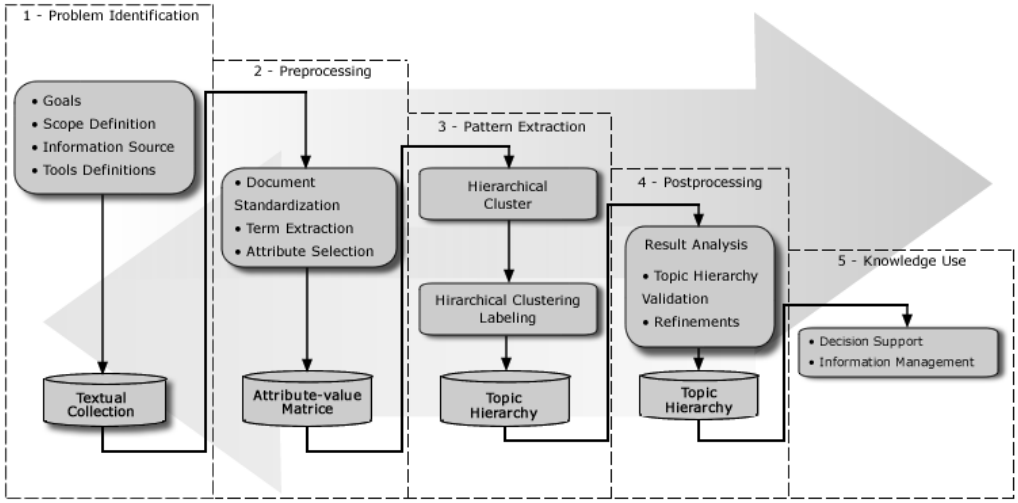

Figure 2.3: Steps of the TopTax methodology. Adapted from: Moura et al. (2008a)

In the **Problem Identification** step, the TopTax methodology suggests the definition of the objectives of the problem to be treated, as well as the identification of the document collection to be employed. The user must select documents which are potentially useful for the specific domain. If the problem domain to be treated has non-textual data (e.g., movies or images), the user must find metadata that describe these data. All these available data are then converted to plain texts with no formatting, assuring a standard, simple and manipulable format for all the documents.

In the next step, **Pre-processing**, the user must assure the significance of the document collection to be explored. In this sense, he/she must verify the need of modifications in the document collection, such as the removal of unnecessary documents. Then, the document collection must be structured under a attribute-value matrix, in order to make the document collection manipulable by propositional pattern extraction algorithms.

To construct the structured representation of the documents, first a term extraction process is carried out. In order to identify similar terms, the TopTax methodology suggests the usage of stemming algorithms; lemmatization algorithms; substantivation algorithms; or mapping variant terms - synonyms, abbreviations, acronyms and alternative orthographies - to one term through a thesaurus. The selection of the term normalization approach to be adopted depends on the user needs and the scenario of the application. A deep discussion on these term normalization methods can be found in the work of Conrado et al. (2012).

Once the terms are normalized, the TopTax methodology suggests the removal of the

general and domain stopwords. As discussed in Section 2.1.2, these words do not bring important information to the process and must be removed. From the remaining single terms, the TopTax methodology searches for compound terms, or n-grams, where n indicates number of single terms that compose the compound term. The user must chose a proper number of maximum n terms that a compound term must have. The bigger the value of n, the more computational time will be spent in the search for compound terms. Moreover, the higher the value of n, the more possibilities of simple term combinations will exist. Then, a bigger number of terms will be generated, increasing the dimensionality of the attribute-value representation.

Using the detected terms, the TopTax methodology constructs the attribute-value matrix. In each cell of this matrix, the methodology employs the absolute frequency (Term Frequency - TF) of a given term in a given document. This matrix is, however, inherently sparse and has high dimensionality. This can be extremely harmful for the knowledge extraction process. In this sense, the next step of the TopTax methodology consists in the application of methods to select the most representative attributes, in order to reduce the dimensionality of the matrix. As discussed in Section 2.1.2, the reduction of the dimensionality of attributes trough attributes extraction processes is not recommended in applications where the comprehensibility of the models is an important feature, which is the case of the TopTax methodology. Thus, TopTax suggests the selection of the the most representative attributes through an attribute selection process. Since it deals with unlabeled data, unsupervised attribute selection can be used. In the work of Nogueira et al. (2008a); Nogueira and Rezende (2009) it is possible to find an analysis of the main algorithms for unsupervised feature selection for Text Mining tasks.

The attribute selection task is the last one in the Pre-processing step of the TopTax methodology. The process advances, then, to the **Pattern Extraction** step. Since the main objective of TopTax is to provide a topic hierarchy, hierarchical clustering algorithms are employed in the Pattern Extraction step of the methodology. Unsupervised agglomerative algorithms may be employed in this step, such as the Complete-Link, Average-Link and the Single-Link algorithms.

In order to apply such hierarchical clustering algorithms, it is needed to form a distance matrix. So, the document $\times$ term matrix is converted to a document $\times$ document matrix. In this new matrix, each inner cell $d_{x,y}$ is filled by the distance between the documents x and y. In TopTax, it is suggested to employ the cosine distance function to calculate the distance between the documents (see Tan et al. (2005)). The cosine distance between two documents can be calculated according to the Equation 2.2:

$$d_{x,y} = 1 - cos_{x,y} = 1 - \frac{x.y}{\|x\|\|y\|} \tag{2.2}$$

In text mining applications, the cosine distance function varies in the interval $[0, 1]$,

where distance 0 indicates the highest similarity between two documents, while a distance 1 indicates the highest dissimilarity between two documents.

When the clustering algorithms are applied in this distance matrix, a hierarchical structure of clusters is generated. In order to transform a cluster hierarchy into a topic hierarchy, the TopTax methodology suggests a descriptors extraction process. Feldman and Sanger (2007) suggest that a good cluster descriptor consists in a very small set of terms that distinguishes the cluster from the others. This process can be considered as a supervised term selection process, considering that the clusters are the classes present in the collection (Weiss et al., 2005).

There are several algorithms in the literature for this intent. One well known example that may be employed in the context of TopTax is the FScore measure (Chu, 2003). For the calculation of the FScore, let us assume that the terms of the document collection $D = \{d_1, d_2, \ldots, d_n\}$ form a set $T = \{t_1, t_2, \ldots, t_m\}$. Also, let us assume a query expression $Q(t)$, where $t \in T$, over the document collection. So, $Q(t)$ retrieves a subset of D that contains the term t. Then, for each cluster C in the set of candidate pairs and each term t, a set of measures can be calculated:

- Accuracy(t,C): number of documents in D retrieved by $Q(t)$ that belong to C;
- Loss(t,C): number of documents in D that belong to C and were not retrieved by $Q(t)$;
- Noise(t,C): number of documents in D retrieved by $Q(t)$ that do not belong to C;
- Rejection(t,C): number of documents in D that do not belong to C and were not retrieved by $Q(t)$.

Using these measures, it is possible to calculate the FScore measure as an harmonic mean between precision and recall, as shown in Equation 2.3.

$$\begin{aligned} Precision(t,C) &= \frac{Accuracy(t,C)}{Accuracy(t,C) + Noise(t,C)} \\ Recall(t,C) &= \frac{Accuracy(t,C)}{Accuracy(t,C) + Loss(t,C)} \\ FScore(t,C) &= \frac{2 * Precision(t,C) * Recall(t,C)}{Precision(t,C) + Recall(t,C)} \end{aligned} \tag{2.3}$$

Using the FScore measure, it is possible to obtain a rank of the terms for each cluster. The descriptors of the cluster are the set of the top k ranked terms. The value of k must be decided according to the context (Bast et al., 2005) and should be empirically selected.

Other interesting algorithms for extracting cluster labels are the Robust Labeling Up Method (RLUM) (Moura and Rezende, 2010) and the SeCLAR algorithm (Santos et al.,

2010). Interesting reviews in this topic may also be found in the work of Treeratpituk and Callan (2006) and Moura et al. (2008b).

After this process, the obtained topic hierarchy must be validated. TopTax deals essentially with unlabeled data. According to Chang et al. (2009), the direct application of objective validation measures in unsupervised datasets is not trivial, since these measures are directed for labeled collections. On the other rand, a subjective evaluation process should be avoided, as it may contain a strong bias. Thus, in the **Post-processing** step, TopTax suggests to mix subjective and objective procedures in one evaluation. The idea is to collect objective measures from the user interactions with the topic hierarchy. For example, it is possible to give a set of documents for the user to search in the hierarchy and measure the search time, the number of clicks required and the the number of successes and failures achieved.

For this validation, TopTax suggests that the topic hierarchy should be exhibited in the form of a knowledge tree (de Souza et al., 2005). In such structure, the knowledge is represented through a tree, such that the more generic an element is, the higher is the level it is represented in the tree. Each node of a knowledge tree is identified with the terms that represent its content - as the set of descriptors extracted in the previous step of TopTax, characterizing the topic hierarchy. As discussed in Section 2.1.4, directory trees or hyperbolic trees may be used to represent a knowledge tree (Lamping et al., 1995; Marcacini, 2008; Alencar et al., 2012).

While navigating through the hierarchy, the user should detect the need for adjustments in the hierarchy, such as the pruning of the knowledge tree or even the edition if the topic hierarchy. If deeper modifications are needed, the process may return to one of the previous steps of the TopTax methodology. Otherwise, the knowledge represented in the topic hierarchy is ready to be used and explored by the user, in the **Knowledge Usage** step.

2.3 Final Remarks

Knowledge extraction from document collections became one important tool to people and corporations. This process brings useful and innovative knowledge from document collections, assisting the decision taking process. The Text Mining process is the intermediate element between textual data and knowledge. It extracts non-trivial information from the documents through patterns detected along the collection.

The Text Mining process is constituted by a sequence of five steps: Problem Identification, Pre-processing, Patterns Extraction, Post-processing and Knowledge Usage. Each of these steps is formed by a set of generic tasks, which can be instantiated according to the specificities of each application. This instantiation implies in, for example, selecting the appropriate machine learning algorithm to be used, or selecting the best representative

set of terms for a given collection.

Concerning the instantiations of the Text Mining processes for extracting topic hierarchies, in this chapter we discussed the TopTax methodology. The aim of this methodology is to provide an efficient way to organize the textual knowledge from document collections. This work has the TopTax methodology as the methodological basis to extract topic hierarchies. We extend the TopTax methodology by proposing SMITH in Chapter 5 of this thesis. While TopTax uses unsupervised clustering algorithms, in SMITH we use semi-supervised clustering algorithms, which considers the user's knowledge about the problem domain during the clustering process. This framework is introduced and explained in Chapter 5.

In the next chapter, we review the current state of the art of semi-supervised clustering algorithms. We discuss in details the classification of the algorithms in this area, as well as the main methods in each category.

Chapter

3

Semi-supervised clustering

Semi-supervised learning (Zhu, 2005; Chapelle et al., 2006) has been used in the last years in a wide spectrum of applications. These algorithms employ labeled (where the target or label is known a priori) and unlabeled data (where there is no information about the label of the data) in the same solution (Seeger, 2002). The reason for mixing these two kinds of data in a same solution is based on their nature. It is common sense that unlabeled data are much easier to obtain than labeled data. On the other hand, labeled data can provide much more information for a learner system than unlabeled data.

Existing applications of the semi-supervised learning paradigm typically fall in one of two cases: (1) to improve supervised classification through the usage of unlabeled data; and (2) to improve clustering algorithms using labelled data. The extraction of topic hierarchies in our framework requires the usage of clustering algorithms. Thus, in this work, we focus on semi-supervised clustering algorithms, which has been extensively explored in last years. For an overview of semi-supervised classification, we recommend the work of Zhu (2005).

In this chapter, we give an overview of the semi-supervised clustering research area. Such review of the area was not done before and is one of the contributions of this thesis to the state of the art. To this purpose, we present a classification of the current work in the area, discussing important sub-areas and aspects that we find relevant. Also, we discuss some results and point out some open questions that represent future research perspectives in the area.

3.1 Introduction to semi-supervised clustering

Semi-supervised clustering, also known as constrained clustering, can improve the clustering quality by employing external knowledge during the clustering process. Besides finding groups guided by an objective function, as in unsupervised clustering, semi-supervised algorithms also incorporate external knowledge conveyed in the form of constraints. These constraints can be directly derived from the original data (using partially labelled data) or provided by an user, trying to adapt the clustering results to his/her expectations (Dasgupta and Ng, 2010). The great challenge in semi-supervised clustering is to obtain sufficient and quality knowledge from a small amount of external information (i.e., a small amount of constraints), as the labeling of large amounts of data is expensive (Bilenko et al., 2004).

One of the first initiatives in semi-supervised clustering was the work of Talavera and Béjar (1999). The method proposed in their work is a variation of the ISAAC algorithm that adds background knowledge in hierarchical clustering processes, using a declarative approach. The great interest for research in semi-supervised clustering, however, emerged in the earlies 2000's, when Wagstaff and Cardie (2000) introduced COP-Cobweb, an algorithm that employs *pairwise constraints must-link* and *cannot-link*. Through these constraints, the user can indicate whether two instances must or must not belong to the same cluster. The authors have shown that these constraints could significantly improve the accuracy performance when compared to unsupervised methods.

Due to its simplicity and good results, pairwise constraints have been employed in a great number of works. For example, there are applications involving distance metric learning for clustering (Xing et al., 2003; Bilenko et al., 2004; Domeniconi et al., 2010), feedback over a clustering result (Cohn et al., 2003), application in hierarchical clustering (Davidson and Ravi, 2009; Hamasuna et al., 2011), active discovery of constraints (Huang and Lam, 2007, 2009) and propagation of constraints in a space-level context (Klein et al., 2002). A good summary of works that use pairwise constraints can be found in the survey of Davidson and Basu (2007).

However, efforts in semi-supervised clustering were not limited to exploring pairwise constraints. These constraints have some limitations and cannot be efficiently applied in some scenarios. For example, pairwise constraints are not suitable for semi-supervised hierarchical clustering as objects are linked over different hierarchy levels (Zheng and Li, 2011) and can carry limited information. Thus, in order to improve clustering quality in scenarios where pairwise constraints are not efficient, different approaches have been proposed. These approaches vary, for example, according to the constraints employed (such as initial seeds for the KMeans algorithm (Basu et al., 2002), cluster-level constraints (Klein et al., 2002) and graph-based methods (Wang et al., 2012b)). Another possible variation occurs in the selection of the instances or clusters to insert constraints (i.e.,

active or passive methods (Huang and Lam, 2009)). In general, these different approaches achieved performance equal or better than pairwise constraints.

In the remaining sections of this chapter, we discuss and explore these different approaches for semi-supervised clustering reported in the literature. We present the main methods and discuss the advantages and disadvantages of using the methods of each category.

3.2 Background

Before exploring the existing work on semi-supervised clustering, in this section we present some basic concepts on unsupervised and semi-supervised clustering that will be useful in the remaining of this work, as well as to delimit its scope.

Clustering algorithms aim at obtaining a set of groups (clusters) where elements in a same cluster are similar, whereas elements in different groups are the most dissimilar as possible. According to Jain and Dubes (1988), clustering algorithms can be firstly divided in terms of overlapping allowance: exclusive and non-exclusive algorithms. In exclusive clustering, each object must be assigned to only one cluster. Non-exclusive algorithms, on the other hand, may assign one object to more than one cluster through a membership function. Fuzzy clustering algorithms (Yang, 1993; Baraldi and Blonda, 1999) and some probabilistic clustering algorithms (Bock, 1996; Xu and Wunsch, 2005) are non-exclusive. In this work, however, we focus on exclusive semi-supervised clustering algorithms. Exclusive clustering algorithms attend the needs of our process of extracting topic hierarchies and are more commonly found in the semi-supervised literature. There are, however, some interesting works in semi-supervised non-exclusive clustering that may interest the reader (Pedrycz, 1985; Bensaid et al., 1996; Pedrycz and Waletzky, 1997; Pedrycz, 2004; Bouchachia and Pedrycz, 2006; Zeng et al., 2012).

On the exclusive clustering category, six groups of clustering algorithms can be pointed out (Zhong et al., 2011): (1) *hierarchical clustering*; (2) *partitional clustering*; (3) *density-based clustering*; (4) *grid-based clustering*; (5) *model-based clustering*; and (6) *graph-based clustering*. The first five groups refer to non-relational (propositional) algorithms, whereas the last one refers to a class of relational clustering methods (Motta et al., 2013). As hierarchical and partitional clustering are the most common categories (Lin and Chen, 2005), most of the semi-supervised algorithms cited in this chapter belong to one of these categories. However, some representative algorithms from the other four categories will be also discussed.

To differentiate partitional and hierarchical clustering approaches, let us consider a set of objects $X = \{x_1, x_2, \ldots, x_n\}$. Partitional clustering algorithms aim at obtaining a set of k partitions of the dataset $C = \{C_1, C_2, \ldots, C_k\}$, such that $C_1 \cup C_2 \cup \ldots \cup C_k = X$ and for every pair of clusters i and j ($i \neq j$), $C_i \cap C_j = \varnothing$. Hierarchical clustering algorithms,

on the other hand, create a hierarchical decomposition of the set of objects according to some criteria. This hierarchy is formed through the discovery of nested partitions of the dataset. A partition C_i is nested into another partition C_j if all of its elements are also elements of C_j (i.e., $C_i \subset C_j$).

There are two approaches for discovering nested partitions in hierarchical clustering: *agglomerative (bottom-up)* and *divisive (top-down)*. Agglomerative hierarchical clustering starts with a set of clusters C where each object is a cluster ($C = \{C_1, C_2, \ldots, C_n\}$). At each step, the best pair of clusters according to some criteria, C_i and C_j, is selected to be merged. This procedure is repeated until every element is in one cluster. On divisive hierarchical clustering, the strategy adopted is the opposite. The algorithm starts with every element in the same cluster and this set is recursively divided in subclusters until every cluster contains only one element.

Unsupervised clustering algorithms use only intrinsic information about the data to find clusters, as the distance between elements according to some distance metric. On the other hand, semi-supervised clustering algorithms employ external information to guide clusters discovery. This information is conveyed as a set of m constraints $\delta = \{\delta_1, \delta_2, \ldots, \delta_m\}$. These constraints can be derived from labeled data or be provided by an human expert, which formalizes his/her background knowledge in form of constraints.

Algorithm 1: Basic framework for semi-supervised clustering

Input: $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$: set of objects; $\delta = \{\delta_1, \delta_2, \ldots, \delta_m\}$: set of constraints; $\mathbb{W} = \{W_{(1,2)}, W_{(1,3)}, \ldots, W_{(1,n)}, \ldots, W_{(n,n-1)}\}$: weight criteria for pairs of objects; $\mathbb{C}_i = \{C_{i,1}, C_{i,2}, \ldots, C_{i,j}\}$: current set of clusters;

Output: $\mathbb{C}_f = \{C_1, C_2, \ldots, C_k\}$: final set of clusters;

```
repeat
    Check the best unsupervised clustering decision D on C_i according to W;
    Consult the set δ of constraints;
    if There is a subset δ' of constraints that can improve D
    then
        D' = Optimize(D, W, C_i, δ');
        C_{i+1} = Update(C_i, D');
    end
    else
        C_{i+1} = Update(C_i, D);
    end
    C_f = C_{i+1};
until stopping criterion is reached;
```

A basic framework for semi-supervised clustering is presented in Algorithm 1. During the clustering process, in each clustering decision, semi-supervised algorithms consult the δ set in order to acquire information about the data and check whether this information can help the clustering decision in that point. This decision considers the set of clusters C_i, present in that clustering step, the unsupervised clustering decision D (the one that would

be taken without any external information) and the weight criterion W (for example, a distance between each pair of examples x_a and x_b). From this scenario, it is selected a subset δ' of constraints that can improve the clustering decision, i.e., constraints which information is related to clusters in C_i and that could lead to a better clustering decision.

According to the basic framework presented in Algorithm 1, semi-supervised approaches differ in three main aspects: (i) the *Update* procedure; (ii) the assembly of the δ set; and (iii) the *Optimize* function. The *Update* procedure updates the set C of clusters according to the clustering decision. This procedure varies according to the kind of the algorithm (one of the six groups of exclusive clustering cited above).

The other two procedures (assembling δ and defining the *Optimize* function) delimit three important aspects of the semi-supervised algorithms: (i) how to add information to the clustering process; (ii) to which cases - instances or clusters - the user should provide information; and (iii) how the algorithm deals with the inserted information.

The process of assembling the δ set defines the constraint-related aspects of the semi-supervised clustering algorithm. One thing is the kind of constraint that better fits the objective of the application. In some applications, the comprehensibility of the constraints by the humans is an essential issue. Thus, for these cases, it is advisable to use instance-level constraints or to find appropriate ways for representing cluster level constraints, as it is easier to understand and analyse sets of constraints than sets of clusters.

Another aspect that must be determined before assembling the δ set is how the algorithm will select the instances or clusters to be presented to the user to insert information. For example, the algorithm can randomly choose instances or clusters or employ some active learning approach to select proper examples to enquire the user about.

When solving the *Optimize* function, the algorithm makes a clustering decision that considers the set δ' of constraints, the weight function W and the current set of clusters C_i. There are several approaches for obtaining a clustering decision D. For example, the algorithm can modify its distance function or the objective function to make a decision that considers the information provided by δ'.

All of these aspects will be further discussed in the following sections of this chapter. We will also give an overview of the semi-supervised clustering area, propose a classification of the different existing semi-supervised clustering approaches and point out some important aspects that should be considered by researchers when applying and developing semi-supervised clustering algorithms.

3.3 Semi-supervised Clustering Approaches

In this work, we introduce a classification of the semi-supervised approaches according to three aspects: (i) how the user interacts with the algorithm; (ii) how the algorithm uses the knowledge inserted by the user; and (iii) the level of the information inserted by

the user. This classification is illustrated in Figure 3.1. Every semi-supervised method can be classified according to these three aspects. In the next sections, we discuss these categories, presenting the main work in each of them.

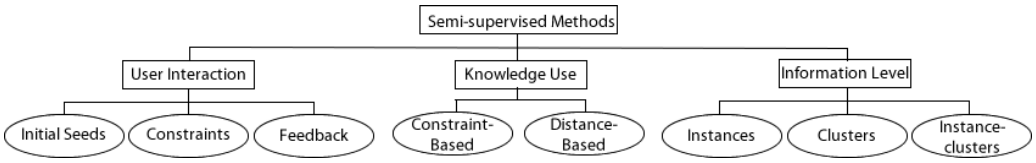

Figure 3.1: Classification of semi-supervised clustering algorithms

3.3.1 User interaction

One of the main decisions in semi-supervised clustering, according to Jain (2010), is the way that the user interacts with the clustering process in order to specify his/her knowledge domain. The appropriate way of user interaction in a clustering process can vary according to the problem domain. So, it is possible to find different approaches for user interaction in the literature.

In this work, we classify semi-supervised methods in three different categories of user interaction, as proposed by Zhong (2006). The first category is formed by the seed-based methods, which use the external information before starting the clustering process to indicate initial representatives for cluster formation. The second category contains the constraint-based methods, which employ external information during the clustering process for limiting cluster formation. Finally, the third category contains the feedback-based method, which permits the addition of information after an initial clustering process, based on its results. These three categories will be discussed in the following sections.

Seed-based methods

On seed-based methods, the external information is provided over a set of objects in order to initialize the cluster representatives. Most of the work in this category is based on the KMeans algorithm (MacQueen, 1967).

The best-known seed-based methods are the ones proposed in the work of Basu et al. (2002). In that paper, the authors presented two algorithms that are modifications of the KMeans algorithm: Constrained KMeans and Seeded KMeans. For both of these algorithms, the user provides a set of points to be used as initial seeds for the KMeans algorithm. In the Constrained KMeans algorithm, the cluster membership of these elements in the set of initial seeds is not modified along the iterations. On the other hand, in the Seeded KMeans, the cluster membership of the initial seeds can be modified along the iterations. Both approaches presented a significant improvement on clustering performance, when compared to unsupervised clustering algorithms and also to COP-KMeans

(Wagstaff et al., 2001), a constraint-based method that employs pairwise constraints. These methods also provide robustness to noise and outliers.

The major problem in both Constrained KMeans and Seeded KMeans is that they require the user to indicate an initial seed to every cluster to be formed during the clustering process. Wang et al. (2007) smoothed this problem by proposing two new methods: Farthest-Seeded-KMeans (FS-KMeans) and Splitting-Seeded-KMeans (SS-KMeans). FS-KMeans and SS-KMeans use what the authors call "incomplete knowledge domain", in which the user does not provide initial seeds for all the clusters in the dataset. On the FS-KMeans, the algorithm iteratively selects the farthest point from all other centroids to be the new cluster centroid, until all cluster centroids are assigned. On the SS-KMeans, the dataset is initially divided in clusters according to the cluster centroids indicated by the user. Next, iteratively, the cluster with highest sum of squared error is divided in two clusters, until the desired number of clusters is achieved.

Other interesting variation of these methods is the DualSeededKMeans (Hu et al., 2012). This algorithm is designed for document clustering and employs a dual seeding scheme: documents seeding and feature seeding. The user is queried to provide documents as initial seeds for the clustering process (one document for each cluster to be formed). This method considers that each cluster has a topic, which consists in at least one feature. These features are obtained indirectly by the document that the user provided as a seed for that cluster.

Constraint-based methods

In constraint-based methods, external information limits the formation of clusters during the clustering process. The clustering process is adjusted in order to satisfy the constraints provided over some objects or clusters.

One of the first constraint-based methods (and also one of the first semi-supervised clustering methods) is the COP-Cobweb algorithm (Wagstaff and Cardie, 2000), already discussed in this chapter. In this algorithm, the authors present the well known pairwise constraints must-link and cannot-link. Must-link constraints indicate that two objects must be clustered in a same group, while cannot-link constraints indicate that two objects cannot be in the same group.

The usage of pairwise constraints lead COP-Cobweb to achieve a significant improvement in clustering performance. Besides, must-link and cannot-link constraints are easy to implement and can be easily expressed by a human. These features make pairwise constraints extensively used in constraint-based methods (Wagstaff et al., 2001; Cohn et al., 2003; Xing et al., 2003; Bilenko et al., 2004; Shental et al., 2004; Huang and Lam, 2007; Pelleg and Baras, 2007; Domeniconi et al., 2010). However, the information carried by these constraints is somehow limited, as they only introduce information about a pair of instances at a time.

In another work, Klein et al. (2002) extend pairwise constraints, proposing the Constraint Complete-Link algorithm. This algorithm considers an inductive spatial generalization of must-link and cannot-link constraints, affecting not only objects present in the constraints provided by the user, but also objects near to these objects. This algorithm considers that objects affected by a must-link constraint are near in the space, while objects involved in a cannot-link constraint are far from each other. So, objects which are near to one of the objects involved in pairwise constraints are also near to (in case of must-link constraints) or far from (in case of cannot-link constraints) the other object involved in the pairwise constraint. This propagation is possible through a spatial distortion, applying weights to the original distances using a shortest-path algorithm among all pair of objects.

Another framework for pairwise constraints propagation is presented by Li et al. (2008). Their framework uses a semidefinite programming approach to formulate a mapping function to introduce a distortion in the feature space. Their approach maps two instances involved in a must-link constraint, while instances involved in cannot-link constraints are told to be orthogonal. These constraints are considered during a kernel learning process, which is taken as s semidefinite programming problem.

The efforts in constraint-based methods, however, are not limited to the pairwise constraints and their variants. For instance, the algorithm proposed by Davidson and Ravi (2005) employs, besides must-link and cannot-link constraints, two other constraints that determine the minimum distance and the maximum distance among two objects in a same cluster (minimum separation constraint). These constraints are used in a modification of the KMeans algorithm and the authors prove that the combination of these constraints is computationally viable.

In another constraint-based method, Kumar et al. (2005) proposed the usage of relative comparison constraints. The proposed algorithm, Semi-Supervised SVaD, use triple comparisons of the type "x is nearer to y than to z". This algorithm uses Spatial Variant Dissimilarities (SVaD) in order to model dissimilarities among objects. These constraints achieved significant improvements to the clustering result. The relative constraints are also used in other works in the literature and are sometimes referred as "must-link-before" constraints (Bade and Nurnberger, 2006; Zheng and Li, 2011).

In this thesis, we introduce clustering algorithms that use constraints concerning clusters instead of objects: HCAC (Hierarchical Confidence-Based Active Clustering) and HCAC-LC (Hierarchical Confidence-Based Active Clustering with Limited Constraints). Through cluster-level constraints, the user indicates the next pair of clusters to be merged in an agglomerative hierarchical clustering process. These algorithms are explained in detail in Chapter 4 of this thesis.

Feedback-based methods

The feedback-based methods perform an initial clustering process and then adjust the resulting clusters based on the user information, which is provided in form of new constraints.

For example, the work of Cohn et al. (2003) performs the addition of constraints over an initial partitional clustering of documents. This method allows the user to answer questions like "this document does not belong to this cluster", "move this document to that cluster" and "these two documents must (or cannot) be in the same cluster". These constraints are used in an algorithm based on the Expectation Maximization (EM) algorithm using an approximation of the Bayes probability theorem to model the clusters. The work of Huang et al. (2006) explores these same constraints, also based on the EM algorithm and incorporating user-provided feedback over an initial clustering in order to refine the clusters model through local weights learning.

In the work of Huang and Mitchell (2006), the user can interact with the clustering obtained through the probabilistic model SpeClustering. The user can indicate whether an object belongs to a cluster or not. In a posterior work, Huang and Mitchell (2008) extended the usage of feedback to a hierarchical clustering process, allowing the user to indicate the necessity of addition, deletion, fusion and splitting of clusters, as well as the modification of cluster membership of examples.

In Dubey et al. (2010), a KMeans-based algorithm using user feedback, Cluster-Level Interactive KMeans algorithm (CLIKM). In this algorithm, the user can indicate, based on a KMeans clustering result, the membership of a object to a given cluster, as well as adjusting the cluster centroids according to its domain knowledge.

3.3.2 Information level

In a semi-supervised clustering method, external information can be related to different levels. They can involve from the more specific elements of the process, the objects, to the more general elements, the clusters. In this sense, in this work we divide semi-supervised algorithms in three categories, according to its information level: instance-level, cluster-level, and instance-cluster-level.

Instance-level

Instance level constraints specify mandatory relationships between instances or objects. Most of the constraint-based methods employ this kind of constraints, mainly due to their simplicity and comprehensibility by the humans. These constraints, however, do not assume any information about clusters and thus introduce less information than considering class labels. Instance-level constraints provide information only about the spatial distribution of the objects.

Examples of instance-level constraints are: the user can indicate whether two instances must or must not be grouped in the same cluster (examples are must-link and cannot-link constraints (Wagstaff and Cardie, 2000)); whether two instances may or may not be grouped in a same cluster (Ares et al., 2009) (e.g., may-link and may-not-link constraints); and whether a given instance is nearer to a second instance than to a third one (as the relative constraints, also referred as must-link-before (Kumar et al., 2005; Bade and Nurnberger, 2006; Zheng and Li, 2011)).

Cluster-level

Cluster-level constraints allow the process to exploit information in a higher level of the clustering process. As each constraint is related to a larger number of objects, cluster-level constraints can obviously convey more information than the instance-level constraints. However, dealing with instance-level constraints may not be trivial for a human user.

Using cluster-level constraints, the Constrained Complete-Link algorithm (Klein et al., 2002) allows the user to indicate, during the hierarchical agglomerative clustering process, whether the roots of the next proposed merge must be clustered together. This may be considered as an adaptation of the must-link and cannot-link pairwise constraints.

In another work, Huang and Mitchell (2008) proposed constraints that allow the user to indicate the necessity of cluster removal, addition, moving and merging in hierarchical clustering algorithms. These constraints are added by the user over an initial hierarchical clustering with maximum depth two. Considering this initial clustering and the constraints provided by the user, the authors propose the use of an adaptation in cascade of the SpeClustering algorithm (Huang and Mitchell, 2006) to train a classifier that associates the examples to the new clusters. The SpeClustering algorithm is also employed in the work of Pham et al. (2008), where the authors propose an algorithm that allows the user to indicate cluster removal in an initial hierarchical clustering process.

In another work that employs cluster-level constraints, Balcan and Blum (2008) proposed a method that allows split and merge of clusters. The authors present a simple and theoretical model, which is similar to the learning model through queries equivalence.

The feedback-based algorithm Cluster-Level Interactive KMeans (CLIKM) (Dubey et al., 2010) allows the user to adjust the cluster centroids as a feedback over a KMeans clustering result. During the clustering process, the user can iteratively indicate a preferred weight vector for each cluster, in the form of a set of constraints. These cluster weights are considered during the cluster reassignment process.

Instance-cluster-level

It is also possible to give information which concerns objects and clusters. The most common constraints in that category allows the user to indicate the membership of one object to one cluster. This constraint is, in general, introduced through a feedback over an initial clustering or as previously labeled data.

Works that use information over objects and clusters provided by feedback over initial clustering can be found in the work of Huang and Mitchell (Huang and Mitchell, 2006, 2008) and Pham et al. (2008), previously discussed in this chapter.

In the work of Eick et al. (2004), the user provides information about the cluster membership of some examples of the dataset. Based on this information, probabilistic clustering algorithm are applied, maximizing the class purity degree inside the clusters. In another work, Finley and Joachims (2005) propose the usage of an initially labeled set of examples, i.e., to which the cluster membership is already known. These examples are used to train an SVM classifier and to learn the distance function, which will be used to cluster the remaining examples.

On the Cluster-Level Interactive KMeans (CLIKM) algorithm (Dubey et al., 2010), besides the cluster-level constraints discussed in Section 3.3.2, the user can also provide feedback to the KMeans algorithm pointing out a instance-cluster dependency, in the form of a set of constraints. Each instance-cluster constraint assigns one element to one cluster. Given these instance-cluster constraints and considering the cluster centroids updates proposed by the user, the algorithm pose a penalty for breaking a instance-cluster constraint. This penalty is proportional to the distance between the instance and the centroid of the cluster it is assigned to in the constraint.

3.3.3 Knowledge usage

The information provided by the user to the semi-supervised clustering can be used in several ways. Here, we adapt the classification of Huang and Lam (2009) and divide the semi-supervised clustering algorithms according to the knowledge usage in two categories: constraint-based and distance-based methods. In some cases, the same method can be placed in both categories, using the constraints both to modify the objective function and to learn a new distance measure.

Constraint-based

Constraint-based methods directly employ user-provided information, modifying the evaluation objective function of the clustering process in order to satisfy the given constraints. For example, the work of Wagstaff and Cardie (2000) modifies the objective function such that the must-link and cannot-link constraints provided by the user are not broken. The must-link set is fully explored before considering the distances in clustering

decisions. Also, in every clustering decision, the cannot-link set is consulted in order to check whether the proposed clustering decision is allowed. This approach is equivalent to considering the distances between objects in a must-link constraint as zero, and objects involved in a cannot-link constraint as infinity (Davidson and Ravi, 2009).

Other approaches also employ pairwise constraints in order to modify the objective function. Some of them allow the constraints breaking by imposing a cost. This cost is considered in the objective function. Generally, the objective function in these methods has the form $MIN(D, W + \alpha)$, where MIN is a minimization function, D is the clustering decision, W is the weight function among pairs of objects and α is the cost of breaking any constraints in the δ set.

The HMRF-KMeans algorithm (Basu et al., 2004b), a KMeans-like iterative algorithm based on Hidden Markov Random Fields, employs an objective function like this. This algorithm is composed by two steps, in an EM scheme. The cluster assignment to all data points is computed, considering penalties for breaking must-link and cannot-link constraints and using iterated conditional modes to compute an approximation to the objective function optimum.

Also, in the Constrained KMeans algorithm (Basu et al., 2002), the cluster membership of the initial centroids given by the user is always kept along the iterations. In all the iterations of this algorithm the cluster representatives are initialized considering the initially indicated centroids. This way, the objective function is directly influenced, since it aims at minimizing the distance of the remaining data points to the cluster centroids.

Distance-based

Distance based methods enhance the clustering quality through learning measures of distortion over the data space using the constraints provided by the user. The methods in this category use the information provided in form of constraints indicating similarity and dissimilarity among pairs of examples to learn the distance concept.

Some distance measures have been used in distance learning for semi-supervised clustering (Basu et al., 2004b; Vu et al., 2012): string-edit distance (Bilenko and Mooney, 2003), KL divergence using gradient descent (Cohn et al., 2003), Euclidean distance modified by shortest-path algorithm (Klein et al., 2002) and Mahalanobis distance. This last one is the simplest and most common approach and consists in considering the problem of obtaining a similarity function as a Mahalanobis distance on the form $d(x, y) = ||x - y||_A^2$, where A is a parameter matrix. The values of the A matrix are modified along the iterations in order to optimize the adequacy of the distance function to the constraints. The main drawback of this approach is that it only allows the model of linear relationships among features (Maggini et al., 2012). The work of Xing et al. (2003) follows this approach, using must-link and cannot-link provided by the user to indicate whether two elements are near or not. The authors address this problem as a convex optimization

problem, in order to obtain results without problems of local optimum. A similar approach can be found in the work of Kim and Lee (2002), which explores the usage of a gradient descent search technique to adapt the Mahalanobis weight matrix according to the given constraints.

The MPC-KMeans algorithm (Bilenko et al., 2004) also uses a Mahalanobis-like distance function. This algorithm consists in a modification of the traditional KMeans algorithm that employs pairwise constraints and metric learning. The constraints are relaxed, allowing the violation of some constraints in cases that this would result in clusters with higher cohesion. The distance learning is performed for each cluster, obtaining more cohesive clusters and with different shapes. This algorithm follows the EM approach. On the E-step, each object is clustered such that the sum of the distance of this object to the centroid of his cluster and the penalty for the possible violation of constraints for that attribution is minimized. Next, on the M-step, a distance learning process for each cluster is performed, in order to learn the distance that better fits for that dataset. Mahalanobis-like distance learning is also used in Redundant Component Analysis (Bar-Hillel et al., 2003) and regression (Zhang et al., 2003).

Another way to determine the distance relations with the knowledge conveyed by the constraints is to employ a boosting-like solution. These approaches try to find an optimal distance measure based on the given constraints. At each iteration of boosting-based clustering algorithms, a data representation is created and it is used by the clustering algorithm in order to obtain an appropriate cluster assignment (Liu et al., 2007). In the following iteration, the procedure is then reapplied and the distance measures are modified trying to satisfy all the unsatisfied constraints of the previous iteration. This procedure is repeated until most of the constraints are satisfied.

A boosting-like approach is presented in the BoostCluster framework (Liu et al., 2007), which can be applied with any clustering algorithm. At each clustering step, this framework proposes a new data representation by minimizing the inconsistency between a kernel similarity matrix and the given pairwise constraints. A subspace is obtained through eigenvalue-decomposition weight matrix function is learned by penalizing the violation of must-link and cannot-link constraints. Similar approaches, which also merges weak hypotheses in a boosting approach, can be found in the literature (Hertz et al., 2004; Sublemontier et al., 2011; Wang et al., 2012a).

Another common approach in distance-based methods is the kernel-based. These methods commonly transform the data using a nonlinear function Ψ (Kulis et al., 2009). Usually, these methods use a kernel matrix K in which each entry (i, j) corresponds to the result of the kernel function $\kappa(a_i, a_j) = \Psi(a_i) \cdot \Psi(a_j)$.

In the SCKMM (Semi-supervised Clustering Kernel Method based on Metric learning) method (Yin et al., 2010), an objective function is constructed from pairwise constraints given by the user. The parameters of a Gaussian kernel are further estimated. Using this

objective function, a constraint-based KMeans algorithm is used to solve the constraints violation and to cluster the data. Next, distance learning methods are applied in order to enhance the data separability for the clustering.

In another work, Kulis et al. (2009) proposed a method that learns low-rank positive semi-definite kernel matrices using Bregman matrix divergences as the distance function. This method linearly scales in the number of data points and quadratically in the rank of the input matrix. Other approaches use the intrinsic information from the data and, based on must-link and cannot-link constraints, performs metric learning as an optimization problem (Basu et al., 2004b; Kulis et al., 2005; Baghshah and Shouraki, 2010; Wang et al., 2012a).

Another recent distance-based method is based on Similarity Neural Networks (SNNs) (Maggini et al., 2012). The SNN networks are feedforward Multi-Layer Perceptron trained to learn non-linear similarity measures for pairs of examples given pairwise constraints. The SNN model is similar to symmetric functions, maintaining its basic properties: symmetry and non-negativity. The clustering algorithm is based on the KMeans algorithm, and the initial centroids are chosen by backpropagation on the input layer.

3.3.4 An analysis of the semi-supervised clustering approaches

As in all problems of machine learning, choosing the proper semi-supervised clustering approach for solving a problem depends on the problem domain. The approaches differ in aspects that may turn one approach more appropriate to one given kind of problem than to others.

Concerning the user interaction, the main difference between approaches is in what point of the clustering process the external information is employed: before the beginning of the clustering process (initial seeds); during the clustering process (constraint-based); or after an initial clustering process (feedback-based). Methods that employ initial seeds are basically based on the KMeans method and, thus, are not appropriate to problems that demand an hierarchical structure from the clustering process (as, for example, problems of organizing object collections). On the other hand, feedback-based methods have an additional cost due to the need of generating an initial cluster structure before asking the user's opinion.

Among the different approaches for knowledge usage, distance-based methods achieve, in general, a better performance in clustering quality. These methods can generate clusters of different forms, that fit in different kinds of data. However, distance-based methods have a computational cost that is greater than the cost of most of the other clustering algorithms. Moreover, distance-based methods may have some parameters that have to be tuned. For example, kernel-based methods are highly sensitive to the parameters value and tuning these parameters is not an easy task (Yin et al., 2010).

Finally, the different levels of information addition also have specifics that may turn each one applicable to different scenarios. The first difference concerns amount of information carried by each constraint. The information of instance-level constraints are, in general, limited to pairs or triples of instances. Cluster-level constraints, on the other hand, may affect a larger number of elements, since they are related to groups of instances. Thus, each cluster-level constraint can carry more information than one instance-level constraint. Due to this nature, according to the uncertainty theory (Garner, 1962), the correct delimitation of clusters in a dataset may be achieved with less cluster-level constraints than instance-level constraints.

The second difference between the different levels of constraints that can be pointed out relies on the practicality of imposing the constraints. Instance-level constraints are much easier to the user to pose than cluster-level or instance-cluster-level constraints. The interpretation of single instances requires much less effort from the user than the interpretation of clusters. So, constraints involving clusters may require a bigger expertise in the problem domain from the user.

Finally, the third difference concerns the computational cost involved in posing different levels of information. There is not much difference, in average, between methods based on instance-level constraints and methods based on cluster-level constraints. However, most of the methods based on instance-cluster-level constraints have an intrinsic additional cost, since they are feedback-based methods and require an initial clustering procedure.

3.4 Constraints flexibility in Semi-supervised clustering

The semi-supervised clustering algorithms described in the previous sections do not allow the constraint violation during the clustering process. The resulting clustering must satisfy all constraints imposed to it. This scenario can be defined as clustering with hard constraints. However, in many real-world applications, there may be some noise in the constraints provided to the semi-supervised clustering algorithms, specially when the number of constraints is high (Pelleg and Baras, 2007). In the cases where there is the need to satisfy all the constraints, the clustering process can be intractable and the clustering algorithm would return an empty partition as a clustering result. For example, Wagstaff et al. (2001) shows that a simple cannot-link constraint incorrectly defined can compromise the clustering results of the Constrained KMeans algorithm. In general, the proximity values between objects are drastically distorted in the presence of few noisy constraints.

To overcome these limitations, approaches which are more flexible were developed. These approaches allow the violation of constraints when these constraints can harm the results (Bilenko et al., 2004; Law et al., 2004; Davidson and Ravi, 2005; Pelleg and Baras,

2007; Zeng and Cheung, 2012). This scenario is known as clustering with soft constraints. In this case, the objective is to obtain the partition of the data that agrees the most with the constraint set, minimizing the number of constraints violation. The results of experimental evaluation reported in the literature indicate that clustering algorithms that employ soft constraints are more robust in scenarios with a large number of constraints. This makes clustering algorithms with soft constraints interesting in situations where there is uncertainty associated to the given constraints (Covoes et al., 2013).

In general, semi-supervised clustering approaches with soft constraints are based in the KMeans algorithm. In these approaches, the objective function is adapted to insert a penalty to each constraint violation. Algorithms that employ this strategy verify how beneficial the violation of one constraint can be. Thus, the constraints that drastically affect the geometrical properties of the data (for example, by linking two very different objects or by separating two very similar ones) are more likely to be broken during the clustering process. A similar approach is followed by semi-supervised clustering algorithms based on metric learning, discussed in Section 3.3.3, as the MPC-KMeans (Bilenko et al., 2004). These algorithms allow an indirect violation of the constraints that compromise the cohesion of the clusters.

The Constrained Vector Quantization Error (CVQE) algorithm, proposed by Davidson and Ravi (2005), is one of the main algorithms with soft constraints. In CVQE, two terms are inserted in the KMeans objective function to consider (i) the cost of violating the must-link constraints and (ii) the cost of violating the cannot-link constraints. This function is described as follows:

- Let $c_{ml} = (a, b)$ be one must-link constraint involving objects a and b. If this constraint is violated, then $l_a \neq l_b$, where l_i is the label of the cluster associated to object i. The cost of violating $c_{ml} = (a, b)$ is given by the distance between the centroids of the clusters l_a and l_b; and

- Analogously, let us consider $c_{cl} = (a, b)$ as a cannot-link constraint involving objects a and b. If this constraint is broken, then $l_a = l_b$. The cost of violating the constraint $c_{cl} = (a, b)$ is given by the distance between the centroids of the clusters l_a and l_n, where l_n is the nearest cluster to l_a, $l_a \neq l_n$.

Along the iterations of the CVQE algorithm, the objects that are not involved in any constraint are associated to the nearest cluster, according to the KMeans traditional process. On the other hand, objects that are involved in some constraint are associated to a cluster considering the cost of violating these constraints. To this end, all combinations of cluster association are verified. The cluster association that minimizes the objective function is selected. In some situations, the cost of violating a constraint is lower than the quadratic error obtained by maintaining the constraint, leading the CVQE algorithm to disregard that constraint.

One of the main disadvantages of the CVQE algorithm is the high computational cost to verify all combinations of cluster association. Due to this operation, the CVQE algorithm presents a quadratic computational complexity, with respect to the number of clusters $O(k^2)$. This makes CVQE very expensive in scenarios with a large number of clusters. One variant of the CVQE is the LCVQE algorithm (Linear CVQE), proposed by Pelleg and Baras (2007). This algorithm uses an alternative version to verify the constraints that can be violated, described as follows:

- Initially, all objects are associated to the nearest centroid, according to the conventional KMeans process.
- For each broken must-link constraint $c_{ml} = (a, b)$, three cases for cluster association are verified: (1) the value of the objective function maintaining the constraint violation; (2) the value of the objective function when associating objects a and b to cluster l_a; and (3) the value of the objective function when associating objects a and b to the cluster l_b. The algorithm LCVQE selects the case which minimizes the objective function.
- For each broken cannot-link constraint $c_{cl} = (a, b)$, two cases are verified. In the first case, the algorithm verifies the value of the objective function while maintaining the constraint violation. In the second case, assuming that l_n is the label of the cluster associated to objects a and b, the label of the object that is nearer to the cluster l_n is maintained. The other object is associated to the nearest neighbor cluster. Again, LCVQE selects the case that minimizes the objective function.

In Covoes et al. (2013), it is presented an extensive experimental comparison of CVQE and LCVQE. The LCVQE algorithm produces cluster partitions with similar accuracy to CVQE, but with a lower computational cost. Besides, the LCVQE algorithm obtains clustering solutions with less constraints violation than CVQE. On the other hand, the CVQE algorithm is more robust when there is a large number of noisy constraints.

3.5 Active Learning in Semi-Supervised Clustering

In all of the semi-supervised approaches described in the previous sections, one of the main factors for achieving good results is the choice of the cases (instances or clusters) to which the user provides information about. Depending on the level of the information to be inserted, different kind of queries are posed. The addition of information to more representative cases tends to add more relevant information to the process, i.e., information that can efficiently guide the formation of good cluster partitions. Besides, the better the choice of the cases to query the user, the less information the user has to add. Consequently, the lower is the effort required from the user.

Davidson et al. (2006) define two parameters to measure the utility of the information provided by the user to a clustering algorithm: informativeness and coherence. Informativeness is related to the quantity of information contained in the set of constraints which the clustering algorithm can not determine by itself. Coherence, on the other hand, measures the concordance between the constraints and the given distance measure. Thus, good sets of constraints present high informativeness and high coherence and can provide more quality information to the clustering process.

The simplest strategy of case selection to generate constraints for the clustering process is by selecting instances or clusters at random. This strategy can lead to a great negative bias in the results, given that the selected cases may not be representative for the clustering process. Another simple approach is to transfer to the user the responsibility of choosing the cases to add constraints, without providing any previous information. In most contexts, this strategy is extremely expensive, since browsing all examples or clusters and detecting the most representative cases is not a trivial task.

In this sense, automatic methods for selecting better cases to receive constraints are very important to improve the clustering quality. One example are the *active learning* algorithms. The aim of active learning is to automatically select the best examples to learn from, performing better with a smaller effort. Thus, the objective of the active learning methods is to choose the best examples - the ones with more useful information for the learning process - and present them to the learning algorithm. Combining active learning algorithms and semi-supervised learning algorithms is natural. In semi-supervised learning the learning algorithm explores what is known about labelled data. In active learning, on the other hand, the algorithm explores the non-trivial aspects of these data (Settles, 2009).

Active learning techniques have been explored in the context of semi-supervised learning, both for classification and clustering. In this work we focus on active learning algorithms for clustering activities. An informative overview of active learning algorithms for classification can be found in Settles (2009).

Hofmann and Buhmann (1998) present an active approach for clustering proximity data. According to Jain and Dubes (1988), features in proximity data are composed by similarity values between pairs of objects. In this method, the authors deal with sparse data caused by incomplete pairwise similarities - i.e., some similarities between pairs of objects are not known. The main motivation for this method is to deal with large datasets, since the exhaustive generation of proximity for all pairs of data is prohibitive. The active approach tries to estimate the relevance of the missing data through estimations from information in the given data, in order to pick some data pairs to actively querying the similarities. This has proven to be efficient in both flat and hierarchical clustering (Zoller and Buhmann, 2000).

In the work of Klein et al. (2002) an active learning algorithm is used to select instances

and clusters for building pairwise queries. The active algorithm performs a complete process of the Complete Link clustering algorithm. Considering that the distance between clusters in the distance matrix is always increasing along the iterations of the Complete Link algorithm, the active learning algorithm defines a maximum threshold distance α from which the algorithm can start querying the user and does not perform more than m queries (m defined a priori). Thus, the clustering process proceeds until finding a cluster merge in which the distance between the involved elements is equals or greater than α. From this point, the clustering algorithm starts asking the user whether the roots of the next proposed merge must or must not be on the same cluster. Based on the answer, the clustering algorithm imposes a must-link or a cannot-link constraint.

In Basu et al. (2004a), the active learning process is carried out in the initialization of the clustering algorithm. The active learning process starts by randomly choosing one example from the dataset and putting it in one set S. Then, iteratively, the farthest point from the set S is selected. This process finishes when S has k elements, where k is defined a priori. The user is then asked to introduce pairwise constraints involving the elements in the set S.

The Active Fuzzy Constrained Clustering algorithm (AFCC) (Grira et al., 2008) aims at minimizing a competitive agglomeration cost function. The objective function of the clustering algorithm is modified in order to consider both the similarity between pairs of objects and the pairwise constraints. The active learning approach used selects candidates for pairwise constraints provided by the user. The principle of the method is to lead the user to add constraints to define clusters which are neither compact nor well separated from their neighbors. A well defined cluster is detected through the fuzzy hypervolume measure (FHV) (Gath and Gev, 1989). The FHV of a cluster is proportional to its spatial volume and inversely proportional to the concentration of its data near to its center. At each iteration of the clustering process, data items at the boundary of the least well defined cluster are selected. For each of these items, the user is then asked to add pairwise constraints involving an item and the closest item from the closest cluster. Results indicate that AFCC can significantly improve clustering results with few constraints.

In the work of Huang and Lam (2009) is proposed a gain function to chose the examples to add constraints. Based on an intermediate clustering result, the algorithm provides pairs of documents such that the clustering algorithm achieves a local optimum in the next iteration. The clustering algorithm is probabilistic and term-to-term dependencies are calculated from the constraints provided by the user. Terms that co-occur frequently in must-link constraints present high probability of cohesion. Thus, the selection of pair of documents to query the user is done according to following equation: $\Omega^* = max(F(\Omega, \Theta, \gamma))$, where Ω^* is an optimal set of pairs of documents, Ω is the generated pair of documents, Θ are the current cluster attribution discovered by the semi-supervised clustering algorithm and γ is a set of probability of term co-occurrence.

Another proposal is presented by Zhao et al. (2009). The authors propose the Constrained DBScan, an adaptation of the DBScan algorithm (Ester et al., 1996). The Constrained DBScan algorithm considers pairwise constraints in its process. To elicit these constraints, the process tries to obey two main properties: (i) at least one point in each pair of underlying clusters must be in the set; and (ii) at least one constraint help to control the borders of each cluster. An active learning algorithm is employed in the search for this set using the concepts from the DBScan of core objects and border objects. In this algorithm, two parameters are used: Eps and $MinPts$. For each example in the dataset, in a radius of Eps there must be at least $MinPts$ examples. If an example has at least $MinPts$ examples in the radius Eps, it is taken as a core example. Otherwise, it is taken as a border example. In the first iteration of the active learning algorithm, one core object x is randomly chosen and is added to a set S. Then, two border points are selected: y, the nearest point to x, and z, the farthest point to S. The user is, then, queried do add pairwise constraints involving x and y, as well as x and z. In the following iterations, pairwise constraints are imposed to the farthest core point from S and all the points in S. Moreover, as in the initial iteration, two border points are selected and pairwise constraints are imposed involving these points and the new selected core point.

In a recent work, Marcacini et al. (2012) proposed the AL^{2}FIC algorithm - Active Learning to Frequent Itemset-based Text Clustering. In this algorithm, the user is asked to add constraints to cluster descriptors instead of documents. This algorithm considers that the cluster descriptors are composed by the frequent itemsets, where each item is a term that occurs in one or more documents in that cluster. An initial clustering is carried out adopting an Expectation Maximization approach under a vector-space model representation of the document collection. Then, this initial clustering is refined. In each iteration of this clustering refinement, the active learning approach selects the top frequent itemsets for each document cluster (according to the coverage and the proximity to the cluster center) to be presented to the user. The user is asked to select the most representative itemset for each cluster. Based on this selection, the algorithm reweights the cluster center in order to meet the user's expectations. This procedure is repeated until a maximum number of queries is reached. This approach has the advantage of not requiring any previous knowledge about cluster labels from the user, since users's feedback is provided with respect to the cluster descriptors and does not have document-level or cluster-level assumptions.

3.6 Some applications of Semi-Supervised Clustering

Semi-supervised clustering algorithms have been explored in a wide spectrum of applications. The improvement in clustering results achieved by these methods lead researchers to explore semi-supervised clustering algorithms in applications involving, for example,

textual data, images, genetic data and social network data.

In the context of clustering textual data, semi-supervised clustering algorithms are applied in order to provide a personal organization of document collections and to improve document retrieval. In the work of Kim and Lee (2000), the authors presented one of the first initiatives to employ semi-supervised clustering methods in organizing document collections. The proposed method is based on relevance feedback from the user. The algorithm randomly selects a document and takes this document as a search query. Then, the retrieved documents are presented to the user, who is asked to add constraints between each retrieved document and the original document. These constraints are then considered as the seeds of genuine cluster formation. As the clusters are subjectively formed, the documents in a given cluster may not share common words. Therefore, this approach can improve document retrieval by smoothing the problem of word mismatch.

In applications involving image data, semi-supervised clustering algorithms have been used mainly to allow a better image segmentation, as well as to image categorization and indexing of image databases. One of the first initiatives can be found in (Bensaid et al., 1996), where a semi-supervised Fuzzy CMeans using partially labeled data is used in the segmentation of magnetic resonance images (MRI). A similar approach can be seen in the work of Filipovych et al. (2011), where the authors use a KMeans based semi-supervised algorithm to image segmentation.

In the work of Chang and Yeung (2006), the authors use the Locally Linear Metric Adaptation (LLMA) algorithm, a semi-supervised clustering algorithm based on metric learning, in an image retrieval application. The semi-supervised algorithm is used to discover groups of similar images. The LLMA algorithm performs metric learning by a nonlinear global transformation and a linear local transformation. As a result, the algorithm achieved significant improvement in image retrieval when compared to other methods. In the work of Grira et al. (2008), the AFCC algorithm, explained in Section 3.5, achieved significant results in clustering image datasets. Recently, Lai et al. (2013) proposed the usage of an interactive semi-supervised clustering method based for indexing image datasets. The proposed method improves the quality of clusters of images by deducing pairwise constraints from the feedback provided by an user over an initial clustering process.

Semi-supervised clustering algorithms have also been employed with interesting results in bioinformatics applications. More specifically, significant results were achieved in the cluster analysis of genetic data. In the works of Zhu et al. (2005); Chung et al. (2006); Maraziotis (2012); McNicholas and Subedi (2012), the gene expression data is clustered using some labeled data samples which are previously known.

Recently, there is an emerging research field in exploring social network data. In the work of Ben Ahmed et al. (2013), a hierarchical clustering algorithm is used to group people based on their profile in the LinkedIn social network. The objective is to obtain

professional communities by employing quantitative constraints ranking. The user provides quantitative ranking of cluster-level constraints to indicate priority in cluster merge operations. Different criteria for group detection according to the user's profile were tested and relevant practical results were achieved in the analysis of the users behaviour in this social network.

In the context of this work, semi-supervised clustering algorithms are applied in order to organize document collections according to the content of the documents. In the literature, there are applications of semi-supervised clustering algorithms that aim at organizing documents in significant groups considering the user's domain knowledge. It is possible to find good results reported in the literature using semi-supervised clustering algorithms to organize textual data from emails (Huang and Mitchell, 2006, 2008), news (Ji and Xu, 2006; Huang and Lam, 2009; Zhao et al., 2012), scientific papers (Marcacini et al., 2012), software requirements (Duan et al., 2008), newsgroups (Ji and Xu, 2006; Huang et al., 2008; Zhao et al., 2012), web pages (Zhong, 2006; Bade et al., 2007), among other scenarios. These algorithms use different approaches for document clustering, as model-based approaches (Zhong, 2006; Huang and Lam, 2009), KMeans based approaches (Duan et al., 2008) and frequent-itemsets based approaches (Marcacini et al., 2012).

3.7 Open questions and perspectives in Semi-Supervised clustering

Semi-supervised clustering is a relatively new research field and most of the work in the area has been developed in the last fifteen years. As a result, there are some gaps in the research field that have not been suppressed. These gaps may indicate future directions in the research area. In this section, we briefly discuss some of the most important.

3.7.1 Constraints utility

One of the main issues of semi-supervised clustering algorithms is whether it is worthy to explore external knowledge to improve unsupervised clustering results. In general, semi-supervised clustering methods do not present significant additional computational cost when compared to unsupervised methods. However, eliciting constraints over part of the datasets has an intrinsic cost related to the user effort.

Besides, there is no consensus on how to measure the cost of the user effort, and it is common sense that it is important to minimize it. The minimization of the user effort is directly related to the quality of the information he/she provides to the clustering process - the better constraints are added, the fewer constraints are required. However, measuring the quality of a constraint set is not trivial and identifying constraint set properties that correlate with their utility is a difficult task (Wagstaff, 2006). In Davidson et al. (2006), an initial black-box analysis of such features is carried out and two indexes as proposed:

informativeness and coherence. As explained in Section 3.5, informativeness is related to the quantity of new information brought by the set of constraints, while coherence measures the concordance between the constraints and the given distance measure. The usage of these measures to elicit good queries significantly improved the clustering results. However, despite their importance, not much effort has been expended in determining other objective measures for constraints utility and this still is an open question in semi-supervised clustering.

3.7.2 Active learning

This issue is directly related to the problem of determining the constraints utility. Exploring active learning algorithms to detect good instances or clusters to add constraints has proven to be efficient in the literature. The usage of a proper active approach tends to reduce the number of constraints needed from the user to achieve a good cluster structure.

Algorithms using active learning algorithms for semi-supervised clustering are relatively rare. Moreover, most of the proposed active learning measures are designed to help eliciting instance-level constraints in KMeans-like algorithms. In addition, to the best of our knowledge, there is only one approach, proposed by Klein et al. (2002), specifically designed to actively elicit constraints during a hierarchical clustering process. We believe that considering the cluster structure in different levels of the hierarchy would considerably improve the clustering result. Given this lack of convincing active solutions to clustering algorithms other than KMeans-like, we believe that an interesting research line to be explored in the next years is the development of new techniques of active learning for semi-supervised clustering.

3.7.3 Constraints propagation

Except for some distance-based methods, most of the constraints imposed by the semi-supervised clustering algorithms have local impact. These methods influence the cluster membership of instances or clusters which are explicitly mentioned in the constraints that are posed. This imposes a severe restriction in the amount of information added by each constraint. In some cases, it could be interesting to extend this information to other instances or clusters, inferring some non-trivial constraints from the user's knowledge. For example, in a cluster border region, a cannot-link constraint between two instances could be extended to other instances near these two, as they probably will belong to different clusters as well.

The usage of some sort of constraints propagation lead clustering algorithms to achieve better cluster structures, as in the Constrained Complete-Link algorithm (Klein et al., 2002). Besides adding an additional computational cost to clustering algorithms, constraint propagation methods have the potential to reduce the amount of constraints re-

quired from a user, reducing his/her cognitive effort. Thus, the practical application of these methods and the lack of solutions in the literature turn this one of the points to be researched in semi-supervised clustering.

3.7.4 Ensemble of methods

Different clustering algorithms have different clustering biases, as they follow different objective functions. Due to this fact, different algorithms may generate different cluster structures from the same data. One possible solution to this problem is to make a consensual decision, considering that the information provided by different sources and the different clustering partitions are complementary. This combination is possible through an ensemble of methods (Topchy et al., 2003; Strehl and Ghosh, 2003; Fred and Jain, 2005).

There is very limited work in combining semi-supervised clustering algorithms. As the usage of different types of external knowledge bring different improvements to the clustering process, it could be interesting to explore the combination of clustering structures formed using different kinds of constraints. For example, using instance-level and cluster-level constraints could introduce in a consensual clustering process some specialized and general information. The initiatives in mixing different levels of information focuses in using them in a same clustering process. In Davidson and Ravi (2009), the authors have combined cluster-level and instance-level constraints in semi-supervised hierarchical clustering processes. Despite some combinations of these constraints are NP-Complete problems, the feasible combinations brought significant improvements to the clustering process.

Moreover, one of the biggest problems of constructing an ensemble of clustering methods relies on handling the combination of cluster structures with different number of clusters. In this sense, constraints could be explored to limit the cluster formation and to guide the formation of comparable cluster structures in different methods. This field has been recently explored by Forestier et al. (2010) and promising results were obtained. However, the results are limited to specific scenarios, requiring a deeper investigation.

3.7.5 Incremental clustering

Semi-supervised clustering algorithms, especially hierarchical clustering algorithms, have the limitation of the high computational cost. In general, semi-supervised hierarchical clustering algorithms have computational cost in time and space that are quadratic with respect to the number of documents.

This computational cost makes the algorithms hard to apply in two scenarios: (i) when dealing with large datasets; and (ii) when dealing with dynamic data streams. As the knowledge about a given domain evolves, new data naturally emerges leading to an

increment of the size of data collections. One example is the growing number of scientific papers as new discoveries are made. On the other hand, some previous knowledge may become deprecated with new discoveries. In this sense, data collection representations may degrade over time and become incapable of synthesizing the knowledge in these dynamic data collections.

According to Jain (2010), managing large and dynamic datasets is still a challenge to overcome. Static approaches need all the data collection to be reprocessed when the dataset changes. However, most of the previously existing data tends not to change cluster membership (Sahoo et al., 2006). Thus, re-applying the clustering algorithm over all this old data implies redundant reprocessing and unnecessary computational cost. Thus, techniques which are able to incrementally process modifications in the datasets, as the hierarchical clustering algorithms, are one of the tendencies in information organization.

To the best of the authors' knowledge, there is no clustering algorithm which can deal with both external information and incremental scenarios. Using incremental structures which are successfully used in unsupervised clustering algorithms, as the Suffix Trees (Zamir et al., 1997) and the term co-occurrence networks for document clustering (Marcacini and Rezende, 2010a), are potentially useful in the creation of semi-supervised clustering algorithms that can deal with dynamic and high dimensionality datasets. One possible solution is the creation of an initial cluster structure in a semi-supervised way and expanding it as new documents arrive.

3.7.6 Impact of incorrect constraints

Another controversial question in semi-supervised clustering is the robustness of the methods against noisy constraints. Most of the experiments with semi-supervised clustering algorithms simulate the human supervision and consider "ideal" users, i.e., users that do not insert wrong or contradictory constraints. In real-world applications, however, this is not always true, as the human supervisor is willing to make mistakes in the interaction with the clustering process.

For example, let us consider a semi-supervised clustering algorithm that employs pairwise constraints must-link and cannot-link over a dataset with two different concepts (clusters). There are three common mistakes that can be present in such an environment: (i) adding a must-link constraint involving two instances that belong to different concepts; (ii) adding a cannot-link constraint between two instances that belong to the same concept; and (iii) imposing conflicting constraints between objects - directly or indirectly.

These are common mistakes and may occur due to a variety of reasons, from misunderstandings about the problem domain to a simple incorrect interaction with the constraints-posing process. However, very few works perform an analysis of noisy constraints in semi-supervised algorithms. The exceptions are the works of Pelleg and Baras

(2007); Yoshida (2012); Covoes et al. (2013). These works, however, perform evaluations over a very limited variety of methods. All of these papers focus in measuring the robustness of methods that employ pairwise constraints must-link and cannot-link. In Pelleg and Baras (2007) and Covoes et al. (2013), these constraints are modified before being presented to KMeans-like algorithms - CVQE, LCVQE and MPCKMeans. In Pelleg and Baras (2007), graph-based methods were compared. In all of these papers, the only conclusion obtained is whether one method is more robust than other methods against different quantities of noisy constraints.

By analysing these works, we can conclude that further questions are still not answered. For example, which cluster-level constraint is more robust to noise? Which level of noisy constraints has worst impact in clustering results - instance-level or cluster-level constraints? Are distance-based methods more robust to noisy constraints than constraint-based methods? Does the dataset characteristics have influence in the methods robustness? These, among other unanswered questions, are important issues on robustness to noise to be considered when choosing the semi-supervised method to use and may be point of research in the future.

3.7.7 Data representation and visualization for user interaction

In order to pose constraints correctly, it is convenient that the user has a complete vision of the dataset. The less information is provided to the user about the instances or clusters to be involved in the constraints to be posed, the more tacit knowledge the user have to have about the problem domain.

In some cases, having an intuitive representation is quite trivial. For example, in document clustering or image clustering, posing instance-level constraints is easy if the user has access to the documents or images involved. Posing cluster-level constraints, however, is harder. Since accessing all the documents or images present in each cluster may demand a huge effort, the user needs to observe representative summaries of the elements of each cluster. The problem is worse when the number of constraints to be added is significant. In this case, even posing instance-level constraints demands a great number of instances analysis and, as a consequence, a great effort from the user. When dealing with other kind of data, the problem is even more complicated. For example, when dealing with sets of multidimensional numerical data, having an intuitive and correct representation of the data is quite complicated.

So, using appropriate data representation is essential to achieve efficient user interaction. However, to the best of our knowledge, there is no extensive comparison of these and other possible representations in semi-supervised clustering applications. There are questions like whether the usage of some data representation implies losing much information about the data (and consequently damaging the interaction with the clustering

process) or the cognitive effort demanded from the user in each representation. As the efficiency of the user interaction is directly related with the efficacy of the semi-supervised clustering algorithms, we consider this as one of the main questions to be investigated in semi-supervised clustering.

3.8 Final remarks

Semi-supervised clustering has proven to be interesting for exploring large datasets. While traditional unsupervised clustering algorithms search for clusters in datasets guided only by an objective function, semi-supervised algorithms use a limited external information in the clustering process. When compared to unsupervised clustering algorithms, semi-supervised algorithms can achieve a significant improvement in clustering quality without increasing the computational cost of the methods in most of the cases.

Research in semi-supervised clustering is relatively new. Apart from some work in the literature in the 1990's, the great interest in this field emerged in the early 2000's, with the proposal of the pairwise constraints by Wagstaff and Cardie (2000). Despite being a relatively new problem, different approaches have been proposed in the literature. We gave an overview of the main algorithms in these approaches along this paper. In general, these algorithms presented convincing results, that stimulate the continuous research in the area.

Considering the advances achieved, semi-supervised clustering algorithms can be regarded as a promising technique for solving complicated problems in the near future. As presented in Section 3.6, semi-supervised clustering algorithms are already employed to solve a large variety of problems, from image clustering to genetic data clustering. This is good evidence that it is worthy the usage of external information during clustering processes. From these evidences, we consider that investigating the research gaps in semi-supervised clustering, as the open questions presented in Section 3.7, may increase the potential of semi-supervised clustering algorithms. As a consequence, we expect an increasing number of problems that can be solved by using these methods in the near future.

In the context of this work, semi-supervised clustering algorithms are used to construct a hierarchical cluster structure that is afterwards converted to a topic hierarchy. According to our needs, any semi-supervised clustering approach would be adopted, as long as it provides a cluster hierarchy. However, while investigating the state-of-the-art algorithms in semi-supervised hierarchical clustering, it was possible to detect some research gaps in two main aspects: (i) the selection of informative cases to elicit constraints from the user; and (ii) how to query the user, in a practical way, during the hierarchical clustering process. In order to suppress these gaps, in this work, we introduce the HCAC (Hierarchical Confidence-based Active Clustering) and HCAC-LC (Hierarchical Confidence-based

Active Clustering with Limited Constraints) algorithms. These algorithms employ innovative active clustering approaches and are described in the next chapter. The second issue also motivated us to propose the SMITH framework to extract topic hierarchies from document collections. This framework is presented in Chapter 5 of this thesis.

Chapter

4

HCAC: Hierarchical Confidence-Based Active Clustering

As discussed in the previous chapter, semi-supervised clustering has been successfully explored in the last years in a wide spectrum of applications. These algorithms can improve the clustering quality by employing external knowledge during the clustering process. Instead of finding groups guided only by an objective function, as in unsupervised clustering, semi-supervised algorithms generally incorporate external knowledge conveyed in the form of constraints. These constraints can be directly derived from the original data (using partially labelled data) or provided by an user, trying to adapt the clustering results to his/her expectations (Dasgupta and Ng, 2010).

In this work, we employ semi-supervised clustering algorithms in order to obtain topic hierarchies to organize document collections that fit the user's expectations. This application requires semi-supervised algorithms that achieve good results even when using few user interactions, as well as provide an interactive and comprehensive way of user interaction. Moreover, appropriate cases must be selected for the user interaction, in order to maximize the potential of the constraints introduced by the user.

When analysing the state-of-the-art algorithms, there was no hierarchical clustering algorithm that convincingly fitted these expectations. Given this scenario, in this chapter we present HCAC (Hierarchical Confidence-based Active Clustering) and HCAC-LC (Hierarchical Confidence-based Active Clustering with Limited Constraints) algorithms. These algorithms were proposed during our research and aim at suppressing these existing gaps. These algorithms employ innovative ideas for active learning and user querying that would be interesting in the context of hierarchical document clustering.

In this chapter, we compare the efficiency of HCAC and HCAC-LC algorithms against two other state-of-the-art hierarchical semi-supervised clustering algorithms in three different scenarios: artificial datasets, real-world numerical datasets and real-world textual datasets. The artificial datasets, as a controlled environment, help us to better understand the behavior of the clustering algorithms. The other scenarios allow us to measure the performance of the algorithms in real-world numerical and textual datasets. We also evaluate the sensitivity of both HCAC and HCAC-LC with respect to their main parameters - the number of pairs in the pool of clusters presented to the user, the minimum cluster size in the cases where the user intervenes and the distance function employed by the algorithms.

In the next section, we present the scenario that motivated the proposal of HCAC and HCAC-LC.

4.1 Motivation

The great challenge in semi-supervised clustering is to obtain sufficient and quality knowledge from a small amount of external information. As discussed in Section 3.2 of this thesis, this challenge is related to the way the user interacts with the clustering process and to which instances or clusters the user provides information. The user interaction is defined by the characteristics of the constraints - level, kind and usage. On the other hand, to solve the cases selection issue, active learning algorithms can be used to choose proper instances or clusters to add information.

Both the information addition and case selection issues have been well explored for partitional clustering. Informative overviews of these algorithms can be seen in the works of Basu et al. (2004a); Bilenko et al. (2004); Davidson and Ravi (2005); Davidson and Basu (2007); Vu et al. (2012). On the other hand, as presented in Chapter 3 of this thesis, there is relatively little research on semi-supervised hierarchical clustering. However, there are no convincing proposals for the appropriate addition of information nor for the selection of good cases to add constraints in semi-supervised hierarchical clustering processes.

A summary of the semi-supervised hierarchical clustering algorithms is reported in Table 4.1. It is possible to see that only three of these works exploit cluster-level characteristics in hierarchical clustering. Despite cluster-level constraints can carry more information than instance-level constraints, the former are harder to present to the user than the latter. In general, cluster-level queries are harder to understand and comprehend than instance-level constraints. However, the usage of proper cluster representations can smooth this difficulty.

We can also observe that there are few active learning approaches in hierarchical semi-supervised clustering. Active learning algorithms would boost the performance of clustering algorithms by selecting informative cases to add constraints. Among these few

Table 4.1: Summary of related work on semi-supervised hierarchical clustering.

Algorithm	**Constraint Level**	**Clustering Algorithm**	**Active Learning**	**Evaluation Measure**
Talavera and Béjar (1999)	Instance	ISAAC (Conceptual)	No	Accuracy
Klein et al. (2002)	Cluster	Complete-link	Yes	Corrected Rand Index
Kestler et al. (2006)	Instance	Divisive	No	Rand Index
Daniels and Giraud-Carrier (2006)	Both	Complete-link	Yes	F-measure
Bade et al. (2007)	Instance	Agglomerative	No	F-measure
Böhm and Plant (2008)	Instance	Agglomerative	No	Mutual Information
Davidson and Ravi (2009)	Both	Complete-link	No	Error rate
Vu et al. (2010)	Instance	Agglomerative	Yes	Rand Index
Miyamoto and Terami (2011)	Instance	Ward	No	Rand Index
Zheng and Li (2011)	Instance	Agglomerative	No	F-measure

algorithms, just Klein et al. (2002) focused in finding proper cases to ask the user using cluster-level constraints. This algorithm, however, has the limitation of being based on the Complete-link algorithm. Some researches have pointed out that the Complete-Link algorithm performs worse than other agglomerative clustering algorithms, as in document clustering tasks (Zhao et al., 2005).

One other important aspect is that most of these work consider binary clustering problems only. Thus, these studies do not assess the behavior of the algorithms in multi-cluster domains, which is the case of many real-world problems.

Given this scenario, in this chapter we introduce **HCAC** (Hierarchical Confidence-based Active Clustering) and **HCAC-LC** algorithm (Hierarchical Confidence-based Active Clustering with Limited Constraints). HCAC and HCAC-LC are semi-supervised clustering algorithms designed for better exploiting external knowledge during the agglomerative hierarchical clustering process. These algorithms allow the user to pose cluster-level constraints to indicate the best decisions in cluster merging along the clustering process. The user, when requested, chooses the next pair of clusters to be merged among a pool of pre-selected pairs. In order to optimize the queries to the user, the HCAC-based algorithms employ an active learning approach to detect indecisions in cluster merges and ask for the user intervention. In this active approach, the quality of a cluster merge is measured through the **confidence** measure. This measure reflects the quality of the unsupervised merging decisions.

To the best of our knowledge, both features which are the basis of the HCAC-based algorithms - confidence and query type - were not exploited in hierarchical clustering before these algorithms. Both these features fit our problem of extracting semi-supervised topic hierarchies by better selecting the points of user interaction and providing a simple and effective way of introducing the constraints during the clustering process. These features, as well as the HCAC-based algorithms are explained in detail in the next section.

4.2 HCAC and HCAC-LC: Confidence-based Active Clustering

HCAC (Hierarchical Confidence-Based Active Clustering - pronounced h-cac) is a semi-supervised clustering algorithm based on agglomerative hierarchical clustering that was originally introduced by us in Nogueira et al. (2012b). HCAC allows the user to impose cluster-level constraints along the iterations of the agglomerative hierarchical clustering algorithm. These constraints are posed in points where there is doubt if the unsupervised cluster merge - the one that merges the nearest pair of clusters - is the best option. In order to measure the quality of the unsupervised merge, the algorithm uses the confidence measure. Basically, the idea is to consider at each cluster merge step the different clustering possibilities. This is done by measuring the confidence of the merging decisions with respect to the alternatives. When the merging confidence is sufficiently low, the algorithm queries the user.

We also propose an extension of HCAC, called **HCAC-LC** (Hierarchical Confidence-based Active Clustering with Limited Constraints). HCAC-LC is an alternative to HCAC that improves its performance by limiting the interaction with the user to larger clusters. This improvement is possible by modifying the selection of points for user's interaction along the clustering process. For that, HCAC-LC only requires user intervention when merging clusters with more than one element. Thus, the impact of the constraints is amplified in the HCAC-LC algorithm. This leads HCAC-LC to outperform HCAC when the number of constraints is small.

The basics of HCAC and HCAC-LC algorithms are explained in detail in the next sections. First, we will explain the motivation to create HCAC. Then, we will explain our approach to deal with these situations by adding cluster-level constraints. Finally, we will present the motivations of HCAC-LC and the modifications in relation to HCAC.

4.2.1 Confidence-based active clustering

The cluster merging decision in unsupervised agglomerative hierarchical clustering considers the distance function only. In each step, the nearest pair of elements is selected to be merged. However, sometimes the distance function may not perfectly represent different concepts. In these situations, unsupervised clustering algorithms are highly susceptible to make misclusterings by clustering objects that represent different concepts.

(a) Low confidence merge.

(b) High confidence merge.

Figure 4.1: Confidence of cluster merges in cluster borders.

The situation mentioned above occurs, for example, in regions of cluster borders, as in Figure 4.1(a). In this figure, we have two underlying clusters (dashed circles), corresponding to two different concepts. Let us consider a distance function $dist(\cdot,\cdot) = d(\cdot,\cdot)$ between elements in a dataset. Considering Figure 4.1(a), elements a and b (in the rectangle) belong to different concepts. Since they are nearest (i.e., $d_{a,b} = min\, dist(x,y)$, $x \neq y$) they would be the first to be merged by an unsupervised approach. However, there are better options close to the elements a and b that would generate better clustering results, since they belong to the same concept. For example, c belongs to the same concept as b and $|d_{b,c} - d_{a,b}| = \delta$, where δ is a very small value.

Motivated by this kind of situation, we proposed the concept of **confidence** of a merge (Nogueira et al., 2012b). The confidence of a merge is related to the distance between the elements from the proposed merge and other elements near them. If a pair of elements are close to each other but far from other elements (as in Figure 4.1(b)), the confidence of merging these two elements is high since apparently there is no good alternative. However, if they are also close to other elements (Figure 4.1(a)), it might be advisable to ask the user to check if there is a better merge.

Formally, a confidence value can be calculated as follows. The natural merge (unsupervised merge) in a given step of the agglomerative hierarchical clustering process involves the nearest pair of elements a and b. The confidence C of this merge is calculated by the difference between $d_{a,b}$ and $d_{e,f}$, where $d_{e,f} = min\, dist(x,y), x \neq y, (x,y) \neq (a,b), x \in \{a,b\} \oplus y \in \{a,b\}$.

Along the clustering process, cluster merges that present low confidence values are taken as points where the algorithm is more likely to make incorrect decisions (misclusterings). So, HCAC detects low confidence merges and queries the human user to check if there is a better alternative.

In practical terms, low confidence merges are those where confidence is below a predefined threshold. The higher the threshold value, the more cluster merge decisions are considered as an uncertain decision and so the more user interventions are required.

We propose a calibration procedure to estimate the confidence threshold value with respect to the amount of tolerated interaction. This calibration is performed through an unsupervised execution of the hierarchical clustering algorithm. This procedure is described in Algorithm 2. At each step of the unsupervised execution, the confidence value is calculated by measuring the difference between the two shortest distances in the distance matrix in that step. At the end of the unsupervised clustering process, all calculated confidences are ordered and an adequate threshold value is selected according to the desired number of human interactions.

With a calibrated threshold, we have a criterion for deciding when to require the intervention of the user. In the next section we describe how users interact with HCAC in order to guide the clustering process.

Algorithm 2: Threshold calibration procedure of the HCAC algorithm

Input: n: number of elements in the dataset D; $dist(.,.)$: distance function; q: desired number of human interactions
Output: $confT$: confidence threshold value

Initialize vector C holding confidence values with $n-1$ positions;
for $k = 1 : n-1$ **do**
  $minDist_k = d_{i,j} = min\, dist(x,y), x \neq y$;
  $secMinDist_k = d_{r,s} = min\, dist(x,y), x \neq y, (x,y) \neq (i,j), x \in \{i,j\} \oplus y \in \{i,j\}$;
  $C_k = secMinDist_k - minDist_k$;
end
Sort elements in vector C;
$confT = C[q]$;

4.2.2 User interaction through cluster-level constraints

When a low confidence merge involving clusters is spotted, the user is queried for additional information. The response comes in the form of a constraint. These are cluster level constraints since the merging of subclusters instead of single instances. In our proposal, we use cluster-level constraints. According to the principles of Information Theory (Shannon, 1948, 2001), as cluster-level constraints impact in a larger number of instances, they can convey more information than instance-level ones. This can reduce the number of the user's interventions required to achieve a quasi-optimal clustering solution. Instance level queries, however, can be more easily resolved by humans.

In HCAC, a cluster-level query is posed to acquire a cluster-level constraint when a low confidence merge is detected. For that, a pool of pairs of clusters is presented to the user in order to choose the pair that corresponds to the best merge. The pool contains c nearest pairs of clusters, where c is given a priori. The generation of this pool is described in Algorithm 3. It starts by finding the best unsupervised merge (the two nearest clusters i, j). After that, the $c-1$ best unsupervised merges involving i or j are included. This assembling procedure has a linear-time cost as a function of the number of elements ($O(n)$, where n is the number of elements).

Algorithm 3: Procedure for assembling the pool of cluster pairs in the HCAC algorithm

Input: n: number of elements in the dataset D; $dist(.,.)$: distance function; c: size of the pool of clusters
Output: P_k: pool of pairs of clusters on the k-th iteration

Initialize vector P holding the candidate pairs with c positions;
$P[1] = (i,j) = \underset{x,y \in D}{\arg\min}\, dist(x,y), x \neq y$;
for $l = 2 : c$ **do**
  $P[l] = (r,s) | (r,s) \notin P, dist(r,s) = min\, dist(x,y), x \neq y, (x,y) \neq (i,j), x \in \{i,j\} \oplus y \in \{i,j\}$;
end

The higher the value of c, the more options the user has, and the brighter the chances are of finding a good choice. However, a large number of cluster pairs may imply excessive

human effort. Moreover, dealing with a pool of clusters may not be trivial. The direct visualization and interpretation of clusters is generally a difficult task. This drawback could be smoothed using good summarizing cluster representations, such as wordclouds or document summarization (Cai and Li, 2011) for textual datasets and parallel coordinates (Inselberg, 2009) for non-textual datasets.

The adoption of the active confidence-based approach tries to optimize the user's intervention. Moreover, the cluster-level constraints and the new kind of queries tend to generate clusters with high purity degrees, as they help to better determine the cluster boundaries. This fact makes HCAC useful when dealing with datasets with a large number of clusters.

Following this procedure, the computational cost of HCAC is dominated by the hierarchical clustering process. As HCAC is based on the agglomerative clustering approach, this cost is $O(n^2 * log(n))$, where n is the number of elements in the dataset. The process of assembling the pool of pairs, presented in Algorithm 3, has linear cost with respect to the number of clusters in the clustering step. Likewise, the process of threshold calibration is done through an unsupervised clustering process, also with cost $O(n^2 * log(n))$.

4.2.3 HCAC-LC: Improving the performance of HCAC with few constraints by solving the singletons problem

According to results of an initial evaluation of HCAC, reported by Nogueira et al. (2012b), HCAC presented unstable results in scenarios where the user provides a small number of constraints. An example of these results can be seen in Table 4.2. In this table, the results of the statistical comparison of the clustering quality is presented, where HCAC is compared with another semi-supervised hierarchical clustering algorithm based in pairwise constraints (Davidson and Ravi, 2009) and an unsupervised clustering algorithm (Average-Link).

Table 4.2: Results of the statistical comparisons of an initial evaluation of HCAC.

	10 Pairs				20 Pairs			
%	Pairwise		Average		Pairwise		Average	
1	△	13 - 5	▽	5 - 6	△	13 - 5	△	7 - 4
5	▽	8 - 14	▽	7 - 10	△	13 - 9	△	12 - 7
10	△	12 - 10	△	11 - 8	△	12 - 10	△	14 - 6
20	▽	10 - 12	▽	7 - 12	△	14 - 8	△	13 - 7
30	-	11 - 11	▲	14 - 8	△	15 - 7	▲	18 - 4
40	△	12 - 10	▲	16 - 6	△	12 - 10	▲	16 - 6
50	-	11 - 11	▲	18 - 4	△	13 - 9	▲	18 - 4
60	△	12 - 10	▲	19 - 3	▲	15 - 7	▲	20 - 2
70	△	13 - 9	▲	21 - 1	▲	17 - 5	▲	22 - 0
80	▲	14 - 8	▲	21 - 1	▲	17 - 5	▲	22 - 0
90	▲	18 - 4	▲	22 - 0	▲	20 - 2	▲	21 - 1
100	▲	22 - 0	▲	22 - 0	▲	22 - 0	▲	22 - 0

These initial experiments considered 22 numerical datasets from the UCI repository[1].

[1]http://archive.ics.uci.edu/ml/datasets.html

We have varied the number of desired interventions in 1%, 5%, 10%, 20% ... 100% of the number of merges in the agglomerative clustering. In the HCAC, we have also tested two different number of pair of elements in the pool: 10 and 20. The experiments were carried out using 10-fold cross validation and evaluated through the FScore measure (Larsen and Aone, 1999). The Wilcoxon statistical test was used to compare HCAC against the other algorithms using an α of 0.05. In the table, the symbol ▲indicates that HCAC obtains better results with statistical significance; △indicates that HCAC obtains better results with no statistical significance; ▽indicates that HCAC obtains worse results with no statistical significance. Each symbol is followed by the number of datasets that HCAC performs better and worse than the compared algorithm.

As can be observed, in intervals with less than 50% of interventions, HCAC was not able to outperform the compared algorithms in most of the comparisons. Scenarios where the user has little interaction with the process have a great practical importance. So, it is worthwhile to search for a solution that improves the performance of HCAC in these cases.

An analysis of the queries posed by HCAC indicated that one of the causes of this limitation is the insertion of constraints to pairs of singletons. According to this analysis, a large number of queries in HCAC were posed when low-confidence merges involving two singletons were detected. In these situations, most part of the options for the user in the pool of candidate pairs were also pairs of singletons. So, the constraints provided by the user were, in great part, about single examples.

Adding constraints for pairs of clusters which are singletons limits the HCAC ability to reduce the uncertainty (Garner, 1962) in clustering decisions. To analyse this statement, let us consider two clusters C_a and C_b with $|C_a|$ and $|C_b|$ elements, respectively. Also, let us consider a clustering decision D indicating whether C_a and C_b must or must not be clustered in that step. In this scenario, D implies clustering each of the $|C_a| * |C_b|$ pairs of elements in a same or in different clusters, with $2^{(|C_a|*|C_b|)}$ possible instance-level outcomes. The uncertainty of this decision can be calculated using Equation 4.1.

$$U_D = log_2(|C_a| * |C_b|) \tag{4.1}$$

Following this reasoning, we can see that the interactions that introduce the least amount of information are the ones using constraints between singleton clusters. In these situations, HCAC is not able to fully explore the advantages of posing cluster-level queries.

To illustrate the magnitude of this limitation in the context of HCAC, in Figure 4.2 we present an analysis of the queries posed in an execution of the HCAC algorithm. This analysis was carried out considering two well known datasets from the UCI repository, Iris and Pima. In this figure, the Y-axis represents the percentage of queries posed when low confidence merges of singletons were detected. These percentages were calculated in

12 different scenarios, which are presented on the X-axis. Each of these scenarios use different number of constraints. The number of constraints are calculated according to percentages of the total merges in an agglomerative clustering process, varying from 1% to 100% of the total cluster merges.

According to the results in Figure 4.2, a high percentage (in average, 20% to 30%) of the constraints were posed considering low-confidence merges of a pair of singletons. The main drawback in considering low-confidence merges of pairs of singletons is that most of them are also surrounded by singletons. Thus, must of the options in the pool of candidate pairs are composed by pairs of singletons and the constraints would cover a small number of elements. For example, considering 5 pairs in the pool, 73% of the pairs in the pool were composed by two singletons in the Iris dataset and 69% in the Pima dataset. Thus, the exploration of these regions is not interesting in terms of exploited information.

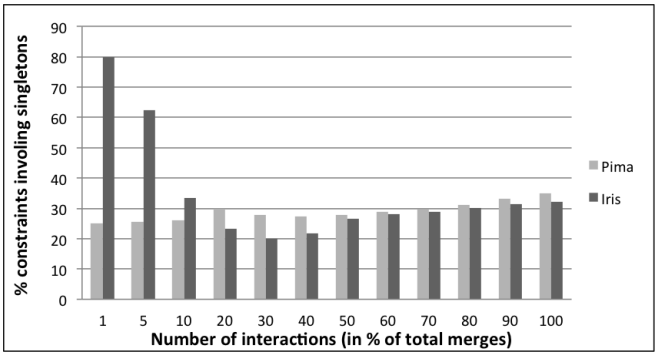

Figure 4.2: Analysis of low-confidence cluster merges involving singletons.

As expressed in Equation 4.1, we can insert more information to the clustering process by either increasing the amount of constraints or increasing the amount of elements in each cluster influenced by one constraint. Adding more constraints, however, is not desirable in real-world applications, as users are not likely to respond to many queries.

In order to amplify the information addition in a user-friendly way, we introduce HCAC-LC which only considers non-singleton clusters in the constraint posing process. The restriction affects both confidence threshold calibration and posing queries to the user. Considering a cluster merge involving two clusters x and y, HCAC-LC only allows the imposition of constraints if $|x| \geqslant 2$ and $|y| \geqslant 2$, where $|.|$ is the number of elements in a given cluster. These modifications would be added in lines 3 and 4 of Algorithm 2 and lines 2 and 4 of Algorithm 3.

These principles lead HCAC-LC to improve its performance in scenarios with less intervention from the user (limited constraints) when compared to HCAC. On the other hand, HCAC-LC performs worse than HCAC in scenarios with more user intervention (more constraints are provided by the user). This is due to the fact that the principles introduced in HCAC-LC limit the maximum constraints that can be posed in this algo-

rithm. For instance, let us consider a dataset with n elements and, consequently, $n-1$ cluster merges. In this situation, at least $s = \lceil \frac{n}{2} \rceil$ cluster merges would involve singletons. According to this formulation, HCAC-LC is allowed to pose at most $n-s-1$ constraints.

So, in higher percentages of interaction, HCAC-LC may not be allowed to pose the number of required constraints if it exceeds $n-s-1$. However, in real-world applications, this drawback does not have a big impact, as posing a big number of constraints is not interesting. These behaviors can be seen in the results of the experimental evaluation, presented in the next section.

4.3 Experimental evaluation

To evaluate HCAC and HCAC-LC, we have carried out three sets of experiments. The first one used 22 artificially generated bi-dimensional datasets[2], varying the number of clusters in each dataset from 2 to 100. All datasets are perfectly balanced, with 30 examples in each cluster. Each cluster is formed by the combination of two normal distributions (one for the x-axis and other for the y-axis), separated by a constant distance and, therefore, are well shaped. The main objective of this experiment with this controlled environment is to see how HCAC and HCAC-LC performance varies according to three parameters: (i) the minimum number of elements in the clusters in the cases where the user intervention is required, comparing HCAC and HCAC-LC algorithms; (ii) the number of clusters in the dataset; and (iii) the number of pairs in the pool.

In the second set of experiments, we have assessed the performance of HCAC-LC in 31 real-world numerical datasets from the UCI repository and from the MULAN repository[3]. These datasets are approximately balanced and have labelled instances which enables the objective evaluation of clustering results. For these datasets, the Euclidean distance function was used on the experiments, since it is the most adequate distance function for numerical datasets. A brief description of these datasets are presented in Table 4.3.

Finally, our third set of experiments used 19 real-world textual datasets, from the Cluto Project[4] and from one other collection assembled by our research group[5]. These datasets are described in the Table 4.4. They also are approximately balanced and have labelled instances. For these datasets, the cosine distance function was used on the experiments. This analysis has a special practical objective for this work, as we look for improving the organization of textual collections in topic hierarchies through semi-supervised hierarchical clustering.

The evaluation methodology applied on all sets of experiments using these datasets and the obtained results are presented in the following sections.

[2]Datasets available at http://sites.labic.icmc.usp.br/bmnogueira/artificial.html
[3]http://mulan.sourceforge.net/datasets.html
[4]http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download
[5]http://sites.labic.icmc.usp.br/ragero/arffs

Table 4.3: Description of the real-world numerical datasets used in the experiments. MULAN datasets are highlighted with the symbol '*'.

Dataset	# Examples	# Classes	Dataset	# Examples	# Classes
Balance	625	3	MFeat	2000	10
Breast Cancer Wisconsin	683	2	Musk	476	2
Breast Tissue	106	6	Pima	768	2
Cardiotocography	2126	10	Scene*	2417	15
Ecoli	336	8	Secom	1151	2
Emotions*	593	27	Sonar	208	2
Glass	214	6	Soybean	266	15
Haberman	306	2	Spectf	267	2
Image Segmentation	210	7	Statlog Satellite	4435	7
Ionosphere	351	2	Transfusion	748	2
Iris	150	3	Vehicle	846	4
Isolet	1559	26	Vertebral Column	310	3
Libras	360	15	Vowel	990	10
Lung Cancer	27	3	Wine	178	3
Madelon	600	2	Zoo	101	7
Mammographic Masses	830	2			

Table 4.4: Description of the real-world textual datasets used in the experiments. CLUTO datasets are highlighted with the symbol '*'.

Dataset	# Examples	# Classes	Dataset	# Examples	# Classes
ACM	3493	5	RE0*	1504	13
Classic 3	7095	4	RE1*	1657	25
CSTR	299	4	Review Polarity	2000	2
FBIS*	2463	17	Reviews	4069	5
Hitech*	2301	6	Syskill Webert	334	4
Irish	1660	3	TR23*	204	6
K1A*	2340	20	TR31*	927	7
K1B*	2340	6	TR41*	878	10
LA1*	3204	6	WAP*	1560	20
LA2*	3075	6			

4.3.1 Evaluation methodology

We have compared HCAC and HCAC-LC against three standards: an unsupervised algorithm (average-link), which is used as a baseline; a semi-supervised algorithm using must-link and cannot-link pairwise constraints (Wagstaff and Cardie, 2000); and the active constrained hierarchical clustering process proposed in Klein et al. (2002) (Constrained Complete Link - CCL), which also uses cluster level constraints along with the clustering process. The CCL algorithm uses a complete-link strategy to perform the cannot-links propagation while the other two approaches use the average-link strategy (Jain and Dubes, 1988). The comparison with the baseline unsupervised algorithm is done for assessing the ability of the semi-supervised algorithms to exploit the information provided by the user.

We simulated the human interaction in the semi-supervised algorithms by using the labels provided with the datasets. The idea is to automatically answer the queries using a sensible criteria that models the user's behaviour. In HCAC and HCAC-LC, for the cluster-level queries, the criteria to choose the best cluster merge was entropy (Shannon, 1948). Among the pairs in the pool, the one with the lowest entropy value is selected for merging. For the algorithm using pairwise constraints, we randomly picked pairs of instances before the clustering process starts. As suggested by Davidson and Ravi (2009), if the elements belong to the same class, then a must-link constraint was added and the

distance between this pair was set to zero. Otherwise, a cannot-link constraint was added and the distance was set to infinity. Finally, for the CCL algorithm, it was established that the roots of the next proposed merge have to be merged if they present an entropy equals or lower than 0.2. This value was achieved through preliminary empirical tests.

We have tried different numbers of human interventions in the clustering process (number of pairwise queries or cluster-level queries). We have varied the number of desired interventions in 1%, 5%, 10%, 20%, 30%, ..., 100% of the number of merges in the agglomerative clustering process (which is equal to the number of instances in the dataset minus one). In the case of the HCAC and HCAC-LC algorithms, we have also tested two different numbers of pairs of elements in the pool: 5 and 10. In a real application, the usage of 10 pairs in the pool may not be a viable configuration, since it would demand too much effort from the user. However, we decided to compare this configuration in order to analyse how the size of the pool impacts the performance of HCAC and HCAC-LC.

In the evaluation, we have used 10-fold cross validation. Each resulting clustering was evaluated through the FScore measure (Larsen and Aone, 1999; Aliguliyev, 2009) which is very adequate for hierarchical clustering. The FScore for a class K_i is the maximum value of FScore obtained at any cluster C_j of the hierarchy, which can be calculated according to Equation 4.2:

$$F(K_i, C_j) = \frac{2 * R(K_i, C_j) * P(K_i, C_j)}{R(K_i, C_j) + P(K_i, C_j)}. \tag{4.2}$$

where $R(K_i, C_j)$ is the recall for the class K_i in the cluster C_j, defined as n_{ij} / size of K_i (n_{ij} is the number of elements in C_j that belongs to K_i) and $P(K_i, C_j)$ is the precision, defined as n_{ij} / size of C_j. The FScore value for a clustering is calculated by the weighted average of the FScore for each class, as shown on Equation 4.3.

$$FScore = \sum_{i=1}^{k} \frac{n_i}{n} F(K_i) \tag{4.3}$$

The final FScore value for a given dataset is the average of the FScore values for each of the 10 folds. The non-parametric Wilcoxon (Wilcoxon, 1945) statistical test was used to detect statistical significance in the differences of the algorithms performance considering an α of 0.05. The test was applied to compare HCAC and HCAC-LC algorithms against one of the other algorithms.

4.3.2 Results and discussion

In the next subsections, we present and discuss the results of each of the three sets of experiments. In the first one, with artificial datasets, we assess and compare the performance of HCAC and HCAC-LC in a controlled environment. In the second and third sets, using real-world datasets, we evaluate the performance of HCAC and HCAC-LC in numerical and textual datasets. These experiments with real-world datasets aim at

measuring HCAC and HCAC-LC performance in real-world problems and their sensitivity to different distance functions, analysing both the euclidean and the cosine distance function.

In the tables presented in this section, we have the results of the statistical comparison of HCAC or HCAC-LC against other algorithms. In these tables, as previously explained in this chapter, the symbol ▲indicates that our algorithm (HCAC or HCAC-LC) obtains better results with statistical significance; △indicates that our algorithm obtains better results with no statistical significance; ▽indicates that our algorithm obtains worse results with no statistical significance; and ▼indicates that our algorithm obtains worse results with statistical significance. Each symbol is followed by the number of datasets that HCAC or HCAC-LC performs better and worse than the compared algorithm.

Artificial Datasets

The first set of experiments used 22 artificial datasets. The main objective of these experiments is to measure the performance of HCAC and HCAC-LC in a controlled environment according to three different parameters: (i) the minimum number of elements in the clusters in the cases where the user intervention is required; (ii) the number of clusters on the dataset; and (iii) the number of pairs in the pool.

The statistical analysis of these results is shown in Tables 4.5 and 4.6. Also, we present a direct comparison of HCAC and HCAC-LC in Table 4.7. By looking at these results, it can be easily noticed that both HCAC and HCAC-LC statistically outperform all other compared algorithms in most of the configurations.

Table 4.5: Results of the statistical comparisons of HCAC against other algorithms in the artificial datasets.

	5 Pairs			10 Pairs		
%	Pairwise	CCL	Average	Pairwise	CCL	Average
1	△13 - 4	▲22 - 0	△12 - 5	▲14 - 4	▲22 - 0	▲12 - 6
5	▲19 - 3	▲22 - 0	▲16 - 5	▲19 - 3	▲22 - 0	▲19 - 3
10	▲20 - 2	▲22 - 0	▲21 - 1	▲21 - 1	▲22 - 0	▲19 - 2
20	▲21 - 1	▲22 - 0	▲21 - 1	▲22 - 0	▲21 - 1	▲19 - 3
30	▲21 - 1	▲20 - 2	▲22 - 0	▲22 - 0	▲21 - 1	▲21 - 0
40	▲22 - 0	▲20 - 2	▲22 - 0	▲22 - 0	▲21 - 1	▲22 - 0
50	▲21 - 1	▲20 - 2	▲22 - 0	▲21 - 1	▲21 - 1	▲22 - 0
60	▲20 - 2	▲18 - 4	▲21 - 0	▲21 - 1	▲21 - 1	▲22 - 0
70	▲22 - 0	▲19 - 3	▲22 - 0	▲22 - 0	▲20 - 2	▲21 - 1
80	▲21 - 1	▲18 - 4	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0
90	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0
100	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0	▲22 - 0

The first analysis carried out in this set of experiments is about the minimum number of elements in the clusters where the user interacts. This was done by comparing the clustering results obtained applying HCAC and HCAC-LC. A summary of these results is presented in Table 4.7. It is possible to see that the principles of HCAC-LC, proposed in this work, have a positive impact when the number of constraints is small (between 1% and 20% of interaction). As discussed in Section 4.2.3, HCAC-LC only allows the imposition

Table 4.6: Results of the statistical comparisons of HCAC-LC against other algorithms in the artificial datasets.

	5 Pairs			10 Pairs		
%	Pairwise	CCL	Average	Pairwise	CCL	Average
1	▲15 - 3	▲22 - 0	▲12 - 5	▲16 - 3	▲22 - 0	▲14 - 4
5	▲20 - 1	▲22 - 0	▲19 - 2	▲20 - 1	▲22 - 0	▲20 - 1
10	▲21 - 1	▲22 - 0	▲20 - 2	▲21 - 1	▲22 - 0	▲21 - 1
20	▲20 - 2	▲22 - 0	▲21 - 1	▲20 - 2	▲22 - 0	▲22 - 0
30	▲20 - 2	▲20 - 2	▲20 - 2	▲20 - 2	▲20 - 2	▲21 - 1
40	▲21 - 1	▲20 - 2	▲20 - 2	▲22 - 0	▲20 - 2	▲21 - 1
50	▲20 - 2	▲16 - 6	▲21 - 1	▲20 - 2	▲19 - 3	▲21 - 0
60	▲20 - 2	▽8 - 14	▲21 - 1	▲20 - 2	▷11 - 11	▲22 - 0
70	▲21 - 1	▼5 - 17	▲21 - 1	▲21 - 1	▼5 - 17	▲21 - 1
80	▲20 - 2	▼4 - 18	▲21 - 1	▲20 - 2	▼4 - 18	▲21 - 1
90	▲20 - 2	▼4 - 18	▲20 - 2	▲20 - 2	▼3 - 18	▲21 - 1
100	▲19 - 3	▼1 - 21	▲20 - 2	▲19 - 3	▼1 - 21	▲20 - 2

Table 4.7: Statistical comparison of the two HCAC approaches (HCAC-LC vs. HCAC).

	HCAC	
%	5 Pairs	10 Pairs
1	△14 - 8	△12 - 9
5	△18 - 4	△14 - 8
10	▲17 - 4	▲16 - 5
20	▲14 - 8	△12 - 10
30	▽10 - 12	▽8 - 14
40	▽8 - 14	▼2 - 20
50	▼1 - 21	▼0 - 22
60	▼0 - 22	▼0 - 22
70	▼0 - 22	▼1 - 21
80	▼1 - 21	▼0 - 22
90	▼0 - 22	▼0 - 22
100	▼0 - 22	▼0 - 22

of constraints in cases that do not involve singletons. This principle lead the constraints in HCAC-LC to cover a consistently larger number of examples. As a consequence, the performance of HCAC-LC is enhanced with a smaller number of interactions.

On the other hand, not considering low confidence merges involving singletons limits the performance of HCAC-LC when the number of constraints is higher (especially between 50% and 100% of interaction). As explained in Section 4.2.3, by not considering singletons HCAC-LC is allowed to pose at maximum $n - \lceil \frac{n}{2} \rceil - 1$ constraints (approximately 50% of the number of cluster merges). This drawback, however, is generally not important in real applications such as in the construction of topic hierarchies, since a large number of user's interventions is not of practical interest in most of the applications.

To analyse the two other parameters - the number of pairs in the pool and the number of clusters in the dataset -, we present an FScore analysis of both HCAC and HCAC-LC in some of the artificial datasets in Figure 4.3. In both analysis, we used 5 and 10 pairs in the pool for HCAC and HCAC-LC.

Concerning the number of clusters in the dataset, it can be observed that the performance of the algorithms decay as the number of clusters increases. It can be also noticed that this decay is stronger for the algorithms that use none or instance-level constraints (pairwise and average). So, algorithms that employ cluster-level constraints (HCAC and

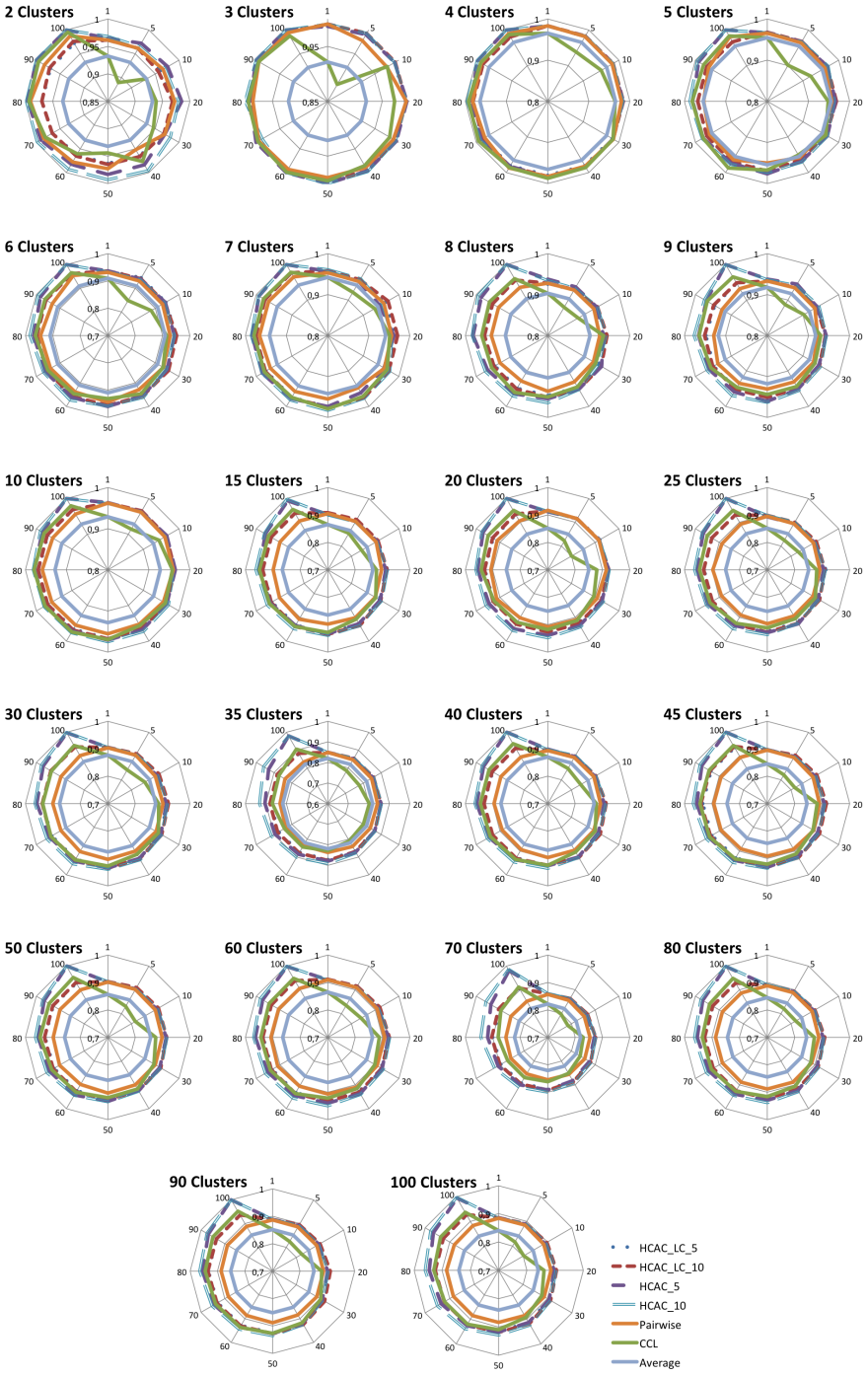

Figure 4.3: Results obtained by HCAC and HCAC-LC in the artificial datasets. In the perimeter we have the level of human intervention (in %). In the radius we have the F-score.

CCL) tend to perform much better than the other algorithms, especially when the number of clusters is high. Particularly, HCAC and HCAC-LC tend to outperform all other algorithms when the number of clusters in the dataset is greater than three.

This tendency can be explained by the nature of the constraints. In general, the more clusters a dataset has, the more complex it is and the more information will be needed to correctly delimit them. With the pairwise constraints, the user indicates whether two instances do or do not belong to the same cluster. On the other hand, our proposed cluster-level constraints indicate that two groups of instances must be merged. So, in the cluster-level constraint the number of instances influenced and the quantity of information added are higher. Moreover, our active learning approach tends to require the user's intervention on points that can be regarded as cluster borders. The more clusters a dataset has, the more border regions are present and the higher the chances are of misclusterings.

In order to highlight this behaviour variation as the number of clusters increases, we present in Figure 4.4 a comparison of HCAC and HCAC-LC with pairwise-constrained and CCL approaches according to the number of clusters in the dataset. In the horizontal axis we have the number of clusters. For each number of clusters, we calculated the victory rate of the HCAC and HCAC-LC algorithms over the compared algorithm. The victory rate is the proportion of the cases where HCAC or HCAC-LC present higher FScore than the other algorithm with respect to the total number of comparisons. Each victory rate was calculated considering all datasets with the same number of clusters.

In this victory rate analysis, we considered the results of cases with constraints addition rate between 1% and 50% of the total cluster merges. We delimited this interval of constraints addition to guarantee a fair comparison, since in higher percentages of intervention the number of constraints added by HCAC-LC may not be the same as in other semi-supervised algorithms. Two different victory rate lines were plotted, one for each experiment configuration (5 and 10 pairs in the pool). According to the results, both HCAC and HCAC-LC algorithms tend to have more advantage over Pairwise and CCL (rate above 0.5) in datasets with a large number of clusters.

The last analysis carried out in this set of experiments was focused on the influence of the size of the pool of clusters in the performance of both HCAC and HCAC-LC. This analysis was based on the results reported in Tables 4.6 and 4.5, as well in Figures 4.3 and 4.4. In all comparisons in this controlled environment, we can see that there is a non-significant improvement on the performance of HCAC and HCAC-LC when more pairs of clusters are presented to the user. This improvement of performance is expected, since with more pairs in the pool HCAC and HCAC-LC are able to exploit extra information. This improvement, however, was not as large as expected, showing that both HCAC and HCAC-LC are not as sensible to this parameter as to the other two parameters. Moreover, increasing the number of pairs in the pool implies an extra cognitive cost to the user, since he/she has to analyse more options for cluster merging.

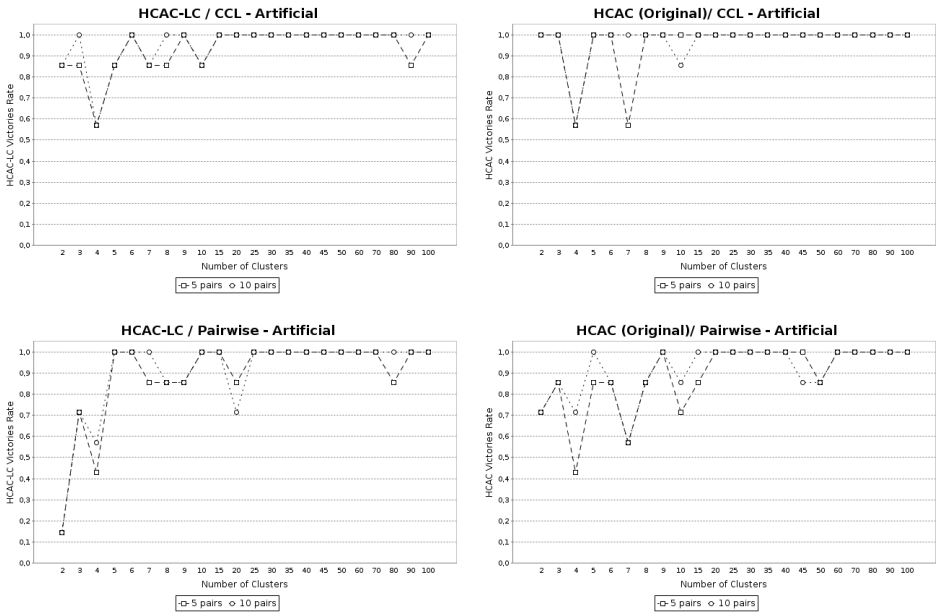

Figure 4.4: Comparison of the performance of HCAC and HCAC-LC against other semi-supervised approaches on the artificial datasets. On the X axis we have the number of clusters in the dataset. On the Y axis, we have the HCAC or HCAC-LC victory rate.

Numerical datasets

The first set of experiments using real-world datasets considered numerical datasets, described in Table 4.3, and the euclidean distance function. To highlight the behavior of clustering algorithms in contexts with different number of clusters, this set of experiments was divided in two groups of analysis. The first group of analysis considered all 31 numerical datasets, containing both binary (with 2 clusters) and non-binary datasets (with more than two clusters). The second group considered the 20 numerical non-binary datasets. These two groups are explained in the sections bellow.

a) First group of analysis - binary and non-binary datasets

The results of the statistical comparisons of all of the real-world numerical dataset experiments can be observed in Tables 4.8 and 4.9 and Figure 4.5. According to these results, it can be noticed that the performance of all algorithms present a great decrease when compared to the artificial datasets. This can be explained by the fact that the clusters in real-world datasets are not as clearly shaped as the ones in the artificial datasets.

Comparing the HCAC algorithm with the unsupervised algorithm there is only a clear advantage after 30% of user's interventions, with 5 pairs in the pool, and 40% of user's interventions, with 10 pairs in the pool. The results in Figure 4.5 and Table 4.8 show that with less interventions the performance of HCAC and average-link are very similar and there are non significant wins and losses.

Table 4.8: Results of the statistical comparisons of HCAC on the real-world numerical datasets.

	5 Pairs			10 Pairs		
%	**Pairwise**	**CCL**	**Average**	**Pairwise**	**CCL**	**Average**
1	△15 - 10	▲23 - 8	△9 - 7	△16 - 9	▲23 - 8	△11 - 7
5	△16 - 14	▲21 - 9	▽11 - 14	▽12 - 18	▲19 - 11	△13 - 11
10	△15 - 14	▲21 - 9	▲18 - 8	▷15 - 15	▲22 - 7	△16 - 10
20	▷15 - 15	▲20 - 10	△15 - 12	▽14 - 16	△16 - 14	▷13 - 13
30	▷15 - 15	▲25 - 5	△16 - 13	▽14 - 16	△21 - 9	▲20 - 9
40	▽13 - 17	▲22 - 8	▲23 - 6	▷15 - 15	▲23 - 7	▲23 - 6
50	▽14 - 16	▲24 - 6	▲22 - 8	△17 - 13	△20 - 10	▲26 - 4
60	△19 - 11	▲23 - 7	▲26 - 4	△18 - 12	△18 - 12	▲27 - 3
70	△16 - 14	▲22 - 8	▲26 - 4	▲20 - 10	△19 - 11	▲29 - 1
80	△19 - 11	▲23 - 7	▲30 - 0	▲21 - 9	▲23 - 7	▲29 - 1
90	▲20 - 10	▲23 - 7	▲29 - 1	▲26 - 4	▲26 - 4	▲30 - 0
100	▲29 - 1	▲25 - 5	▲30 - 0	▲30 - 0	▲29 - 1	▲30 - 0

Table 4.9: Results of the statistical comparisons of HCAC-LC on the real-world numerical datasets.

	5 Pairs			10 Pairs		
%	**Pairwise**	**CCL**	**Average**	**Pairwise**	**CCL**	**Average**
1	△16 - 11	▲24 - 7	△15 - 8	△15 - 12	▲24 - 7	△16 - 7
5	△16 - 14	▲24 - 6	△19 - 10	△17 - 13	▲24 - 6	▲22 - 7
10	△17 - 13	▲24 - 6	▲23 - 7	△21 - 9	▲24 - 6	▲23 - 7
20	△16 - 14	△20 - 10	▲22 - 8	△17 - 13	▲19 - 11	▲23 - 7
30	▽14 - 16	△20 - 10	▲24 - 6	▷15 - 15	▲22 - 8	▲23 - 7
40	▽13 - 17	△19 - 11	▲21 - 9	△17 - 13	△22 - 8	▲26 - 4
50	▽13 - 17	△16 - 14	▲28 - 2	△16 - 14	△22 - 8	▲27 - 3
60	▽13 - 17	△16 - 14	▲25 - 5	△17 - 13	△21 - 9	▲26 - 4
70	▷15 - 15	▽13 - 17	▲27 - 3	▲19 - 11	△21 - 9	▲27 - 3
80	▷15 - 15	▽11 - 19	▲27 - 3	▲18 - 12	△20 - 10	▲28 - 2
90	△17 - 13	▼10 - 20	▲27 - 3	▲20 - 10	△16 - 14	▲29 - 1
100	▲20 - 10	▽13 - 17	▲29 - 1	▲21 - 9	△17 - 13	▲29 - 1

Also, the performance of HCAC is very similar to the pairwise constrained approach, alternating winnings and losses. This indicates that, in general, the quality of information added is very similar in both approaches. In the comparison with CCL, HCAC tends to have advantage in all comparisons, presenting statistically significant better performance even with just few user's interventions.

As presented in Table 4.9, HCAC-LC tends to outperform CCL, Pairwise constrained and baseline algorithms in most of the comparisons in the interval between 1% and 20% of intervention. In higher percentages of interventions, the Pairwise algorithm tends to outperform HCAC-LC with 5 pairs in the pool of candidate pairs. This is an expected behaviour, since HCAC-LC was designed to efficiently exploit a reduced number of constraints. As previously explained in Section 4.3.2, in some cases with high intervention rate, HCAC-LC poses less queries than the other semi-supervised algorithms. It is important to highlight that the original HCAC algorithm tends to outperform the Pairwise constrained algorithm in these datasets when considering more constraints, as reported in Nogueira et al. (2012a).

HCAC-LC also presented a good and stable performance in the presence of a larger number of constraints. The performance of HCAC-LC is very similar to the other semi-supervised algorithms in these scenarios and is superior to the baseline (Average) algo-

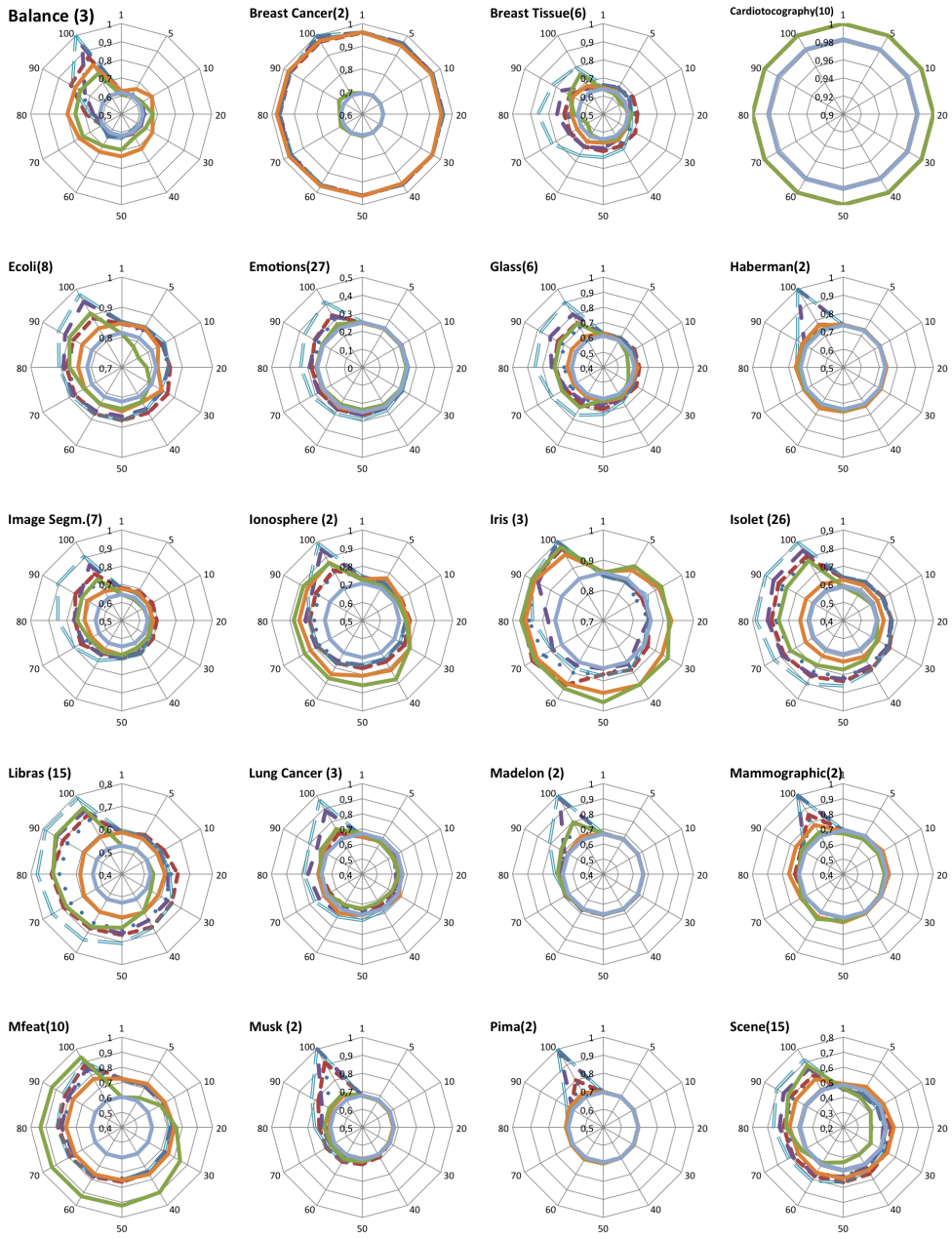

Figure 4.5: (Part I of II) Results for numerical datasets (number of clusters in parenthesis).

rithm. Considering 10 pairs in the pool leads HCAC-LC to outperform all other algorithms in most of the comparisons.

An important improvement was achieved when comparing the performance of HCAC-LC against the unsupervised algorithm (average), as shown in Table 4.9. HCAC-LC statistically outperformed the unsupervised algorithm when having at least 5% of the user's interventions, with 10 pairs in the pool, and 10% of the user's interventions, with 5 pairs in the pool. Since the original HCAC algorithm only outperformed the unsupervised algorithm in the same datasets from 30% of intervention, HCAC-LC clearly achieved better results with fewer interventions. This reinforces the efficacy of the modifications

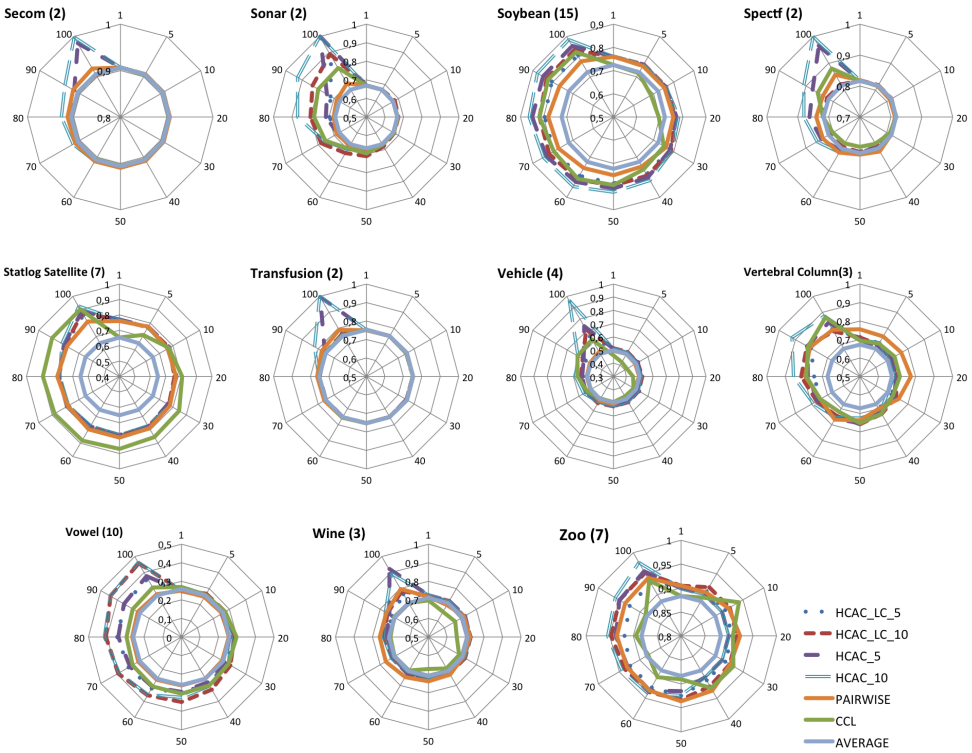

Figure 4.6: (Part II of II) Results for numerical datasets (number of clusters in parenthesis).

introduced in HCAC-LC in these scenarios.

b) Second group of analysis - non-binary datasets only

In this real-world numerical dataset experiment, it can be noticed that HCAC and HCAC-LC do not achieve a better performance than other algorithms in some comparisons, especially when compared to the Pairwise constrained algorithm. These results are highly influenced by the datasets with two and three clusters. As presented in Figure 4.7, in binary datasets (with two classes), the performance of algorithms that use none or pairwise constraints are very similar to the performance of HCAC-LC. In these datasets, as there are less cluster borders than in datasets with more clusters, it is not as hard to correctly delimit clusters boundaries as in datasets with more clusters. So, less information has to be inserted, which makes instance-level constraints efficient in this context.

To compare the performance of the algorithms in the presence of more clusters in real-world datasets, we have also carried out a statistical comparison of the performance of the algorithms in datasets with more than two clusters. This comparison used the 20 real-world datasets that contain three or more clusters. The results of this comparison can be seen in Tables 4.10 and 4.11.

According to these results, it is possible to observe that HCAC and HCAC-LC perform better than all of the other algorithms when the number of clusters in the dataset is greater

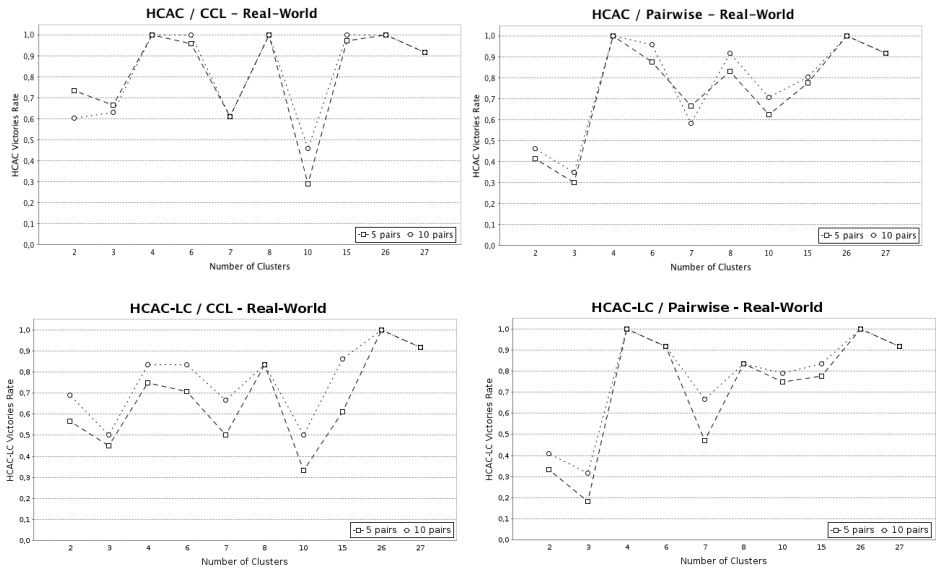

Figure 4.7: Comparison of the performance of HCAC and HCAC-LC against other semi-supervised approaches on the real-world numerical datasets. On the X axis we have the number of clusters in the dataset. On the Y axis, we have the HCAC victory rate, measured as the proportion of the cases where HCAC or HCAC-LC present higher FScore than the other algorithm.

than two. The only exception is the CCL algorithm, which presents better performance than HCAC-LC in scenarios with a very large number of constraints. As demonstrated in Figure 4.7, HCAC and HCAC-LC tend to present a winning rate above 0.5 against all of the other algorithms in almost all of the non-binary datasets. In this figure, the unexpected result for 10 cluster datasets is influenced by the results of the Cardiotocography dataset, in which all algorithms achieve the optimal solution. This behavior suggests that both the cluster-level constraints and the active learning approaches employed in HCAC and HCAC-LC are efficient in scenarios with more cluster borders.

Table 4.10: Results of statistical comparisons of HCAC on real-world datasets with more than two clusters.

	5 Pairs			**10 Pairs**		
%	**Pairwise**	**CCL**	**Average**	**Pairwise**	**CCL**	**Average**
1	△9 - 6	▲17 - 3	△7 - 4	△8 - 7	▲17 - 3	△8 - 4
5	△10 - 9	▲14 - 5	△9 - 8	△10 - 9	▲15 - 4	△11 - 6
10	△10 - 9	▲14 - 5	▲16 - 2	△10 - 9	▲15 - 4	△11 - 7
20	△12 - 7	▲12 - 7	△13 - 5	△12 - 7	△11 - 8	△12 - 6
30	△11 - 8	△14 - 5	▲12 - 6	△11 - 8	△13 - 6	▲14 - 4
40	△11 - 8	▲14 - 5	▲17 - 1	△11 - 8	△14 - 5	▲16 - 2
50	△11 - 8	▲15 - 4	▲17 - 2	△14 - 5	△15 - 4	▲18 - 1
60	△15 - 4	▲15 - 4	▲18 - 1	▲14 - 5	△14 - 5	▲18 - 1
70	▲13 - 6	▲15 - 4	▲17 - 2	▲15 - 4	△15 - 4	▲18 - 1
80	▲15 - 4	▲15 - 4	▲19 - 0	▲15 - 4	▲15 - 4	▲19 - 0
90	▲16 - 3	▲16 - 3	▲19 - 0	▲18 - 1	▲17 - 2	▲19 - 0
100	▲18 - 1	▲14 - 5	▲19 - 0	▲19 - 0	▲18 - 1	▲19 - 0

Table 4.11: Results of statistical comparisons of HCAC-LC on real-world numerical datasets with more than two clusters.

	5 Pairs			10 Pairs		
%	**Pairwise**	**CCL**	**Average**	**Pairwise**	**CCL**	**Average**
1	△10 - 6	▲15 - 5	△10 - 5	△9 - 7	▲15 - 5	△12 - 3
5	△10 - 9	▲17 - 2	△14 - 5	△11 - 8	▲16 - 3	▲14 - 5
10	△12 - 7	▲13 - 6	▲15 - 4	△14 - 5	▲14 - 5	▲15 - 4
20	△11 - 8	△12 - 7	▲16 - 3	△11 - 8	▲12 - 7	▲15 - 4
30	△11 - 8	△12 - 7	▲17 - 2	△12 - 7	△14 - 5	▲18 - 1
40	△10 - 9	△12 - 7	▲16 - 3	△12 - 7	△14 - 5	▲18 - 1
50	△10 - 9	△11 - 8	▲19 - 0	△13 - 6	△15 - 4	▲18 - 1
60	△12 - 7	△12 - 7	▲19 - 0	△14 - 5	△14 - 5	▲18 - 1
70	△14 - 5	△10 - 9	▲19 - 0	▲16 - 3	△14 - 5	▲19 - 0
80	△12 - 7	▽8 - 11	▲19 - 0	▲15 - 4	△13 - 6	▲19 - 0
90	▲14 - 5	▽7 - 12	▲19 - 0	▲17 - 2	△10 - 9	▲19 - 0
100	▲15 - 4	▽6 - 13	▲19 - 0	▲15 - 4	△10 - 9	▲19 - 0

Textual datasets

Finally, our last set of experiments considered 19 textual datasets and the cosine distance function. This set of experiments has a practical objective, as there is a growing interest in using semi-supervised clustering in the organization of textual collections. In special, these experiments are directly related to our objective of constructing semi-supervised topic hierarchies from document collections. The results of the statistical comparisons in these datasets are presented in Tables 4.12 and 4.13 and the FScores for each dataset can be observed in Figure 4.8.

According to these results, HCAC and HCAC-LC outperform all other compared algorithms in most of the comparisons. The exceptions are the intervals of very few constraints (1% of user intervention), where both algorithms perform very similar to the pairwise approach. Also, due to its inherent limit of constraints, HCAC-LC is outperformed by other semi-supervised algorithms in the intervals of very large interaction. Since all except one of these datasets have more than three clusters - which reflects the real condition of many real applications -, this scenario is very favorable to HCAC and HCAC-LC due to the nature of their constraints, as previously stated.

We can also observe that, according to the obtained performance, HCAC and HCAC-LC can use different distance functions with no impact in their active learning performance. This allows the user to select the proper distance function to be used according to the application. For example, in this experiment the clustering algorithms used the cosine distance function, which is the most adequate to textual datasets, while the euclidean distance function was used in the previous experiments. In both cases, the good performance of HCAC and HCAC-LC, when compared to the other algorithms, indicates that our active learning approaches improve the identification of informative cases.

The difference between distance functions in the active learning process is smoothed in the threshold calibration, presented in Algorithm 2. The confidence-based active clustering approach used in HCAC algorithms employs the difference between distances (the shortest and the next in order) to detect low confidence merges. Once the threshold cali-

bration process analyses all the confidences obtained during the clustering process, it can suitably detect, in that distance scale, the interval of confidence that can be considered more willing to make misclusterings.

Table 4.12: Results of statistical comparisons of HCAC on real-world textual datasets.

	5 Pairs			10 Pairs		
%	**Pairwise**	**CCL**	**Average**	**Pairwise**	**CCL**	**Average**
1	▲14 - 5	▲19 - 0	▽8 - 9	△12 - 7	▲19 - 0	▲13 - 5
5	▲15 - 4	▲19 - 0	▲18 - 1	▲14 - 5	▲19 - 0	▲16 - 3
10	▲16 - 3	▲19 - 0	▲19 - 0	▲15 - 4	▲19 - 0	▲17 - 2
20	▲16 - 3	▲19 - 0	▲18 - 1	▲16 - 3	▲19 - 0	▲17 - 2
30	▲16 - 3	▲19 - 0	▲19 - 0	▲15 - 4	▲19 - 0	▲19 - 0
40	▲18 - 1	▲19 - 0	▲19 - 0	▲17 - 2	▲19 - 0	▲18 - 1
50	▲16 - 3	▲18 - 1	▲19 - 0	▲18 - 1	▲18 - 1	▲18 - 1
60	▲17 - 2	▲17 - 2	▲19 - 0	▲18 - 1	▲18 - 1	▲18 - 1
70	▲19 - 0	▲19 - 0	▲18 - 1	▲18 - 1	▲17 - 2	▲19 - 0
80	▲19 - 0	▲19 - 0	▲18 - 1	▲18 - 1	▲17 - 2	▲19 - 0
90	▲19 - 0	▲18 - 1	▲18 - 1	▲19 - 0	▲18 - 1	▲19 - 0
100	▲19 - 0	▲18 - 1	▲19 - 0	▲19 - 0	▲18 - 1	▲19 - 0

Table 4.13: Results of statistical comparisons of HCAC-LC on real-world textual datasets.

	5 Pairs			10 Pairs		
%	**Pairwise**	**CCL**	**Average**	**Pairwise**	**CCL**	**Average**
1	▽7 - 12	▲19 - 0	▲13 - 3	▽9 - 10	▲19 - 0	▲14 - 2
5	△10 - 9	▲19 - 0	▲17 - 2	△10 - 9	▲18 - 1	▲17 - 2
10	▲12 - 7	▲18 - 1	▲18 - 1	▲12 - 7	▲18 - 1	▲19 - 0
20	▲15 - 4	▲19 - 0	▲19 - 0	▲15 - 4	▲19 - 0	▲17 - 2
30	▲14 - 5	▲19 - 0	▲17 - 2	▲14 - 5	▲19 - 0	▲17 - 2
40	▲15 - 4	▲17 - 2	▲18 - 1	▲15 - 4	▲18 - 1	▲18 - 1
50	▲15 - 4	▲17 - 2	▲17 - 2	▲16 - 3	▲19 - 0	▲17 - 2
60	▲15 - 4	▲15 - 4	▲17 - 2	▲15 - 4	▲17 - 2	▲17 - 2
70	▲15 - 4	△14 - 5	▲17 - 2	▲15 - 4	▲16 - 3	▲17 - 2
80	▲14 - 5	△12 - 7	▲17 - 2	▲15 - 4	△15 - 4	▲18 - 1
90	▲15 - 4	▽9 - 10	▲18 - 1	▲15 - 4	△13 - 6	▲18 - 1
100	▲13 - 6	▽8 - 11	▲17 - 2	▲15 - 4	△11 - 8	▲18 - 1

4.3.3 Discussion of the performance of HCAC and HCAC-LC

Considering the results obtained in our first set of experiments in a controlled environment with artificial datasets, we can observe that not querying on the merge of singleton clusters reduces the number of constraints. This approach is followed by HCAC-LC which obtained significantly better results when the amount of interaction is reduced (between 1% and 20% of the total merges). However, HCAC-LC has the limitation of querying the user for at most 50% of the cluster merges, which does not make it viable when the allowed number of queries/constraints is higher than this number. In these intervals, HCAC-LC is outperformed by the other semi-supervised algorithms based on cluster-level constraints (HCAC and CCL).

This behavior is also observed when analysing the results of the real-world numerical datasets. The HCAC algorithm performed worse than other semi-supervised algorithms when only a small number of constraints is allowed. The HCAC-LC algorithm, however, performed better than the other algorithms with a low level of interaction. In this sense,

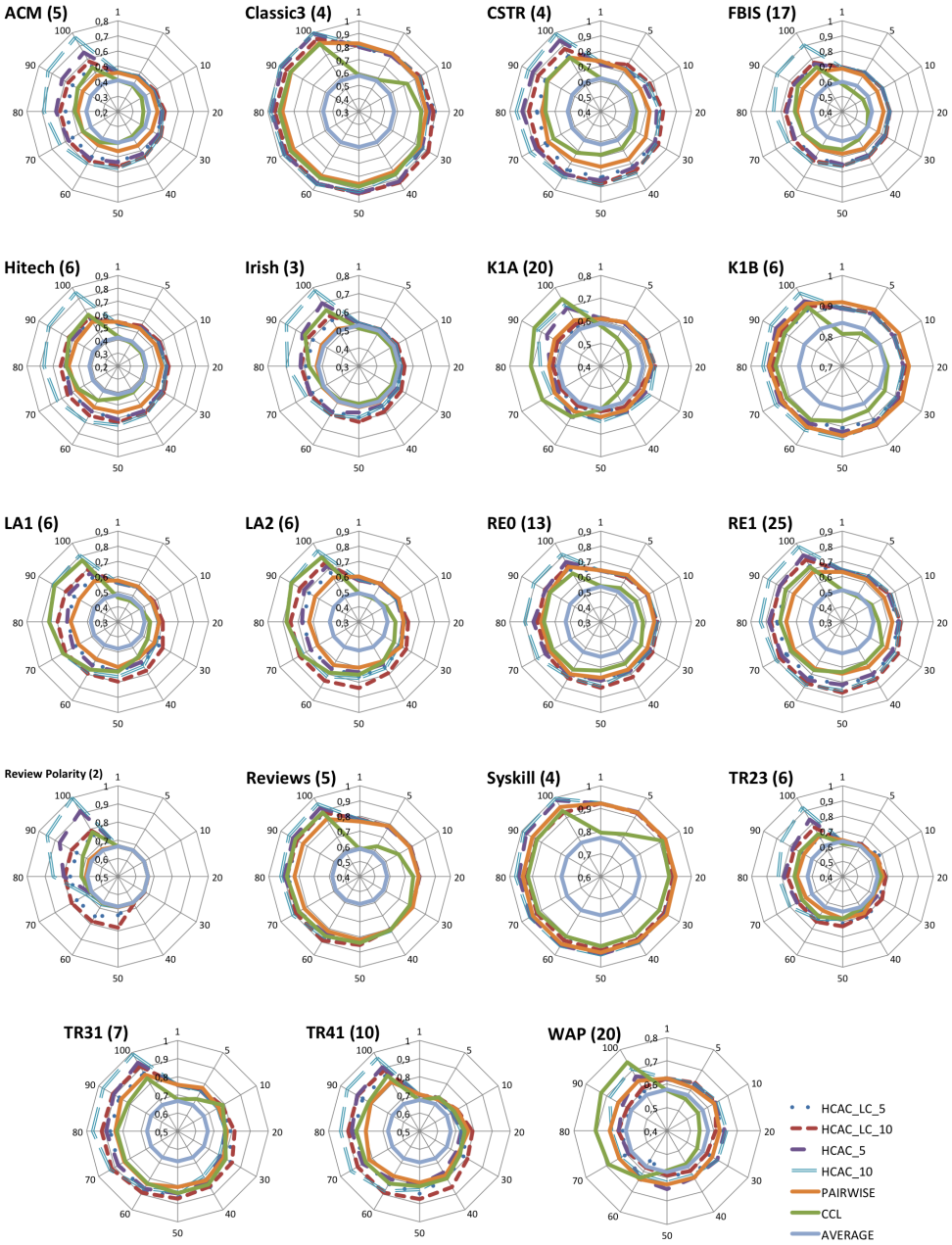

Figure 4.8: Results for textual datasets (number of clusters in parenthesis).

we strongly recommend the usage of HCAC-LC when the allowed number of interventions is small. Otherwise, the original HCAC algorithm presents better results and should be adopted.

The HCAC and HCAC-LC algorithms were also tested in textual datasets, outperforming all other algorithms considered in most of the comparisons. This also indicates that HCAC and HCAC-LC can perform well when using other distance functions, as the cosine distance function. In special, the threshold calibration and confidence calculation procedures showed robustness with respect to different distance functions. Also, these impressive results indicate that HCAC and HCAC-LC can be successfully employed in a

very important application, which is the personal organization of textual datasets.

In all sets of experiments, HCAC and HCAC-LC present slightly better results with more pairs of clusters in the pool. This is an expected result, as the more clusters are presented to the user in the pool of candidate pairs, the more options are available and the brighter the chances are of making a better choice. However, a large number of pairs in the pool implies too much human effort to analyse and correctly decide among the options and might not be practical in real applications.

Finally, the empirical results also indicate that HCAC and HCAC-LC tend to be particularly useful in datasets with a large number of clusters. This characteristic is due to the cluster-level nature of the constraints, as well as the active learning approach which helps to delimit cluster borders. In real-world datasets, HCAC and HCAC-LC presented better performance in datasets with more than 2 clusters, which is the case of many real applications.

4.4 Final remarks

In this chapter, we introduced two novel semi-supervised hierarchical clustering algorithms. The first algorithm is HCAC (Hierarchical Confidence-based Active Clustering), which was introduced during out investigation Nogueira et al. (2012a). Also, to improve the performance of HCAC when the number of constraints is small, we presented HCAC-LC (Hierarchical Confidence-based Active Clustering with Limited Constraints) algorithm, a new approach for the HCAC clustering algorithm. Both HCAC and HCAC-LC use cluster-level constraints where the user can indicate a pair of clusters to be merged. They also use an active learning process based on the concept of confidence of a cluster merge. The user's intervention is required in points where the unsupervised cluster merge presents a confidence value below a predefined calibrated threshold.

In order to reduce interaction, HCAC-LC does not consider cluster merges which involve singletons on threshold calculation, confidence calculation and on the assembly of the pool of clusters.

Experimental results lead us to conclude that HCAC-based algorithms provide efficient clustering in various scenarios. Our experiments involved the variation of different parameters, such as the number of clusters in the datasets (from just two clusters to a hundred clusters), the type of the data (from well-shaped cluster space in artificial datasets to sparse contexts in textual collections) and the distance function employed (euclidean and cosine). In all of these scenarios, HCAC-based algorithms tend to outperform the other algorithms in most of the comparisons. In summary, we can conclude from these sets of experiments that:

- HCAC and HCAC-LC achieved good and stable results in most scenarios. Results

suggest that our algorithms present high robustness to the distance function employed or data type;

- The active learning approach used in HCAC and HCAC-LC is efficient in detecting informative cases to pose queries. In a comparison with a state of the art semi-supervised clustering algorithm based in similar cluster-level constraints, HCAC-based algorithms outperformed the compared algorithm in the vast majority of scenarios;

- HCAC-based algorithms are dominant in scenarios with more than two clusters, which is the case of many real-world applications. This is mainly due to the nature of the cluster-level constraints, which allow the insertion of more information per constraint than instance-level constraints;

- HCAC-LC achieved a clear improvement over HCAC in clustering performance in scenarios with few constraints. In these scenarios, other semi-supervised algorithms perform better than the original HCAC approach but are outperformed by HCAC-LC. It is important to highlight that scenarios with limited interaction from the user have a great importance in practical applications. In these scenarios, we have to ask the user as few times as possible and still obtain sufficient information to correctly guide the clustering process;

- HCAC-based algorithms are dominant when dealing with textual datasets. Both HCAC and HCAC-LC algorithms outperformed the other compared algorithms in most of the comparisons when dealing with textual datasets. This lead us to adopt these algorithms in our work to extract topic hierarchies from document collections.

These algorithms may be improved in a near future by exploiting constraint propagation. HCAC and HCAC-LC presented significant results, achieving statistically significant improvements in comparisons to state-of-the-art clustering algorithms (both unsupervised and semi-supervised). This proved that the information inserted by HCAC-based algorithms to the clustering process is relevant and our algorithm can successfully detect points to ask for user's intervention. By using constraint propagation, we believe that we can minimize the number of constraints and achieve better results through the propagation of the added information to other instances besides the ones involved in constraints.

The application of HCAC and HCAC-LC has the disadvantage of requiring an adequate description of the groups when presenting the pairs of elements to the user. A poor description may lead the user to incorrect decisions. We are investigating adequate ways to formulate cluster-level queries so that the user can provide constraints with minimal cognitive effort, such as parallel coordinates (Inselberg, 2009) for non-textual datasets. In the next chapter we propose a concrete solution for acquiring queries from the user

in textual datasets, in the context of the SMITH framework. This framework employs HCAC-based algorithms, which has proven to be dominant when dealing with textual datasets, in the extraction of topic hierarchies for organizing document collections.

Chapter

5

SMITH: A framework for extracting topic hierarchies through semi-supervised hierarchical clustering algorithms

Topic hierarchies are an efficient way of organizing textual collections. As discussed in Chapter 2, topic hierarchies enables the user to visualize the different levels of granularity of the knowledge in the document collection. Thus, navigating through such structures facilitates an exploratory search in textual collections.

Hierarchical clustering algorithms have been applied in order to obtain topic hierarchies. However, to the best of our knowledge, there is no application of semi-supervised hierarchical clustering algorithms in the extraction of such topic hierarchies. Besides generating efficient topic hierarchies, in some scenarios unsupervised algorithms may construct cluster structures that do not meet the user preferences. Applying semi-supervised clustering algorithms would smooth this problem by considering the constraints provided by the user to limit the formation of the clusters.

The extraction of topic hierarchies involves other activities than the application of the clustering algorithms. There should be performed some activities from the selection of the data to the validation of the topic hierarchy, as suggested in the Text Mining process, presented in Chapter 2 of this thesis. In this chapter, we present SMITH (SeMI-supervised Topic Hierarchies), a framework to extract topic hierarchies through semi-supervised hierarchical clustering. The SMITH framework can be seen as an instantiation and extension of the TopTax framework (Moura et al., 2008a) and suggests a sequence of steps that should be performed in order to extract topic hierarchies. Con-

sidering the results reposted in Chapter 4, where HCAC-based algorithms (Hierarchical Confidence-based Active Clustering and Hierarchical Confidence-based Active Clustering with Limited Constraints) tend to outperform other semi-supervised hierarchical clustering algorithms in Text Mining tasks, we focus in using the HCAC-based algorithms in the SMITH framework. However, any semi-supervised hierarchical clustering algorithm would be used in the context of the SMITH framework. This framework, the tools that support its application, and a case study are discussed in the following sections.

5.1 The SMITH framework

The SMITH framework consists in a set of well defined steps and activities that aim at automatically extracting semi-supervised topic hierarchies from document collections. These topic hierarchies should be navigable by the user, in order to explore the content of document collection. All the activities proposed in SMITH are independent of language and domain, allowing its application in diverse scenarios.

SMITH both instantiates and extends TopTax. While TopTax is a methodology and provides, in each step, a set of possible activities, SMITH presents a more defined set of activities for each step. Also, SMITH extends the TopTax methodology by enabling the user's interaction during the construction of topic hierarchies. The main objective of this interaction is to provide a process that generates topic hierarchies that fit the user expectations in the organization of document collections. We consider that the addition of constraints during the clustering process in SMITH would generate topic hierarchies that fit the user expectations and minimize the user's effort to comprehend and navigate through the resulting organization of a document collection.

Similarly to the TopTax framework, the SMITH framework follows the text mining process and is composed by five steps: Problem Identification, Pre-processing, Documents Clustering, Post-processing and Knowledge Usage. These steps are described in details in the next sections.

5.1.1 Problem Identification

In the Problem Identification step of the SMITH framework, the user has to define the scope of the process and to guarantee the quality of the document collection. The subsequent steps are influenced by the decisions taken in this step.

The first decision to be taken in this step is to establish the aims of the process. It has to be defined the potential users of the topic hierarchy and to which finality it will be used for. These decisions help to choose the appropriate term representation method and to establish validation measures to evaluate the resulting topic hierarchy.

Given this information, the user has to guarantee that the document collection avail-

able are adequate to the scope of the project. The documents have to cover all the knowledge domain and with trustful and quality information. In special, as the user interacts with the clustering process, the document collection must be in accordance to his/her knowledge about the domain. Mainly, the document collection may not cover many topics that extrapolate the user's knowledge, nor be in contradiction with his/her concepts.

After these activities, it is assumed that in the end of this step the document collection is validated, complete and representative, according to the objectives of the process. Despite these concepts are subjective, they are very important to the effectiveness of the process, as the quality of the extracted knowledge depends on the quality of the data collection it is applied on. The processing of these documents begins in the pre-processing step, described in the next section.

5.1.2 Pre-processing

Once the document collection is validated, the process proceeds to the Pre-processing step. The objective of this step is to obtain a vectorial representation of the documents so that typical clustering algorithms can be applied. In the context of the SMITH framework, the vector space model is used to represent textual collections (Salton, 1989). In this model, each document is represented by a vector and each term is a dimension of the space of documents. The set of vectors obtained from all documents forms an attribute-value matrix.

In the SMITH framework, the first task of the documents preprocessing is the **documents standardization**. The framework suggests the transformation of all documents to the plain text format, as well as the removal of punctuations and mathematical symbols. SMITH also suggests the removal of general and domain specific stopwords. These words are not interesting to that specific domain, and should also be eliminated from the documents.

Once the documents are in a standard and clean format, the SMITH framework indicates the **term extraction** process. This process is responsible for detecting candidate terms that represent the textual collection. As discussed in Chapter 2, terms can be simple (composed by a single element, also known as uni-grams) or compound (composed by more than one element, also referred as n-grams). In the SMITH framework, we suggest employing both uni-grams and bi-grams (composed by two words). As the interpretation of the topics and clusters is essential to an efficient user interaction with the clustering process and to an appropriate usage of the topic hierarchies, the usage of bi-grams is very important. The main drawback in considering bi-grams is the extra computational cost, as it increases the number of generated attributes.

We refer to the detected terms as "candidate terms" because they are statistically

identified and not all of them are representative. Moreover, in general, a great number of candidate terms are identified, requiring a further selection. Our procedure of candidate term selection suggested in SMITH follows the steps proposed by Nogueira et al. (2008b). First, we reduce the words to their stem through the Porter's stemming algorithm (Porter, 1980). The stemming technique provides an aggressive yet efficient elimination of words (Conrado et al., 2012). This helps to decrease the amount of data to be processed, since there may be many words in the textual collection which vary in the writing but refer to the same concept.

The next activity consists in forming the bi-grams. Initially, every combination of two stems that appear consecutively is considered as a candidate compound term. In order to reduce the number of candidate compound terms, the SMITH framework first indicated the removal of uni-grams with document frequency (DF) below a threshold. In this initial filter, uni-grams that appear in 2 or less documents are removed. This eliminates part of the non representative uni-grams which are very rare and do not add interesting information, as well as removes possible OCR or format conversion errors. After this, the SMITH framework suggests the construction of candidate bi-grams and selects the most representative by removing bi-grams that occur in less than 4 documents.

The selection based on document frequency detects representative bi-grams. The selection of uni-grams, however, requires a more accurate procedure with a feature selection algorithm. In the SMITH framework, we suggest the adoption of the Luhn's cutoff algorithm (Luhn, 1958), described in Chapter 2 of this thesis. This method was chosen due to its simplicity, scalability and performance.

Since the terms to be considered have been selected, the resulting attribute-value matrix may be formed using these terms. In the context of the SMITH framework, we suggest the adoption of the the tf-idf measure (Salton and Buckley, 1987) to relate an attribute to a document.

In last Pre-processing action, the SMITH framework suggests the reconstruction of the terms with their original words, in a reverse procedure to the stemming process. For each term, we substitute every stem by the most frequent word that generated that stem. For example, let us consider the terms *comput* and *artific_intelli*. Let us also consider that the stem *comput* was generated most of the time from the word *computer*, the stem *artific* was generated most of the time from the word *artificial* and the stem *intelli* was generated most of the time by reducing the word *intelligence*. Then, we replace these stems by their most frequent words, forming the terms *computer* and *artificial_intelligence*. This replacement procedure is performed to improve the comprehensibility of the terms. This comprehensibility is essential to a correct user interaction during the clustering process and to an efficient navigation and usage of the topic hierarchy.

5.1.3 Pattern extraction: documents clustering and topics detection

Once the document collection is pre-processed, the semi-supervised clustering algorithm may now be applied to the data. The SMITH framework uses the HCAC-based algorithms, introduced in Chapter 4. In particular, as the experimental results indicate that HCAC-LC performs better than HCAC in scenarios with limited user interventions, we recommend the adoption of this algorithm in the SMITH framework. However, any other hierarchical semi-supervised clustering algorithms may be used with minor modifications in this step.

The first activity in this step is to form a distance matrix, using the cosine distance function, as described in Chapter 2. Then, the HCAC-based algorithms should be applied over this matrix and the user should be queried in low confidence merges. In this sense, a confidence threshold is calibrated according to the desired amount of constraints to be inserted by the user. The more constraints are expected, the greater is the confidence threshold value. In the construction of topic hierarchies, the more the user interacts with the clustering process, the more adequate to his/her needs the topic hierarchy will be. However, a large number of interactions requires an excessive cognitive effort from the user. So, the number of constraints to be required from the user must be determined according to the application. The following factors should be considered:

- **The number of documents in the collection**: the more documents there are in a document collection, the more clusters merges will be done during the clustering process (the number of merges is equals to the number of documents minus one). Thus, more user interactions are needed in order to correctly guide the cluster formation and to an effective information addition.

- **The diversity of themes in the dataset**: while constructing the topic hierarchies, an optimal solution would allocate different themes in different branches of the cluster hierarchy. The more themes a dataset covers, the more branches (clusters) are detected. Thus, more information is required to achieve an adequate solution in scenarios with a big number of different themes. This is arising from the problem of dealing with big number of classes, discussed in Chapter 4.

- **The user availability**: the user availability is a counterpoint of the two factors cited above. While increasing the number of documents and themes in the dataset implies in more interventions from the user to achieve an optimal solution, considering the user availability, in general, leads us to enquire the user as fewer times as possible. In most of the applications, the user is available in a short period of time. Moreover, if a specialist in the knowledge domain is hired to interact with the clustering process, probably there is a cost associated to the time he/she spends

(i.e., the more time the specialist interacts with the process, the more expensive it gets).

- **The cost of a misclustering**: the interaction of the user with the clustering algorithm aims at avoiding misclusterings. When deciding the amount of user interactions to be required, it should be considered how harmful a misclustering is to the objective of the process. This is a contrast to the cost of the user interaction. If an error has a very high cost, superior to the cost of querying the user, then it would be worthy querying the user as many times as possible. As discussed in Section 3.7 of this thesis, this dichotomy between the cost of an error and the cost of an interaction is still an open question in semi-supervised clustering and should be subjectively analysed by the user.

In the SMITH framework we consider that an adequate quantity of user interactions with the HCAC-based algorithms would be between 10% and 25% of the total cluster merges. Less than 10% of interaction would not be significant, while more than 25% would imply in excessive work for the user in most of the cases. However, this interval is not fixed and the determination of the amount of interaction depends on the context and would consider the three factors cited above.

Once the desired amount of interaction is determined, the HCAC-based algorithm can calibrate the confidence threshold, as described in Algorithm 2 in Chapter 4. When a low confidence is detected, a pool of candidate pairs of clusters is presented to the user in order to decide the best merge. The size of this pool may also vary according to the context. Obviously, the more cluster there are in the pool, the brighter are the chances of making the best choice. However, the addition of the number of pair of clusters implies in an extra cognitive effort from the user in order to correctly analyse them. According to the experimental results presented in Chapter 4, using 5 or 10 pairs in the pool of candidates is enough to achieve a significant improvement in clustering quality when compared to other algorithms.

In order to smooth the problem of the amount of cognitive effort demanded from the user, an adequate representation of the clusters in the pool of candidate pairs should be adopted. In the SMITH framework, it is suggested the presentation of the clusters to the user through two possible representations: a list of the titles of the documents or a list of main words. If the cluster has 10 or less documents, then a list of the titles of these documents should be shown to the user. Otherwise, if the cluster contains more than 10 documents, reading the list of titles would be hard for the user. Then, a more concise representation is needed. In these cases, SMITH suggests the presentation of a list of the main terms of these documents. These terms should be selected through the FScore measure (Forman, 2003), which is a simple, scalable and effective method. From the experiences reported in Moura et al. (2008a) and Moura and Rezende (2010), for most

of the applications, a set of cluster descriptors composed by 5 to 10 terms would be easy to understand and would still being representative.

When the clustering process finishes, the resulting dendrogram forms a hierarchical structure of significant groups. Each cluster in the hierarchy is assumed to be formed by similar documents and belong to one topic. In order to form a navigable and comprehensive topic hierarchy, each cluster needs to be labeled with its main terms. In the SMITH framework, the labeling process is carried out using the FScore method, in a procedure similar to the obtaining of the cluster descriptors. The calculation described in Equation 2.3 is repeated for each term and each cluster in the hierarchy. Then, the top k terms are selected to be the cluster labels.

5.1.4 Hierarchy post-processing

The formed topic hierarchy must be validated by the user, according to the objectives defined in the Problem Identification step. The user must identify if the topic hierarchy achieves its objectives and efficiently helps the navigation and exploration of the document collection.

In order to evaluate the obtained hierarchy, an evaluation measure should be adopted. In the SMITH framework, it is suggested to measure the quality of a topic hierarchies through metrics collected during the interaction of the user with these hierarchies. For example, there should be measured the recall and the time elapsed in the search for documents in the hierarchy.

The evaluation of the topic hierarchy should be carried out assisted by navigation and visualization tools (Moura et al., 2008a; Marcacini and Rezende, 2010b). Moreover, by navigating through the topic hierarchy, the user can detect possible problems and, if needed, the process may return to one of the previous steps, given the iterative nature of the text mining process.

5.1.5 Knowledge Usage

Once the topic hierarchy is validated, it is ready to be used and efficiently explored by the final user. It should be useful ins tasks like information organization and retrieval, as well as to help to comprehend the document collection.

In this step, visualization techniques and tools may also be used to assist the user. These tools facilitate recognizing relationships, tendencies and patterns in the data set analysed, boosting the knowledge exploration. Examples of tools that support this and the other steps of SMITH are presented in the next section.

5.2 Tools for the application of SMITH

The application of the SMITH framework by the user is supported by a series of computational tools. Here, we point out some of the possible tools that would be employed by the user during the complete process of the SMITH framework. All the tools suggested here follow the same input / output schema and can be applied in sequence without any adaptations.

First, for the documents preprocessing, the PreText tool (Soares et al., 2008) should be used. This tool supports the the stemming of the terms, the extraction of simple and compound terms, the generation of the attribute-value matrix and the terms selection through the Luhn's cutoff method. To the stemming process, PreText uses the Porter's algorithm (Porter, 1980), for Portuguese and English languages. The selection of compound terms is done by establishing a minimum of documents in that the compound term occurs. The Pretext tool receives as input a base of plain text documents and generates the attribute-value matrix in the Discover format (Prati et al., 2003).

The resulting attribute-value matrix is ready to be clustered. Then, for the application of the HCAC-based algorithm, the HCAC Tool[1] should be used. This is a tool developed during our research and supports the application of the HCAC-based algorithms. When a low confidence merge is detected, the HCAC tool presents the candidate pairs to the user, as shown in Figure 5.1. In the end of the clustering process, the HCAC tool generates the descriptors for each cluster, applying the FScore measure. This hierarchical structure and the descriptors for each cluster are stored in an XML file, following the format suggested by Marcacini and Rezende (2010b).

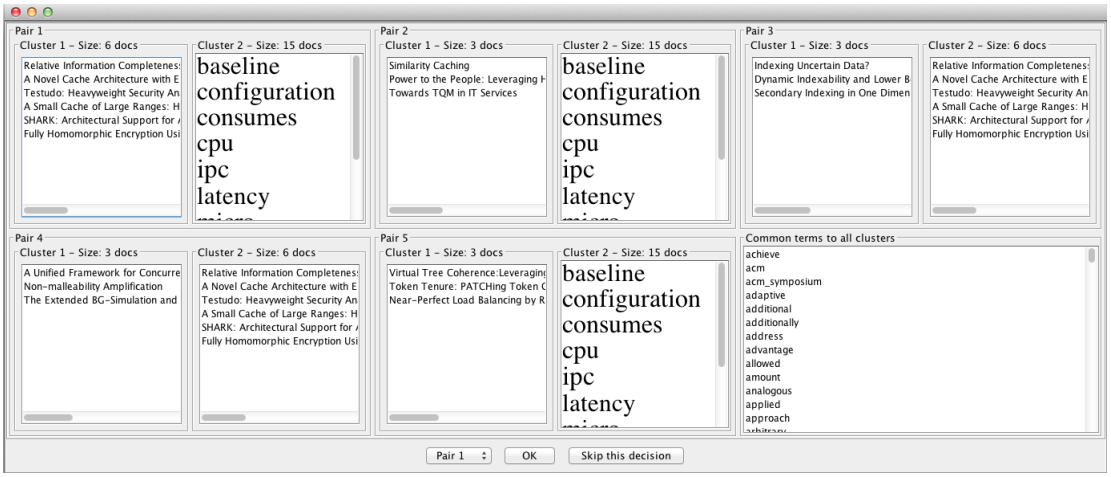

Figure 5.1: Interface for user interaction in the HCAC Tool.

After the clustering process, the user should use the DProcessor tool[1] to validate the topic hierarchy. This tool was developed in the Laboratory of Computational Intelligence

[1]Available at: http://sites.labic.icmc.usp.br/bmnogueira/hcac

(LABIC - ICMC/USP) and collects metrics while the user navigates through a topic hierarchy searching for specific documents. The interface of this tool can be observed in Figure 5.2. It is possible to determine a number of documents to be searched for and a maximum search time for each document. In the end, the DProcessor tool presents a summary containing, among other measures, the number of documents found and the time elapsed in the search for each document.

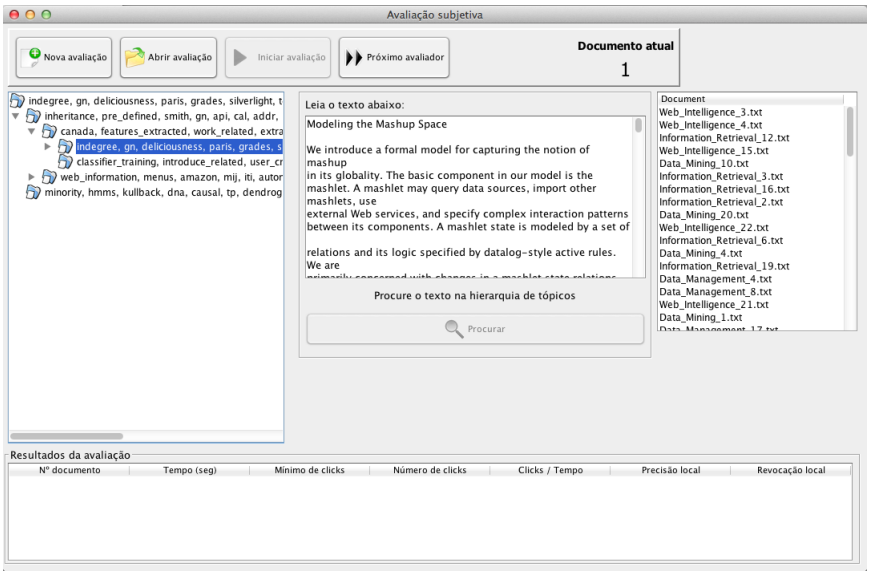

Figure 5.2: Interface of the DProcessor Tool.

With a validated topic hierarchy, the topic hierarchies should be deployed. The usage of these hierarchies demand the use of visualization tools, so that the user can navigate through the hierarchical organization of the document collection. We suggest the usage of the Torch Tool (TOpic HieraRCHies) (Marcacini and Rezende, 2010b), which contains a module that allows the visualization and navigation in topic hierarchies. An example of the exhibition of a topic hierarchy in the Torch Tool is shown in Figure 5.3. In the part A of this figure, the simple exhibition of the hierarchy is shown, using a list of topic. When the user clicks in one topic, the list of documents relative to that topic is shown. In the part B of this figure, an alternative exhibition of the topics is shown, which exhibits the relation between parent and child topics. Both browsing schemas are efficient in helping the user to retrieve and view the documents in the collection.

5.3 Case study

In order to assess the efficiency of the HCAC-based algorithms in obtaining topic hierarchies from document collections, we applied the HCAC-LC algorithms with the SMITH framework. In our case study, three different document collections of scientific papers were used. All of these datasets contained documents written in English from 4

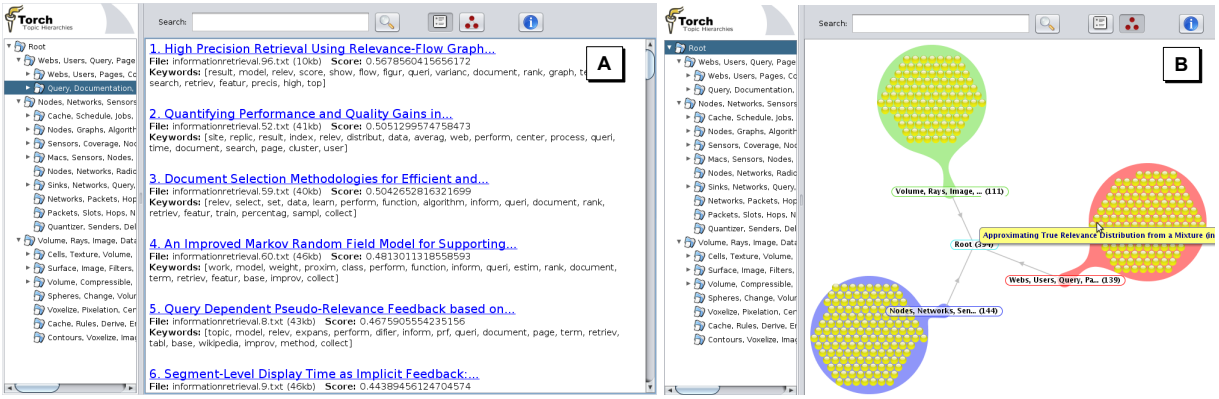

Figure 5.3: Visualization of a topic hierarchy in the Torch Tool.

different areas of Computer Science, distributed as follows:

- Collection Computer Science 1: Data Management, Data Mining, Information Retrieval and Web Intelligence;
- Collection Computer Science 2: Database Systems, Microarchitecture, Software Engineering and Theory of Computing;
- Collection Computer Science 3: Embedded Systems, Mobile Systems, Software Reusability and Virtual Reality.

These collections are perfectly balanced, with 20 papers per area, in a total of 80 documents in each dataset. We opted for using different datasets containing diverse areas of Computer Science in order to minimize the bias in the evaluation caused by the different levels of knowledge of the users in some specific area. To support this case study, use used the PreText, HCAC and DProcessor tools, described in the previous section of this thesis.

In the next section, we describe our experiments by presenting the preprocessing of the collection and the topic hierarchy obtaining and validation.

5.3.1 Applying the SMITH framework

The three document collections were processed under the SMITH framework, in order to obtain one topic hierarchy for each of them. These topic hierarchies are intended to be used by specialists in Computer Science in order to explore the contents of the document collections. A summary of the experimental configuration used in this case study is described in Table 5.1.

In the Pre-processing step, this case study followed the procedures suggested in the description of the SMITH framework. After standardizing the document collections and removing unnecessary characters, candidate terms were extracted and then the main terms were selected. Simple and compound terms were generated to construct the attribute-value matrix. The number of generated terms can be observed in Table 5.2.

Table 5.1: Experimental configuration used in the case study

Problem identification	
Textual Dataset Language	English
Domain	Computer Science scientific papers
Pre-processing of the textual dataset	
Documents standardization	
Format standardization	Conversion from pdf files to plain text files
Cleaning the texts	Punctuations removal
	Removal of numbers and mathematical symbols
	Removal of non informative symbols (like @#$%...)
	Removal of English stopwords
Normalizing the words	Stemming
Candidate terms extraction	
N-grams considered	One-grams and bi-grams
Type of representation	Document-term matrix
Term weighting measure	Term frequency - inverse document frequency (tf-idf)
Term selection	
Minimum document frequency of the term	One-grams - 2
	Bi-grams - 4
Method for selecting one-grams	Luhn's method
	Minimum TF - 10
	Maximum TF - 500
Pattern extraction	
Documents clustering	
Distance measure	Cosine
Clustering algorithm	HCAC-LC
Number of pairs of clusters in the pool	5
Maximum user interaction	10% of the total merges (8 per dataset)
Cluster descriptors selection measure	FScore
Number of cluster descriptors shown to user	10
Descriptors extraction	
Cluster label selection measure	FScore
Number of labels per cluster	10
Post-processing of Topic Hierarchies	
Validation approach	Objective metrics collected through user navigation
	Objective supervised measures through FScore
	Comparison with metrics collected in topic hierarchies constructed with unsupervised clustering
Number of users	10
Documents to search	5 documents randomly selected
Maximum search time	180 seconds
Measures collected	Time elapsed
	Number of documents found

Table 5.2: Number of terms and confidence threshold value for each dataset

Document Collection	Number of terms	Confidence threshold
Computer Science 1	4162	0.0036
Computer Science 2	4188	0.0037
Computer Science 3	4558	0.00634

From the attribute-value matrices, distance matrices were generated. Each document collection was represented by a distance matrix with dimensionality 80 × 80. The inner cells were filled with the cosine distance between the documents. The HCAC-LC

algorithm was then applied over these matrices. First, an unsupervised procedure was applied, in order to calibrate the confidence threshold value. When applying the HCAC-LC algorithm, cluster merges that present confidence values below the thresholds were considered as low confidence merges and the user's intervention were required. The value of the confidence threshold for each dataset can be observed in Table 5.2.

When a low confidence merge was detected, the user was invited to provide a constraint. In these cases, 5 pairs of candidate clusters were shown to the user. As suggested in the SMITH framework, if the cluster contains 10 or less documents, the titles of these documents were shown. Otherwise, a list containing the 10 most discriminative terms were exhibited.

When the clustering process finishes, the cluster hierarchy is formed. In order to obtain a topic hierarchy, these clusters must be labelled with their main terms. In this case study, a total of 10 labels per cluster were selected using the FScore measure.

After this process, the topic hierarchies are ready to be explored and validated by the users. In this case study, 10 users which are researchers in Computer Science were invited to use SMITH for obtaining a hierarchy of the documents for each dataset. These users are Ph.D. students and 9 of them work in Artificial Intelligence and 1 in Software Engineering. For each of these users, their interactions with the clustering process lead the construction of a personalized topic hierarchy for each dataset. An unsupervised clustering algorithm (average-link algorithm) was also applied in each dataset, constructing unsupervised topic hierarchies.

Each topic hierarchy was evaluated in two groups of analysis, as follows:

1. **Using the FScore measure**: we considered the area of the document as its class. Thus, we traversed the topic hierarchy calculating the FScore measure of each cluster, as described in Equation 4.2 presented in Section 4.3.1 of this work. The objective of this evaluation is to measure how domain specialists information can help in practice to form clusters that attend a previous standard categorization. Considering that the users were aware of the existing underlying categorization, our hypothesis is that the user's constraints would help to organize the collection according to this categorization. Thus, the constraints provided by domain specialists may lead the formation of a cluster structure with a higher FScore measure.

2. **Using metrics collected during the user navigation through the hierarchies**: for each hierarchy (both semi-supervised and unsupervised), the users were invited to search for 5 documents chosen at random. The users had 180 seconds to find each document. During the search for the documents, we measured the time spent in the search and the number of the documents correctly retrieved. The objective of this evaluation is to measure how the resulting hierarchies considering the information from the user can help the visualization and comprehension of the docu-

ment collection. The hypothesis in this case is that semi-supervised topic hierarchies facilitate the comprehension of the knowledge present in the document collection, since they are formed according to the user constraints. Thus, the exploration of such structure is easier than the exploration of an unsupervised hierarchy.

The results of these experiments are discussed in the next section.

5.3.2 Results and discussion

The first results discussed in this section refers to the objective evaluation through the FScore measure. These results are reported in Table 5.3.

Table 5.3: FScore values obtained from the semi-supervised and unsupervised topic hierarchies.

User	Computer Science 1		Computer Science 2		Computer Science 3	
	HCAC	**Avg. link**	**HCAC**	**Avg. link**	**HCAC**	**Avg. link**
1	0.52763423	0.51248606	0.77446103	0.75166957	0.85608637	0.8510989
2	0.58453805	0.51248606	0.75349914	0.75166957	0.82820513	0.8510989
3	0.52542109	0.51248606	0.7496711	0.75166957	0.88021685	0.8510989
4	0.52244941	0.51248606	0.741252	0.75166957	0.87167846	0.8510989
5	0.50453805	0.51248606	0.7234185	0.75166957	0.84871795	0.8510989
6	0.51319606	0.51248606	0.73604705	0.75166957	0.83839406	0.8510989
7	0.51897622	0.51248606	0.77856752	0.75166957	0.8510989	0.8510989
8	0.51114114	0.51248606	0.67113344	0.75166957	0.84871795	0.8510989
9	0.53594273	0.51248606	0.75675501	0.75166957	0.86878907	0.8510989
10	0.5055918	0.51248606	0.7411211	0.75166957	0.82014652	0.8510989
Average	0.52494288	0.512486065	0.742592589	0.75166957	0.851205126	0.851098901

According to the results, in average, HCAC-LC achieved better FScores values than the unsupervised clustering algorithm in two of three datasets. In particular, the best results were achieved by the HCAC-LC algorithm in the Computer Science 1 dataset. The areas of this dataset are more related to the Artificial Intelligence, which is the research area of most of the users. Thus, the users are more confident when interacting with the clustering process in this dataset and tend to provide useful information that is in accordance with the standard classification.

In the other two datasets, the hierarchy formed by the interaction of the users were very similar to the unsupervised hierarchies. One possible reason for that behaviour is that the users were not as familiar to these areas of the Computer Science as in the first dataset. This way, the users' expectations on the documents clustering were different from the standard classification. As an example, the user 7 in the dataset Computer Science 3 felt he/she was not able to decide the best clustering merges and skipped every clustering decision.

In summary, this first evaluation confirmed our experimental hypothesis and showed that the more familiar an user is to the problem domain, the more his/her cluster structure will be similar to the standard classification. However, it is important to highlight that in most applications, this similarity is not important. The user may have his/her own

expectations in the classification of the document collections, which is independent to the standard classification of the documents.

In order to investigate the ability of adapting the organization of the document collection to the expectations of the user, we performed the second set of experiments. The users were invited to search for documents which were organized under the topic hierarchy. The results of the collected metrics can be seen in Table 5.4.

Table 5.4: Results of the navigation of the users through semi-supervised and unsupervised topic hierarchies.

User	Computer Science 1				Computer Science 2				Computer Science 3			
	HCAC		Avg. link		HCAC		Avg. link		HCAC		Avg. link	
	Found	Time	Found	Time	Found	Time	Found	Time	Found	Time	Found	Time
1	0	900	0	900	1	826	0	900	0	900	0	900
2	3	713	2	728	5	375	4	563	4	392	3	644
3	3	728	0	900	5	444	3	588	5	350	3	543
4	0	900	0	900	1	785	3	782	3	677	2	701
5	2	776	1	783	1	866	0	900	0	900	0	900
6	1	847	1	868	4	411	4	544	5	425	4	613
7	0	900	0	900	1	813	0	900	1	840	0	900
8	3	753	1	855	1	849	1	812	2	822	2	780
9	4	622	3	711	5	229	4	288	5	255	4	624
10	0	900	0	900	0	900	0	900	1	853	0	900
Avg.	1.6	803.9	0.8	844.5	2.4	649.8	1.9	717.7	2.6	641.4	1.8	750.5

First, the results indicate that the user familiarity with the problem domain is not a problem when exploring the topic hierarchies. In opposite to the FScore results, the navigation results indicate that the users presented similar performances in the three different datasets. There is not much difference in the performance measures when retrieving documents in the Computer Science 2 and 3 datasets, in which most of them were not as familiar with the themes as in the Computer Science 1 dataset. This shows that the users were able to comprehend the topic hierarchies and were able to correctly navigate through the document collection.

According to the results collected, the topic hierarchies constructed using the HCAC-LC algorithm allow the users to find more documents and spending less time than the unsupervised topic hierarchies. In all three datasets, in average, the semi-supervised topic hierarchy presented a recall lift and a speedup in the search for documents when compared to the unsupervised topic hierarchies. Considering the three datasets, only one user, in one specific case, found more documents using the unsupervised topic hierarchy than when using the semi-supervised topic hierarchy. This confirms that the usage of HCAC-LC in the SMITH framework provides an effective way of personalizing the document organization and leads to a better exploration of the document collection.

It is also possible to notice that some users were not able to find some of the required documents while navigating through the hierarchies. The main factor that explains this problem is the great number of levels the users had to navigate before reaching the leaf node where the document was inserted. In the context of the SMITH framework, we decided to do not adopt a pruning process in the cluster hierarchies in order to avoid

introducing a bias on the clustering results. When navigating through a great number of levels, an incorrect decision in one level may require much time to be detected and corrected. Thus, when the user made a mistake in the path until the leaf node, in general he/she was not able to find the required document.

Table 5.5: Average results per user.

User	Found		Time		Fscore	
	HCAC	Avg. link	HCAC	Avg. link	HCAC	Avg. link
1	0.33333333	0	875.333333	900	0.71939388	0.70508485
2	4	3	493.333333	645	0.72208078	0.70508485
3	4.33333333	2	507.333333	677	0.71843635	0.70508485
4	1.33333333	1.66666667	787.333333	794.333333	0.71179329	0.70508485
5	1	0.33333333	847.333333	861	0.69222483	0.70508485
6	3.33333333	3	561	675	0.69587906	0.70508485
7	0.66666667	0	851	900	0.71621421	0.70508485
8	2	1.33333333	808	815.666667	0.67699751	0.70508485
9	4.66666667	3.66666667	368.666667	541	0.72049561	0.70508485
10	0.33333333	0	884.333333	900	0.68895314	0.70508485
Average	2.2	1.5	698.366667	746.9	0.70624686	0.70508485

In order to perform a statistical comparison of the results, we computed the average results per user. These values can be observed in Table 5.5. Over these values, we applied the Student Paired T-Test, in order to verify if there is statistical difference in the results of the semi-supervised topic hierarchies when compared to the unsupervised ones. Assuming a p-value of 0.05, the results indicate that topic hierarchies obtained by the HCAC algorithm statistically outperform the topic hierarchies obtained by the Average-link algorithm in terms of number of documents found and time elapsed during the search. The p-values found for these two metrics were 0.00546 and 0.00517, respectively. On the other hand, no statistical difference was found in terms of the FScore evaluation measure, presenting a p-value of 0.41284.

Finally, it is necessary to highlight that the experiments reported in this section required a very small amount of user interaction. As the users had, in general, not much available time to perform the experiments, we chose to ask the user intervention in the process in at most 8 cluster merges per dataset (10% of the total cluster merges). Thus, it is possible to consider that the user provided information is very limited. Even in this scenario, the usage of the HCAC-LC algorithm under the SMITH framework achieved impressive results in helping the user in comprehending and navigating through the document collections. This is a promising evidence that HCAC-based algorithms are useful in extracting topic hierarchies that fit the user's expectations.

5.4 Final remarks

In this chapter, we presented SMITH (SeMI-supervised Topic Hierarchies), a framework for generating topic hierarchies using semi-supervised hierarchical clustering algorithms. The SMITH framework is an innovative proposal to extract topic hierarchies

that fit the user's expectations. SMITH extends and instantiates the TopTax framework (Moura et al., 2008a), proposed in collaboration with the author of this thesis.

The SMITH framework can be seen as an instantiation of the Text Mining process and is composed by five steps: (i)Problem identification; (ii) Documents pre-processing; (iii) Pattern extraction - documents clustering and topics detection; (iv) Hierarchy post-processing; and (v) Knowledge usage. The tasks suggested in the SMITH framework are simple and effective, allowing its application in different kind of document collections. The SMITH framework is designed to employ semi-supervised hierarchical clustering algorithms. In particular, we employ the HCAC-based algorithms which have proven to outperform other state-of-the-art semi-supervised hierarchical clustering algorithms. However, the SMITH framework supports any semi-supervised hierarchical clustering algorithm.

In order to test the SMITH framework, we carried out a case study. We invited 10 users to interact with the SMITH framework, using the HCAC-LC algorithm in three different datasets containing Computer Science scientific papers. We evaluated in two sets of analysis: using the FScore measure (assuming the themes as the classes of the documents) and using metrics collected during the users' navigation through the topic hierarchies when searching for documents. The same metrics were collected considering unsupervised topic hierarchies, in order to compare the results. In general, the semi-supervised topic hierarchies achieved better results, specially in the navigation metrics. Even considering a small amount of interaction (10%), the usage of semi-supervised hierarchical clustering lead the construction of more user-friendly topic hierarchies.

We consider that the results reported here may be improved in future experiments by considering some improvements in the SMITH framework. For example, we would test different methods for obtaining the descriptors of the clusters in the Pattern extraction step. Besides the FScore measure is able to detect statistically significant descriptors, in some cases non-informative words are still exhibited to the user. For example, in some cases the name of institutions and formula variables were taken as descriptors, but clearly do not represent the content of the clusters. As example of methods to be tested, we cite the RLUM method (Moura and Rezende, 2010), which have achieved interesting results in labelling hierarchical cluster structures. Moreover, we would like to perform experiments considering more user interactions with the HCAC-based algorithms. We believe that a small increment in the number of interactions would bring a substantial improvement in the quality of the topic hierarchies.

Chapter

6

Conclusion and future work

Text Mining processes have been widely employed in the last years in order to aid the organization of document collections and the management of information. In particular, the organization of document collections under a hierarchical structure is a very intuitive form of organizing a data collection, since it provides a visualization of the data in different levels of abstraction. A particular case of these hierarchical structures are the topic hierarchies, also known as topic taxonomies, which associate a set of topics to each cluster in the hierarchy. These topics describe the content of the documents associated to the clusters. Under this representation, the document collection can be navigated through a set of topics and subtopics.

Topic hierarchies can be obtained through unsupervised clustering algorithms. These clustering algorithms, however, are not able to capture the individual preference of the user about how to group the objects. Therefore, these algorithms may not produce a hierarchical organization of textual datasets that are according to the user's needs. In this scenario, semi-supervised hierarchical clustering algorithms emerge as a useful solution for extracting topic hierarchies. These algorithms allow the user to guide the cluster formation by actively, and parsimoniously querying the user for constraints.

The objectives of this work were directly related to these two themes. The first objective was to investigate semi-supervised hierarchical clustering algorithms and to propose algorithms that improve both the selection of informative cases to add constraints and the user interaction with the process. Our second objective was to construct a framework for extracting semi-supervised topic hierarchies, using semi-supervised clustering algorithms, in a procedure similar to the TopTax methodology.

These objectives were motivated by two research gaps related to topic hierarchies and

semi-supervised hierarchical clustering. First, there were no appropriate solutions for semi-supervised hierarchical clustering. In particular, the active learning approaches and the kind of the queries employed were not convincing. The improvements in the clustering performance brought by the existing active learning approaches were not significant. Moreover, most of the existing work focused in using pairwise constraints, which can carry limited information. Moreover, we found no methodology to extract semi-supervised topic hierarchies from document collections.

The second gap is related to the absence of a methodology for extracting semi-supervised topic hierarchies. Topic hierarchies were constructed following unsupervised procedures and sometimes may not be in accordance with the user's expectations. Topic hierarchies that consider the user's knowledge were constructed manually and presented a high cognitive effort.

In order to fulfil these objectives, our research was based on three basic hypothesis:

1. *Topic hierarchies are efficient ways of representing the knowledge present in document collections*: this hypothesis was confirmed through our experiments presented in Chapter 5. According to the results, unsupervised and semi-supervised topic hierarchies were able to summarize the content of the document collections in a comprehensive and navigable way to the users. The users were able make correct decisions when navigating through the topic hierarchies by interpreting the underlying hierarchical structure of clusters and the descriptors of each cluster. This led to a minimization of the effort demanded from the user in the exploration of the contents of document collections.

2. *The incorporation of the user's knowledge provides more intuitive topic hierarchies*: this hypothesis was also confirmed in the experiments presented in Chapter 5 of this thesis. According to the results, when the users were asked to search for documents in the topic hierarchies, semi-supervised topic hierarchies achieved significant improvement when compared to unsupervised topic hierarchies. There was statistically significant difference in terms of the number of documents retrieved and time consumed to find a document. Thus, it is possible to assume that the semi-supervised topic hierarchies were more in accordance with the users' expectations than the unsupervised topic hierarchies.

3. *It is possible to efficiently incorporate the user's knowledge to a clustering process through constraints*: the confirmation of this hypothesis arises from the results presented in Chapters 4 and 5. In these experiments, we considered semi-supervised algorithms that elicit constraints from users. According to the results presented in Chapter 4, where we considered "simulated" users, semi-supervised clustering algorithms outperformed unsupervised clustering algorithms in diverse scenarios. In

particular, the constraints actively inserted by HCAC-based algorithms, proposed in this thesis, presented an impressive performance, outperforming all other algorithms in most of the comparisons. The HCAC-LC algorithm was used in the experiments presented in Chapter 5 of this thesis to elicit constraints from real users. Even considering a small number of constraints, positive results were achieved during the user navigation through the semi-supervised topic hierarchies, according to the collected metrics. This indicates that these hierarchies are in accordance with the users' expectations and that the constraints are an efficient way of incorporating the user's knowledge to the process.

During our research, some contributions to the state of the art of semi-supervised clustering and automatic extraction of topic hierarchies were achieved. The main contributions achieved, as well as the limitations and indications of future work are listed in the following sections of this chapter.

6.1 Contributions

The first contribution of this work, presented in Chapter 3, is related to the review of the research in semi-supervised clustering. We listed and analysed the main works in semi-supervised clustering. There is no such comparison available in the literature. We strongly believe this survey may help the research in semi-supervised clustering by providing a general overview of the area, a classification of the methods and approaches, the main methods available and research gaps and perspectives in the area. An article containing this survey is in final preparation and is going to be submitted to a journal of high impact.

The second contribution of this work, reported in Chapter 4 is related to the proposal new semi-supervised hierarchical clustering algorithms. Two new semi-supervised hierarchical clustering algorithms were proposed: HCAC (Nogueira et al., 2012a) and HCAC-LC. These algorithms achieved impressive clustering performance, outperforming some of the state-of-the-art algorithms. These are multi-domain clustering algorithms, but achieved better results in clustering textual data. HCAC is indicated to scenarios where it is possible to obtain more constraints, while HCAC-LC is designed to improve clustering quality with a smaller number of constraints. Both HCAC and HCAC-LC are dominant in scenarios with more than three clusters, which is the case of most of the real-world applications.

In the context of the HCAC-based algorithms, we proposed a new active learning approach for hierarchical semi-supervised clustering. In Chapter 4, we introduced the concept of confidence of a cluster merge. This concept allows the user to intervene in regions near to cluster borders, which boosted the impact of the constraints by helping

to delimit the cluster boundaries (Nogueira et al., 2012b).

Also in that chapter, we introduced a new kind of query for semi-supervised hierarchical clustering. In HCAC-based algorithms, the user is invited to provide cluster-level constraints, which can carry more information than instance-level constraints. In each constraint, the user indicates the best merge to be done during an agglomerative hierarchical clustering procedure. For this analysis, a pool with candidate pairs is assembled, reducing the user's effort in analysing the different possibilities.

As a related contribution, in Chapter 4 we presented an extensive comparison of state-of-the-art semi-supervised hierarchical clustering algorithms in different clustering scenarios (Nogueira et al., 2012a, 2013). To our knowledge, there was no such comparison in the literature. This comparison would help researchers in detecting strong and weak points of different algorithms and approaches, helping to decide which algorithm to employ.

The other contributions of this work, reported in Chapter 5 are related to the proposal of a framework to extract semi-supervised topic hierarchies. In this chapter, we proposed SMITH, a new framework for extracting semi-supervised topic hierarchies which is based in the TopTax methodology. We also presented an interface for user's interaction in the extraction of topic hierarchies through the HCAC Tool. According to the experimental results, the hierarchies obtained with the semi-supervised process efficiently help the user in organizing document collections, easing the navigation through the hierarchies.

We highlight that some of these contributions introduced innovative ideas to the state of the art of semi-supervised clustering and automatic generation of topic hierarchies. To the best of our knowledge, the queries type and the active learning approach used in the HCAC-based algorithms were not explored before our work. Moreover, the SMITH framework is the first proposal in the literature to support the extraction of semi-supervised topic hierarchies.

6.2 List of publications

The above contributions lead the publication of 4 conference papers directly related to this research:

- **NOGUEIRA, B. M.** ; JORGE, A. M. ; REZENDE, S. O. . On the Comparison of Semi-Supervised Hierarchical Clustering Algorithms in Text Mining Tasks. In: 1st Symposium on Knowledge Discovery, Mining and Learning (KDMiLe), 2013, São Carlos. Proceedings of the 1st Symposium on Knowledge Discovery, Mining and Learning (KDMiLe), 2013. p. 1-8.

- **NOGUEIRA, B. M.** ; JORGE, A. M. ; REZENDE, S. O. . HCAC: Semi-supervised Hierarchical Clustering Using Confidence-Based Active Learning. In:

15th International Conference on Discovery Science, 2012, Lyon. Lecture Notes in Computer Science. Alemanha: Springer Verlag, 2012. v. 7569. p. 139-153.

- **NOGUEIRA, B. M.** ; JORGE, A. M. ; REZENDE, S. O. . Hierarchical confidence-based active clustering. In: 27th Symposium On Applied Computing (ACM SAC), 2012, Riva del Garda, Itália. Proceedings of 27th Symposium On Applied Computing, 2012. v. 1. p. 535-536.

- **NOGUEIRA, B. M.** ; REZENDE, S. O. . Análise comparativa de duas abordagens para o agrupamento semi-supervisionado particional de documentos: sementes iniciais e restrições pareadas. In: IV Congresso da Academia Trinacional de Ciências (C3N 2009), 2009, Foz do Iguaçu, PR. Anais do IV Congresso da Academia Trinacional de Ciências. Foz do Iguaçu, PR: Unioeste, 2009. v. 1. p. 1-12.

Two other conference papers were published as a result of collaborations in related projects:

- MOTTA, R. ; **NOGUEIRA, B. M.** ; JORGE, A. M. ; LOPES, A. A. ; REZENDE, S. O. ; OLIVEIRA, M. C. . Comparing Relational and Non-relational Algorithms for Clustering Propositional Data. In: XXVIII Symposium on Applied Computing, 2013, Coimbra, Portugal. Proceedings of the 28th ACM Symposium on Applied Computing, 2013. p. 150-155.

- DOMINGUES, M. A. ; CHERMAN, E. A. ; **NOGUEIRA, B. M.** ; CONRADO, M. S. ; ROSSI, R. G. ; PADUA, R. ; MARCACINI, R. M. ; SOUZA, V. M. A. ; Batista, G. E. A. P. A. ; REZENDE, S. O. . A Comparative Study of Algorithms for Recommending Given Names. In: The Second International Conference on Informatics & Applications (ICIA2013), 2013, Lodz. Proceedings of The Second International Conference on Informatics & Applications (ICIA2013), 2013. v. 1. p. 66-71.

Moreover, another paper were submitted to journal and is in review process:

- **NOGUEIRA, B. M.** ; JORGE, A. M. ; REZENDE, S. O Efficient hierarchical confidence-based active approaches for semi-supervised clustering. Submitted to: Information Sciences, p. 1-46, 2013.

Finally, three journal papers are in final preparation and must be submitted in the near future:

- **NOGUEIRA, B. M.** ; JORGE, A. M. ; REZENDE, S. O. A survey on semi-supervised clustering. To be submitted to: ACM Computing Surveys, p. 1-27.

- **NOGUEIRA, B. M.** ; JORGE, A. M. ; REZENDE, S. O. A comparative study of semi-supervised hierarchical document clustering. Invited to be submitted to: Journal of Information and Data Management as an extended version of the paper (Nogueira et al., 2013).

- REZENDE, S. O. ; CONRADO, M. S. ; **NOGUEIRA, B. M.** ; ROSSI, R. G. ; SANTOS, F. F. ; MARCACINI, R. M.; NOGUEIRA, T. M. ; ALBUQUERQUE, D. ; MOURA, M.F. . FATHER: a Framework for Automatic Generating Topic Hierarchies. To be submitted to: ACM Transactions in Information Systems, p 1-24, 2013.

6.3 Limitations

Our proposal has some limitations that we intend to work on the near future. The first limitation is related to the cluster representation for the user interaction in the HCAC algorithm. In our experiments, the cluster of documents were represented with a list of descriptors obtained by the FScore measure, or the titles of the documents, according to the size of the cluster. We believe, however, that more expressive representations would improve the user comprehension of the cluster contents and lead the HCAC algorithm to achieve better clustering performance. Moreover, the real user interaction in applications involving other types of data were not investigated during this work.

The second limitation refers to the complexity of the HCAC-based algorithms. HCAC-based algorithms are based in distance matrices and have a space complexity of $O(n^2)$. This limits the maximum size of the datasets HCAC-based algorithms can deal with, forbidding their application in very large datasets.

As a third limitation we can cite the applicability of HCAC-based algorithms in binary datasets. In such datasets, the active learning approach of HCAC is not efficient and other semi-supervised approaches tend to present better performance.

The fourth limitation is related to the cluster labels selection in the Pattern Extraction step of the SMITH framework. We employed the FScore algorithm, which is fast and capable to retrieve a set of significant terms for a given dataset. However, this method does not consider the hierarchical structure of the clusters. If a term is significant for a parent node and for a children of this node, the descriptors selection based on the FScore measure would select the same descriptor for the two clusters. Thus, the exploration of the hierarchy is affected, since the user cannot distinguish the specialization of the knowledge throughout the hierarchy.

Another limitation is related to the decision of not adopting a pruning procedure in the cluster hierarchies obtained in the SMITH framework. Since our objective was to measure the impact of the semi-supervised clustering to the user, we opted by not pruning

the cluster hierarchies in order not to introduce a new bias in the process. However, the user tests indicated that a deep cluster hierarchy is not interesting and may lead the user to make incorrect choices while navigating through the hierarchy.

Finally, the experiments with the user carried out in this work have the limitation of employing few users. Obtaining volunteer users is not a trivial task, since they have to be available for a considerable amount of time and remain motivated to collaborate during the entire task. In this scenario, the conclusions obtained from the results are indicative, but not assertive.

6.4 Future work

As future work, we intend to investigate incremental and online approaches for the HCAC-based algorithms. Incremental clustering is an efficient way of organizing dynamic and large scale datasets and are one of the tendencies in knowledge management (Marcacini and Rezende, 2010a). The transformation of the HCAC-based algorithms into incremental clustering algorithms would bring the ability to deal with more dimensions. Moreover, as the knowledge about a domain is in constant evolution, it is expected that new documents describing this new knowledge emerge along the time. Thus, incremental clustering would help to incorporate this new knowledge in topic hierarchies avoiding to unnecessarily reprocess the existing structure.

Another interesting future investigation is on the propagation of the constraints inserted to HCAC-based algorithms to other elements. The current versions of HCAC and HCAC-LC allow the definition of constraints over a set of elements. The information inserted helps clustering only the elements affected by the constraints. However, it would be interesting to propagate the information inserted to other elements in the dataset. For example, if the user indicates that two clusters are better clustering options than the unsupervised cluster merge, this information would be used to recalculate the distance among the other elements in that region of the dataset.

We also intend to investigate the usage of different cluster labeling approaches. As described, the FScore measure may indicate cluster labels that are not adequate to the hierarchical structure of the topic hierarchies. Other cluster labeling algorithms like the RLUM algorithm (Moura and Rezende, 2010) should be tested in the context of the SMITH framework.

Another point to be investigated is related to the presentation of the queries to the user in both textual and non-textual datasets. We intend to expand the application of the HCAC-based algorithm to other scenarios, involving, for example, images and music. Thus, alternative representation of the clusters should be investigated.

Finally, we would investigate the construction of topic hierarchies considering more user interactions. Despite the results presented in this work indicate that HCAC-based

algorithms can improve domain representations in topic hierarchies according to the user needs, we believe that a small increment in the percentage of user interventions would bring significant improvements in the results. Moreover, it would be interesting to test the generation of topic hierarchies for document collections from different domains, involving different user profiles and different datasets.

Bibliography

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in page 13.

Alencar, A. B., de Oliveira, M. C. F., and Paulovich, F. V. (2012). Seeing beyond reading: a survey on visual text analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):476–492. Cited in pages 15 and 20.

Aliguliyev, R. M. (2009). Performance evaluation of density-based clustering methods. *Information Sciences*, 179(20):3583 – 3602. Cited in page 62.

Arampatzis, A., van der Weide, T., Koster, C., and van Bommel, P. (2000). Linguistically-motivated information retrieval. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York, Basel. Cited in page 10.

Ares, M. E., Parapar, J., and Barreiro, A. (2009). Avoiding bias in text clustering using constrained k-means and may-not-links. In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, pages 322–329, Berlin, Heidelberg. Springer-Verlag. Cited in page 32.

Bade, K., Hermkes, M., and Nürnberger, A. (2007). User oriented hierarchical information organization and retrieval. In *ECML '07: Proceedings of the 18th European Conference on Machine Learning*, pages 518–526, Berlin, Heidelberg. Springer-Verlag. Cited in pages 3, 44, and 53.

Bade, K. and Nurnberger, A. (2006). Personalized hierarchical clustering. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 181–187, Washington, DC, USA. IEEE Computer Society. Cited in pages 30 and 32.

Baghshah, M. S. and Shouraki, S. B. (2010). Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data. *Pattern Recognition*, 43(8):2982–2992. Cited in page 36.

Balcan, M.-F. and Blum, A. (2008). Clustering with interactive feedback. In *ALT '08: Proceedings of the 19th international conference on Algorithmic Learning Theory*, pages 316–328, Berlin, Heidelberg. Springer-Verlag. Cited in page 32.

Bar-Hillel, A., Hertz, T., Shental, N., and Weinshall, D. (2003). Learning distance functions using equivalence relations. In *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, pages 11–18, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in page 35.

Baraldi, A. and Blonda, P. (1999). A survey of fuzzy clustering algorithms for pattern recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(6):778–785. Cited in page 25.

Bast, H., Dupret, G., Majumdar, D., and Piwowarski, B. (2005). Discovering a term taxonomy from term similarities using principal component analysis. In *EWMF-KDO '05: Proceedings of the 2005 European Web Mining Forum*, pages 103–120, Porto, Portugal. Cited in page 19.

Basu, S., Banerjee, A., and Mooney, R. J. (2002). Semi-supervised clustering by seeding. In *ICML '02: Proceedings of the 19th International Conference on Machine Learning*, pages 27–34, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in pages 3, 24, 28, and 34.

Basu, S., Banerjee, A., and Mooney, R. J. (2004a). Active semi-supervision for pairwise constrained clustering. In *SDM '04: Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 333–344, Philadelphia, PA, USA. SIAM. Cited in pages 3, 41, and 52.

Basu, S., Bilenko, M., and Mooney, R. J. (2004b). A probabilistic framework for semi-supervised clustering. In *KDD '04: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68, New York, NY, USA. ACM. Cited in pages 34 and 36.

Beil, F., Ester, M., and Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 436–442. ACM. Cited in page 11.

Bekkerman, R. and Allan, J. (2004). Using bigrams in text categorization. Technical Report IR-408, Center of Intelligent Information Retrieval, UMass Amherst. Cited in page 10.

Ben Ahmed, E., Nabli, A., and Gargouri, F. (2013). Group extraction from professional social network using a new semi-supervised hierarchical clustering. *Knowledge and Information Systems*, pages 1–19. Cited in page 43.

Bensaid, A. M., Hall, L. O., Bezdek, J. C., and Clarke, L. P. (1996). Partially supervised clustering for image segmentation. *Pattern Recognition*, 29(5):859–871. Cited in pages 25 and 43.

Bilenko, M., Basu, S., and Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *ICML '04: Proceedings of the 21st International Conference on Machine learning*, pages 81–88, New York, NY, USA. ACM. Cited in pages 24, 29, 35, 37, 38, and 52.

Bilenko, M. and Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *KDD '03: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48, New York, NY, USA. ACM. Cited in page 34.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, New York, NY, USA. ACM. Cited in page 14.

Bock, H. H. (1996). Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28. Cited in page 25.

Böhm, C. and Plant, C. (2008). Hissclu: a hierarchical density-based method for semi-supervised clustering. In *EDBT '08: Proceedings of the 11th International Conference on Extending Database Technology*, pages 440–451, New York, NY, USA. ACM. Cited in pages 3 and 53.

Bouchachia, A. and Pedrycz, W. (2006). Data clustering with partial supervision. *Data Mining and Knowledge Discovery*, 12:47–78. Cited in page 25.

Cabré, M. T., Estopà, R., and Vivaldi, J. (2001). Automatic term detection: a review of current systems. In Bourigault, D., Jacquemin, C., and L'Homme, M.-C., editors, *Recent Advances in Computational Terminology*, pages 53–88, Amsterdam/Philadelphia. John Benjamins. Cited in page 10.

Cai, X. and Li, W. (2011). A spectral analysis approach to document summarization: Clustering and ranking sentences simultaneously. *Information Sciences*, 181(18):3816 – 3827. Cited in page 57.

Card, S., Mackinlay, J., and Schneiderman, B. (1999). Information visualization. In Card, S., Mackinlay, J., and Schneiderman, B., editors, *Readings in Information Visualization:*

Using Vision to Think, chapter 1, pages 1–34. Morgan Kaufmann Publishers, 1 edition. Cited in page 15.

Carvalho, V. O., Rezende, S. O., and Castro, M. (2007). An analytical evaluation of objective measures behavior for generalized association rules. In *CIDM '07: I IEEE Symposium on Computational Intelligence and Data Mining*, pages 43–50. IEEE. Cited in page 15.

Carvalho, V. R. and Cohen, W. W. (2006). Improving "email speech acts" analysis via n-gram selection. In *ACTS '06: Proceedings of the 2006 Workshop on Analyzing Conversations in Text and Speech*, pages 35–41, NJ, USA. Association for Computational Linguistics. Cited in page 11.

Chakrabarti, S. (2003). *Mining the Web: Discovering knowledge from hypertext data*. Morgan Kaufmann Publishers. Cited in pages 2 and 14.

Chandola, V. and Kumar, V. (2007). Summarization - compressing data into an informative representation. *Knowledge and Information Systems*, 12(3):355–378. Cited in page 14.

Chang, H. and Yeung, D.-Y. (2006). Locally linear metric adaptation with application to semi-supervised clustering and image retrieval. *Pattern Recognition*, 39(7):1253–1264. Cited in page 43.

Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, pages 288–296. Cited in page 20.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, USA. Cited in page 23.

Chu, H. (2003). *Information representation and retrieval in the digital age*. Information Today, Inc. Cited in page 19.

Chung, F.-L., Wang, S., Deng, Z., Shu, C., and Hu, D. (2006). Clustering analysis of gene expression data based on semi-supervised visual clustering algorithm. *Soft Computing*, 10(11):981–993. Cited in page 43.

Cohn, D., Caruana, R., and Mccallum, A. (2003). Semi-supervised clustering with user feedback - technical report tr2003-1892. Technical report, Cornell University. Cited in pages 24, 29, 31, and 34.

Conrado, M. S., Gutiérrez, V. A. L., and Rezende, S. O. (2012). Evaluation of normalization techniques in text classification for portuguese. In Murgante, B., Gervasi, O.,

Misra, S., Nedjah, N., Rocha, A. M. A. C., Taniar, D., and Apduhan, B. O., editors, *ICCSA '12: Proceedings of the 12th International Conference on Computational Science and Applications*, Lecture Notes in Computer Science, pages 618–630, Salvador, BA, Brasil. Springer. Cited in pages 11, 17, and 82.

Conrado, M. S. and Rezende, S. O. (2008). Avaliando a geração de termos a partir de coleções textuais. In *WTDIA '08: Anais do IV Workshop de Teses e Dissertações em Inteligência Artificial - SBIA '08: XIX Simpósio Brasileiro de Inteligência Artificial*, pages 1–10. São Carlos : ICMC/USP. Cited in pages 11 and 12.

Covoes, T. F., Hruschka, E. R., and Ghosh, J. (2013). A study of k-means-based algorithms for constrained clustering. *Intelligent Data Analysis*, 17. Cited in pages 38, 39, and 48.

Daniels, K. and Giraud-Carrier, C. (2006). Learning the threshold in hierarchical agglomerative clustering. In *ICMLA '06: Proceedings of the 5th International Conference on Machine Learning and Applications*, pages 270–278, Washington, DC, USA. IEEE. Cited in pages 3 and 53.

Dasgupta, S. and Ng, V. (2010). Which clustering do you want? inducing your ideal clustering with minimal feedback. *Journal of Artificial Intelligence Research*, 39:581–632. Cited in pages 3, 24, and 51.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156. Cited in page 11.

Davidson, I. and Basu, S. (2007). A survey of clustering with instance level constraints. *ACM Transactions on Knowledge Discovery from Data*, pages 1–41. Cited in pages 24 and 52.

Davidson, I. and Ravi, S. S. (2005). Clustering with constraints: Feasibility issues and the k-means algorithm. In *SDM '05: Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 138–149, Philadelphia, PA, USA. SIAM. Cited in pages 30, 37, 38, and 52.

Davidson, I. and Ravi, S. S. (2009). Using instance-level constraints in agglomerative hierarchical clustering: theoretical and empirical results. *Data Mining and Knowledge Discovery*, 18(2):257–282. Cited in pages 3, 24, 34, 46, 53, 57, and 61.

Davidson, I., Wagstaff, K. L., and Basu, S. (2006). Measuring constraint-set utility for partitional clustering algorithms. In *PKDD '06: Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 115–126. Springer. Cited in pages 39 and 44.

de Souza, K. X. S., Davis, J., Evangelista, S. R. M., Souza, M. I. F., Santos, A. D., and Moura, M. F. (2005). The evolution of knowledge representation within Embrapat's information agency. In *EFITA / WCCA '05: Proceedings of 5th Conference of the European Federation for Information Technology in Agriculture, Food and Environment, 3rd World Congress on Computers in Agriculture and Natural Recources*, pages 464–470, Vila Real, Portugal. Universidade Trás-os-Montes e Alto Douro. Cited in page 20.

Domeniconi, C., Peng, J., and Yan, B. (2010). Composite kernels for semi-supervised clustering. *Knowledge and Information Systems*, 24(1):1–18. Cited in pages 24 and 29.

Duan, C., Cleland-Huang, J., and Mobasher, B. (2008). A consensus based approach to constrained clustering of software requirements. In *CIKM '08: Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 1073–1082, New York, NY, USA. ACM. Cited in page 44.

Dubey, A., Bhattacharya, I., and Godbole, S. (2010). A cluster-level semi-supervision model for interactive clustering. In *ECML PKDD'10: Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, pages 409–424, Berlin, Heidelberg. Springer-Verlag. Cited in pages 31, 32, and 33.

Dupret, G. and Piwowarski, B. (2005). Deducing a term taxonomy from term similarities. In *KDO '05: Proceedings of Second International Workshop on Knowledge Discovery and Ontologies*, pages 11–22, Berlin, Heidelberg. Springer-Verlag. Cited in pages 2 and 16.

Ebecken, N. F. F., Lopes, M. C. S., and de Aragão Costa, M. C. (2003). Mineração de textos. In Rezende, S. O., editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, chapter 13, pages 337–370. Manole, 1 edition. Cited in page 7.

Eick, C. F., Zeidat, N., and Zhao, Z. (2004). Supervised clustering: Algorithms and benefits. In *ICTAI '04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 774–776, Washington, DC, USA. IEEE Computer Society. Cited in page 33.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD '96: Proceedings of the II International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press. Cited in page 42.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *KDD '96: Proceedings of Knowledge Discovery and Data Mining Conference*, pages 82–88, EUA. AAAI Press. Cited in pages 8 and 9.

Feldman, R. and Sanger, J. (2007). *The Text Mining Hand Book - Advanced Approaches in Analysing Unstructured Data*. Cambridge University Press. Cited in page 19.

Filipovych, R., Resnick, S. M., and Davatzikos, C. (2011). Semi-supervised cluster analysis of imaging data. *NeuroImage*, 54(3):2185–2197. Cited in page 43.

Finley, T. and Joachims, T. (2005). Supervised clustering with support vector machines. In *ICML '05: Proceedings of the 22nd International Conference on Machine learning*, pages 217–224, New York, NY, USA. ACM. Cited in page 33.

Forestier, G., Gançarski, P., and Wemmert, C. (2010). Collaborative clustering with background knowledge. *Data and Knowledge Engineering*, 69(2):211–228. Cited in page 46.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305. Cited in pages 11, 12, and 84.

Fred, A. L. N. and Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850. Cited in page 46.

Fung, B. C. M., Wang, K., and Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In *SDM '03: Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 59–70. Cited in page 11.

Gantz, J. F. and Reinsel, D. (2009). As the economy contracts, the digital universe expands. *External Publication of IDC (Analyse the Future) Information and Data*, pages 1–10. Cited in page 1.

Gantz, J. F. and Reinsel, D. (2012). The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. *External Publication of IDC (Analyse the Future) Information and Data*, pages 1–16. Cited in page 1.

Garner, W. R. (1962). *Uncertainty and structure as psychological concepts*. Wiley. Cited in pages 37 and 58.

Garshol, L. M. (2004). Metadata? thesauri? taxonomies? topic maps! *Journal of Information Science*, 30(4):378–391. Cited in page 15.

Gath, I. and Gev, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions Pattern Analysis Machine Intelligence*, 11(7):773–780. Cited in page 41.

Gil-García, R. and Pons-Porrata, A. (2008). Hierarchical star clustering algorithm for dynamic document collections. In *CIARP '08: Proceedings of the 13th Iberoamerican congress on Pattern Recognition*, pages 187–194, Berlin, Heidelberg. Springer-Verlag. Cited in pages 2 and 16.

Gonzalez, M. A. I., de Lima, V. L. S., and de Lima, J. V. (2006). Tools for nominalization: An alternative for lexical normalization. In *PROPOR '06: Proceedings of the 7th Workshop on Computational Processing of Written and Spoken Portuguese*, volume 3960, pages 100–109. Springer Berlin / Heidelberg. Cited in page 10.

Grira, N., Crucianu, M., and Boujemaa, N. (2008). Active semi-supervised fuzzy clustering. *Pattern Recognition*, 41(5):1851–1861. Cited in pages 41 and 43.

Hamasuna, Y., Endo, Y., and Miyamoto, S. (2011). Semi-supervised agglomerative hierarchical clustering with ward method using clusterwise tolerance. In *MDAI '11: Proceedings of the 8th International Conference on Modeling Decisions for Artificial Intelligence*, pages 103–113, Berlin, Heidelberg. Springer-Verlag. Cited in page 24.

Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86. Cited in page 13.

He, Q., Chang, K., Lim, E.-P., and Banerjee, A. (2010). Keep it simple with time: A re-examination of probabilistic topic detection models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1 – 14. Cited in page 2.

Hearst, M. A. (1999). Untangling text data mining. In *ACL '99: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10, Stroudsburg, PA, USA. Association for Computational Linguistics. Cited in page 7.

Hertz, T., Bar-Hillel, A., and Weinshall, D. (2004). Boosting margin based distance functions for clustering. In *ICML '04: Proceedings of the 21st International Conference on Machine Learning*, pages 50–, New York, NY, USA. ACM. Cited in page 35.

Hofmann, T. and Buhmann, J. M. (1998). Active data clustering. *Advances in Neural Information Processing Systems*, pages 528–534. Cited in page 40.

Hu, Y., Milios, E. E., and Blustein, J. (2012). Semi-supervised document clustering with dual supervision through seeding. In *SAC '12: Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 144–151, New York, NY, USA. ACM. Cited in page 29.

Huang, A., Milne, D., Frank, E., and Witten, I. H. (2008). Clustering documents with active learning using wikipedia. In *ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining*, pages 839–844. Cited in page 44.

Huang, R. and Lam, W. (2007). Semi-supervised document clustering via active learning with pairwise constraints. In *ICDM '07: Proceedings of the 7th IEEE International Conference on Data Mining*, pages 517–522, Washington, DC, USA. IEEE Computer Society. Cited in pages 24 and 29.

Huang, R. and Lam, W. (2009). An active learning framework for semi-supervised document clustering with language modeling. *Data and Knowledge Engineering*, 68(1):49–67. Cited in pages 2, 3, 14, 24, 25, 33, 41, and 44.

Huang, R., Zhang, Z., and Lam, W. (2006). Text clustering with limited user feedback under local metric learning. In *AIRS '06: Proceedings of the 3rd Asia Information Retrieval Symposium*, Lecture Notes in Computer Science, pages 132–144, New York, NY, USA. Springer. Cited in page 31.

Huang, Y. and Mitchell, T. M. (2006). Text clustering with extended user feedback. In *SIGIR '06: Proceedings of the 29th ACM Conference on Research and Development in Information Retrieval*, pages 413–420, New York, NY, USA. ACM. Cited in pages 31, 32, 33, and 44.

Huang, Y. and Mitchell, T. M. (2008). Exploring hierarchical user feedback in email clustering. In *EMAIL '08: Proceedings of the Workshop on Enhanced Messaging - AAAI 2008*, pages 36–41. AAAI Press. Cited in pages 31, 32, 33, and 44.

Inselberg, A. (2009). *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag, Secaucus, USA. Cited in pages 57 and 76.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666. Cited in pages 14, 28, and 47.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. Cited in pages 2, 25, 40, and 61.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323. Cited in page 13.

Ji, X. and Xu, W. (2006). Document clustering with prior knowledge. In *SIGIR '06: Proceedings of the 29th Annual International ACM Conference on Research and Development in Information Retrieval*, pages 405–412, New York, NY, USA. ACM. Cited in page 44.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *ICML '99: Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in page 14.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, EUA. Cited in page 11.

Jorge, M. L. R. C. and Pardo, T. A. S. (2010). Experiments with CST-Based multi-document summarization. In *Proceedings of TextGraphs-5: Workshop on Graph-based Methods for Natural Language Processing*, pages 74–82, Uppsala, Sweden. Association for Computational Linguistics. Cited in page 14.

Kantardzic, M. (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons, Piscataway, NJ, EUA. Cited in page 13.

Kashyap, V., Ramakrishnan, C., Thomas, C., and Sheth, A. (2004). Taxaminer: An experimentation framework for automated taxonomy bootstrapping. Technical report, computer Science Department - University of Georgia. Available at: http://lsdis.cs.uga.edu/ cthomas/resources/taxaminer.pdf. Cited in page 1.

Kestler, H. A., Kraus, J. M., Palm, G., and Schwenker, F. (2006). On the effects of constraints in semi-supervised hierarchical clustering. In *Artificial Neural Networks in Pattern Recognition*, pages 57–66. Springer-Verlag. Cited in pages 3 and 53.

Kim, H.-J. and Lee, S.-G. (2000). A semi-supervised document clustering technique for information organization. In *CIKM '00: Proceedings of the 9th International Conference on Information and Knowledge Management*, pages 30–37, New York, NY, USA. ACM. Cited in page 43.

Kim, H.-j. and Lee, S.-g. (2002). An effective document clustering method using user-adaptable distance metrics. In *SAC '02: Proceedings of the 9th ACM Symposium on Applied Computing*, pages 16–20, New York, NY, USA. ACM. Cited in page 35.

King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101. Cited in page 2.

Klein, D., Kamvar, S. D., and Manning, C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML '02: Proceedings of the 19th International Conference on Machine Learning*, pages 307–314, San Francisco, CA, USA. Morgan Kaufmann Publishers. Cited in pages 3, 24, 30, 32, 34, 40, 45, 53, and 61.

Koster, C. and Seutter, M. (2003). Taming wild phrases. In *ECIR '03: Proceedings of the 2003 European Conference on Information Retrieval*, pages 161–176. Cited in page 11.

Krovetz, R. (1993). Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–202, New York, NY, USA. ACM. Cited in page 10.

Kulis, B., Basu, S., Dhillon, I., and Mooney, R. (2005). Semi-supervised graph clustering: a kernel approach. In *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pages 457–464, New York, NY, USA. ACM. Cited in page 36.

Kulis, B., Sustik, M. A., and Dhillon, I. S. (2009). Low-rank kernel learning with bregman matrix divergences. *Journal of Machine Learning Research*, 10:341–376. Cited in pages 35 and 36.

Kumar, N., Kummamuru, K., and Paranjpe, D. (2005). Semi-supervised clustering with metric learning using relative comparisons. In *ICDM '05: Proceedings of the 5th IEEE International Conference on Data Mining*, pages 693–696, Washington, DC, USA. IEEE. Cited in pages 30 and 32.

Lai, H. P., Visani, M., Boucher, A., and Ogier, J.-M. (2013). A new interactive semi-supervised clustering model for large image database indexing. *Pattern Recognition Letters*, (0):1 – 13. Cited in page 43.

Lamping, J., Rao, R., and Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 401–408, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. Cited in pages 15 and 20.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284. Cited in page 11.

Larsen, B. and Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16–22, New York, NY, USA. ACM. Cited in pages 58 and 62.

Law, M. H., Topchy, A., and Jain, A. K. (2004). Clustering with soft and group constraints. In Fred, A., Caelli, T., Duin, R., Campilho, A., and de Ridder, D., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 3138 of *Lecture Notes in Computer Science*, pages 662–670. Springer Berlin Heidelberg. Cited in page 37.

Lawrie, D. and Croft, W. B. (2000). Discovering and comparing topic hierarchies. In *RIAO '00: Proceedings of the 6th Recherche d'Informations Assistee par Ordinateur*, pages 314–330, França. CID. Cited in page 16.

Lawrie, D., Croft, W. B., and Rosenberg, A. (2001). Finding topic words for hierarchical summarization. In *SIGIR '01: Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval*, pages 349–357, New York, NY, EUA. ACM. Cited in pages 1 and 16.

Li, Z., Liu, J., and Tang, X. (2008). Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pages 576–583, New York, NY, USA. ACM. Cited in page 30.

Lin, C.-R. and Chen, M.-S. (2005). Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):145–159. Cited in page 25.

Liu, L., Kang, J., Yu, J., and Wang, Z. (2005). A comparative study on unsupervised feature selection methods for text clustering. In *NLP-KE '05: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pages 597–601. Cited in page 12.

Liu, Y., Jin, R., and Jain, A. K. (2007). Boostcluster: boosting clustering by pairwise constraints. In *KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 450–459, New York, NY, USA. ACM. Cited in page 35.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal os Research and Development*, 2(2):159–165. Cited in pages 12 and 82.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proceedings of the V Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press. Cited in pages 2 and 28.

Maggini, M., Melacci, S., and Sarti, L. (2012). Learning from pairwise constraints by similarity neural networks. *Neural Networks*, 26:141–158. Cited in pages 34 and 36.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). Language models for information retrieval. In *An Introduction to Information Retrieval*, chapter 12. Cambridge University Press. Cited in page 10.

Manning, C. D. and Schtze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, EUA. Cited in pages 10 and 11.

Maraziotis, I. A. (2012). A semi-supervised fuzzy clustering algorithm applied to gene expression data. *Pattern Recognition*, 45(1):637–648. Cited in page 43.

Marcacini, R. M. (2008). Um ambiente interativo para análise visual de agrupamentos hierárquicos. Monografia conclusão de curso de graduação, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. Cited in pages 15 and 20.

Marcacini, R. M., Correa, G. N., and Rezende, S. O. (2012). An active learning approach to frequent itemset-based text clustering. In *ICPR' 12: Proceedings of the 21st International Conference on Pattern Recognition*, pages 3529 –3532. Cited in pages 42 and 44.

Marcacini, R. M. and Rezende, S. O. (2010a). Incremental construction of topic hierarchies using hierarchical term clustering. In *SEKE '10: Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering*, pages 553–558. Cited in pages 47 and 101.

Marcacini, R. M. and Rezende, S. O. (2010b). Torch: a tool for building topic hierarchies from growing text collections. In *WFA'2010: Proceedings of the 10th Worshop on Tools and Applications – Webmedia'2010: Brazilian Symposium on Multimedia and the Web*, pages 133–135. Cited in pages 85, 86, and 87.

McNicholas, P. D. and Subedi, S. (2012). Clustering gene expression time course data using mixtures of multivariate t-distributions. *Journal of Statistical Planning and Inference*, 142(5):1114 – 1127. Cited in page 43.

Miiller, A. and Dorre, J. (1999). The taxgen framework: Automating the generation of a taxonomy for a large document collection. In *HICSS '99: Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*, page 2034, Washington, DC, USA. IEEE Computer Society. Cited in page 16.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Higher Education. Cited in page 13.

Miyamoto, S. and Terami, A. (2011). Constrained agglomerative hierarchical clustering algorithms with penalties. In *FUZZ'11: Proceedings of the 2011 IEEE International Conference on Fuzzy Systems*, pages 422 –427. Cited in pages 3 and 53.

Motta, R., Nogueira, B. M., Jorge, A. M., Lopes, A. A., and Rezende, S. O. (2013). Comparing relational and non-relational algorithms for clustering propositional data. In *SAC '13: Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 150–155, New York, NY, USA. ACM. Cited in page 25.

Moura, M. F. (2009). *Contribuições para a construção de taxonomias de tópicos em domínios restritos utilizando aprendizado estatístico*. PhD thesis, Instituto de Ciências Matemáticas e de Computação (ICMC) - USP, São Carlos. Cited in page 9.

Moura, M. F., Marcacini, R. M., Nogueira, B. M., da Silva Conrado, M., and Rezende, S. O. (2008a). A proposal for building domain topic taxonomies. In *WTI' 08: Proceedings of the I Workshop on Web and Text Intelligence - SBIA'08: XIX Brazilian Symposium on Artificial Intelligence*, pages 83–84, São Calos, SP, Brasil. ICMC/USP. Cited in pages 2, 4, 7, 16, 17, 79, 84, 85, and 94.

Moura, M. F., Marcacini, R. M., and Rezende, S. O. (2008b). Easily labelling hierarchical document clusters. In *WAAMD '08: Anais do 4th Workshop em Algoritmos e Aplicações de Mineração de Dados, SBBD '08: 23rd Simpósio Brasileiro de Banco de Dados*, pages 37–45. Porto Alegre: SBC. Cited in page 20.

Moura, M. F., Nogueira, B. M., Conrado, M. S., dos Santos, F. F., and Rezende, S. O. (2008c). Making good choices of non-redundant n-gramwords. In Library, I. X. D., editor, *DMAI '08: Proceedings of 1st International Workshop on Data Mining and Artificial Intelligence - ICCIT '08: 11th IEEE International Conference on Computer and Information Technology*, pages 64–71. Cited in page 13.

Moura, M. F. and Rezende, S. O. (2010). A simple method for labeling hierarchical document clusters. In *IAI'10: Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Applications*, pages 363–371, Anaheim, Calgary, Zurich : Acta Press, 2010. Cited in pages 19, 84, 94, and 101.

Neto, J. L., Santos, A. D., Kaestner, C. A. A., and Freitas, A. A. (2000). Document clustering and text summarization. In Company, L. T. P. A., editor, *PADD'00: Proceedings of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining*, pages 41–55. Cited in page 1.

Nogueira, B. M. (2009). Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos. Master's thesis, Instituto de Ciências Matemáticas e de Computação (ICMC) - USP, São Carlos. Cited in page 12.

Nogueira, B. M., Jorge, A. M., and Rezende, S. O. (2012a). HCAC: Semi-supervised hierarchical clustering using confidence-based active learning. In Ganascia, J.-G., Lenca, P., and Petit, J.-M., editors, *DS '12: Proceedings of the 15th International Conference on Discovery Science*, volume 7569 of *Lecture Notes in Computer Science*, pages 139–153. Springer Berlin Heidelberg. Cited in pages 68, 75, 97, and 98.

Nogueira, B. M., Jorge, A. M., and Rezende, S. O. (2012b). Hierarchical confidence-based active clustering. In *SAC'12: Proceedings of the 27th ACM Symposium on Applied Computing*, pages 535–536, New York, USA. ACM. Cited in pages 54, 55, 57, and 98.

Nogueira, B. M., Jorge, A. M., and Rezende, S. O. (2013). On the comparison of semi-supervised hierarchical clustering algorithms in text mining tasks. In *KDMILE'13:*

Proceedings of the 1st Symposium on Knowledge Discovery, Mining and Learning, pages 1–8, São Carlos, SP, Brazil. SBC. Cited in pages 98 and 100.

Nogueira, B. M., Moura, M. F., Conrado, M. S., and Rezende, S. O. (2008a). Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos. In *WTI' 08: Proceedings of the 1st Workshop on Web and Text Intelligence - SBIA'08: 19th Brazilian Symposium on Artificial Intelligence*, pages 59–66, São Calos, SP, Brasil. ICMC/USP. Cited in pages 12 and 18.

Nogueira, B. M., Moura, M. F., Conrado, M. S., Rossi, R. G., Marcacini, R. M., and Rezende, S. O. (2008b). Winning some of the document preprocessing challenges in a text mining process. In do IV Workshop em Algoritmos e Aplicações de Mineração de Dados, A., editor, *WAAMD '08: Anais do 4o Workshop em Algoritmos e Aplicações de Mineração de Dados (WAAMD) - XXIII Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 1–9, Campinas, SP. Cited in page 82.

Nogueira, B. M. and Rezende, S. O. (2009). Dois novos métodos para seleção não-supervisionada de atributos em mineração de textos. In *CLEI '09: Anais da 35a Conferencia Latinoamericana de Informática*, pages 1–10, Pelotas, RS, Brazil. Publicado em CD-ROM. Cited in pages 12 and 18.

Paukkeri, M.-S., García-Plaza, A. P., Fresno, V., Unanue, R. M., and Honkela, T. (2012). Learning a taxonomy from a set of text documents. *Applied Soft Computing*, 12(3):1138–1148. Cited in pages 2 and 4.

Pedrycz, W. (1985). Algorithms of fuzzy clustering with partial supervision. *Pattern Recognition Letters*, 3(1):13 – 20. Cited in page 25.

Pedrycz, W. (2004). Fuzzy clustering with a knowledge-based guidance. *Pattern Recognition Letters*, 25(4):469–480. Cited in page 25.

Pedrycz, W. and Waletzky, J. (1997). Fuzzy clustering with partial supervision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 27(5):787–795. Cited in page 25.

Pelleg, D. and Baras, D. (2007). K-means with large and noisy constraint sets. In *ECML '07: Proceedings of the 18th European Conference on Machine Learning*, pages 674–682, Berlin, Heidelberg. Springer-Verlag. Cited in pages 29, 37, 39, 47, and 48.

Peltonen, J., Sinkhonen, J., and Kaski, S. (2002). Discriminative clustering of text documents. In *ICONIP '02: Proceedings of IEEE 9th International Conference on Neural Information Processing*, volume 4, pages 1956–1960. Cited in page 14.

Pham, P., Deschacht, K., and Moens, M.-F. (2008). Document clustering with user feedback. In *DIR '08: Proceedings of the 8th Dutch-Belgian Information Retrieval Workshop*, pages 73–80. Maastricht University. Cited in pages 32 and 33.

Pons-Porrata, A., Berlanga-Llavori, R., and Ruiz-Shulcloper, J. (2007). Topic discovery based on text mining techniques. *Information Processing and Management: an International Journal*, 43(3):752–768. Cited in page 2.

Porter, M. (1980). An algorithm for suffixing stripping. *Program*, 14(3):130–137. Cited in pages 82 and 86.

Prati, R. C., Geromini, M. R., and Monard, M. C. (2003). An integrated environment for data mining. In *LAPTEC '03: Proceedings of 4th Congress of Logic Applied to Technology*, volume 2, pages 55–62, Marília - SP. Pléiade. Cited in page 86.

Punera, K., Rajan, S., and Ghosh, J. (2005). Automatically learning document taxonomies for hierarchical classification. In *WWW '05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, pages 1010–1011, New York, NY, USA. ACM. Cited in page 2.

Rezende, S. O., Pugliesi, J. B., Melanda, E. A., and Paula, M. F. (2003). Mineração de dados. In Rezende, S. O., editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, chapter 12, pages 307–335. Manole, 1 edition. Cited in page 8.

Rossi, R., de Paulo Faleiros, T., de Andrade Lopes, A., and Rezende, S. (2012). Inductive model generation for text categorization using a bipartite heterogeneous network. In *ICDM '12: Proceedings of the 12nd International Conference on Data Mining*, pages 1086–1091. IEEE, IEEE. Cited in page 13.

Sahoo, N., Callan, J., Krishnan, R., Duncan, G., and Padman, R. (2006). Incremental hierarchical clustering of text documents. In *CIKM '06: Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 357–366, NY, USA. ACM. Cited in page 47.

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing, MA, USA. Cited in pages 10 and 81.

Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, EUA. Cited in pages 11, 12, and 82.

Santos, F. F., Carvalho, V. O., and Rezende, S. O. (2010). Selecting candidate labels for hierarchical document clusters using association rules. In *MICAI '10: Proceedings*

of the 9th Mexican International Conference on Artificial Intelligence, pages 163–176, Berlin, Heidelberg. Springer-Verlag. Cited in page 19.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47. Cited in page 13.

Seeger, M. (2002). Learning with labeled and unlabeled data. Technical report, University of Edinburgh. Cited in page 23.

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. Cited in page 40.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 4:379–423. Cited in pages 56 and 61.

Shannon, C. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Reviews*, 5:3–55. Cited in page 56.

Shearer, C. (2000). The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4):13–22. Cited in page 8.

Shental, N., Bar-Hillel, A., Hertz, T., and Weinshall, D. (2004). Computing gaussian mixture models with em using equivalence constraints. *Advances in neural information processing systems*, 16(8):465–472. Cited in page 29.

Silberschatz, A. and Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery. In *KDD '95: Proceedings of Knowledge Discovery and Data Mining Conference*, pages 275–281. Cited in page 15.

Slonim, N. and Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–215, New York, NY, USA. ACM. Cited in page 12.

Sneath, P. H. A. and Sokal, R. R. (1973). *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W H Freeman, San Francisco, EUA. Cited in page 2.

Soares, M. V. B., Prati, R. C., and Monard, M. C. (2008). Pretext ii: Descrição da reestruturacão da ferramenta de pré-processamento de textos. Technical Report 333, ICMC-USP, São Carlos - SP. Cited in page 86.

Sparck-Jones, K. and Willett, P., editors (1997). *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., CA, USA. Cited in page 11.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617. Cited in page 46.

Sublemontier, J.-H., Martin, L., Cleuziou, G., and Exbrayat, M. (2011). Integrating pairwise constraints into clustering algorithms: Optimization-based approaches. In *ICDMW '11: Proceedings of the IEEE 11th International Conference on Data Mining Workshops*, pages 272–279, Washington, DC, USA. IEEE Computer Society. Cited in page 35.

Talavera, L. and Béjar, J. (1999). Integrating declarative knowledge in hierarchical clustering tasks. In *IDA '99: Proceedings of the 3rd International Symposium on Advances in Intelligent Data Analysis*, pages 211–222, London, UK. Springer-Verlag. Cited in pages 3, 24, and 53.

Tan, C.-M., Wang, Y.-F., and Lee, C.-D. (2002). The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546. Cited in page 12.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. Cited in page 18.

Tang, L., Liu, H., Zhang, J., Agarwal, N., and Salerno, J. J. (2008). Topic taxonomy adaptation for group profiling. *ACM Transactions on Knowlowledge Discovery from Data*, 1(4):1–28. Cited in pages 2 and 16.

Tesar, R., Strnad, V., Jezek, K., and Poesio, M. (2006). Extending the single words-based document model: a comparison of bigrams and 2-itemsets. In *DOCENG '06: Proceedings of the 6th Symposium on Document Engineering*, pages 138–146. Cited in page 11.

Topchy, A., Jain, A. K., and Punch, W. (2003). Combining multiple weak clusterings. In *ICDM '03: Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 331–, Washington, DC, USA. IEEE Computer Society. Cited in page 46.

Treeratpituk, P. and Callan, J. (2006). Automatically labeling hierarchical clusters. In *DGO '06: Proceedings of the 2006 International Conference on Digital Government Research*, pages 167–176, New York, NY, USA. ACM. Cited in page 20.

Vu, V.-V., Labroche, N., and Bouchon-Meunier, B. (2010). Boosting clustering by active constraint selection. In *ECAI '10: Proceeding of the 19th European Conference on Artificial Intelligence*, pages 297–302, Amsterdam, The Netherlands. IOS Press. Cited in page 53.

Vu, V.-V., Labroche, N., and Bouchon-Meunier, B. (2012). Improving constrained clustering with active query selection. *Pattern Recognition*, 45(4):1749 – 1758. Cited in pages 34 and 52.

Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. In *ICML '00: Proceedings of the 17th International Conference on Machine Learning*, pages 1103–1110, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in pages 3, 24, 29, 32, 33, 49, and 61.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the 18th International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in pages 29 and 37.

Wagstaff, K. L. (2006). Value, cost, and sharing: open issues in constrained clustering. In *KDID'06: Proceedings of the 5th International Conference on Knowledge Discovery in Inductive Databases*, pages 1–10, Berlin, Heidelberg. Springer-Verlag. Cited in page 44.

Wang, C., Chen, W., Yin, P., and Wang, J. (2007). Semi-supervised clustering using incomplete prior knowledge. In *ICCS '07: Proceedings of the 7th International Conference on Computational Science, Part I*, pages 192–195, Berlin, Heidelberg. Springer-Verlag. Cited in page 29.

Wang, H., Nie, R., Liu, X., and Li, T. (2012a). Constraint projections for semi-supervised affinity propagation. *Knowledge-Based Systems*, 36:315–321. Cited in pages 35 and 36.

Wang, X., Qian, B., and Davidson, I. (2012b). Labels vs. pairwise constraints: A unified view of label propagation and constrained spectral clustering. In *ICDM '12: Proceedings of the IEEE 12th International Conference on Data Mining*, pages 1146 –1151. Cited in page 24.

Weiss, S. M., Indurkhya, N., Zhang, T., and Damerau, F. J. (2005). *Text Mining - Predictive Methods for Analizing Unstructured Information*. Springer Science+Business Media, Inc. Cited in pages 7, 10, and 19.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83. Cited in page 62.

Wyse, N., Dubes, R., and Jain, A. (1980). A critical evaluation of intrinsic dimensionality algorithms. In Gelsema, E. and Kanal, L., editors, *Pattern Recognition in Practice*, pages 415–425. North-Holland. Cited in page 11.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2003). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information*

Processing Systems 15, volume 15, pages 505–512, Cambridge, MA. MIT Press. Cited in pages 24, 29, and 34.

Xu, R. and Wunsch, D., I. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645 –678. Cited in page 25.

Yang, M.-S. (1993). A survey of fuzzy clustering. *Mathematical and Computer modelling*, 18(11):1–16. Cited in page 25.

Yin, X., Chen, S., Hu, E., and Zhang, D. (2010). Semi-supervised clustering with metric learning: An adaptive kernel method. *Pattern Recognition*, 43(4):1320–1333. Cited in pages 35 and 36.

Yoshida, T. (2012). Influence of erroneous pairwise constraints in semi-supervised clustering. In *AMT '12: Proceedings of the 8th International Conference on Active Media Technology*, pages 43–52, Berlin, Heidelberg. Springer-Verlag. Cited in page 48.

Zamir, O., Etzioni, O., Madani, O., and Karp, R. M. (1997). Fast and intuitive clustering of web documents. In *KDD '97: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 287–290, Menlo Park, CA, USA. AAAI Press. Cited in page 47.

Zavitsanos, E., Paliouras, G., and Vouros, G. A. (2011). Non-parametric estimation of topic hierarchies from texts with hierarchical dirichlet processes. *Journal of Machine Learning Research*, 12:2749–2775. Cited in page 2.

Zeng, H. and Cheung, Y.-M. (2012). Semi-supervised maximum margin clustering with pairwise constraints. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):926–939. Cited in page 38.

Zeng, S., Tong, X., Sang, N., and Huang, R. (2012). A study on semi-supervised fcm algorithm. *Knowledge and Information Systems*, pages 1–28. Cited in page 25.

Zhang, Z., Kwok, J. T., and Yeung, D.-Y. (2003). Parametric distance metric learning with label information. In *IJCAI'03: Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1450–1452, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. Cited in page 35.

Zhao, W., He, Q., Ma, H., and Shi, Z. (2009). Active learning of instance-level constraints for semi-supervised document clustering. In *WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 264–268, Washington, DC, USA. IEEE Computer Society. Cited in page 42.

Zhao, W., He, Q., Ma, H., and Shi, Z. (2012). Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowledge and Information Systems*, 30(3):569–587. Cited in page 44.

Zhao, Y., Karypis, G., and Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168. Cited in pages 2 and 53.

Zheng, L. and Li, T. (2011). Semi-supervised hierarchical clustering. In *ICDM '11: Proceedings of the IEEE 11th International Conference on Data Mining*, pages 982 –991. Cited in pages 3, 24, 30, 32, and 53.

Zhong, C., Miao, D., and Fränti, P. (2011). Minimum spanning tree based split-and-merge: A hierarchical clustering method. *Information Sciences*, 181(16):3397–3410. Cited in page 25.

Zhong, S. (2006). Semi-supervised model-based document clustering: A comparative study. *Machine Learning*, 65(1):3–29. Cited in pages 28 and 44.

Zhu, D., Hero, A. O., Cheng, H., Khanna, R., and Swaroop, A. (2005). Network constrained clustering for gene microarray data. *Bioinformatics*, 21(21):4014–4020. Cited in page 43.

Zhu, X. (2005). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison. Cited in pages 14 and 23.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley. Cited in page 12.

Zoller, T. and Buhmann, J. (2000). Active learning for hierarchical pairwise data clustering. In *ICPR' 00: Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 186–189 vol.2. Cited in page 40.