Universidad Nacional de Educación a Distancia
Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática

AN IMPROVED FUZZY SYSTEM FOR REPRESENTING WEB PAGES IN CLUSTERING TASKS

PhD THESIS

Alberto Pérez García-Plaza
MSc in Computer Science

2012

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática

AN IMPROVED FUZZY SYSTEM FOR REPRESENTING WEB PAGES IN CLUSTERING TASKS

Alberto Pérez García-Plaza
MSc in Computer Science, Universidad Rey Juan Carlos

Advisors:

Víctor Fresno Fernández

Assistant Professor in the Lenguajes y Sistemas Informáticos Department at Universidad Nacional de Educación a Distancia

Raquel Martínez Unanue

Associate Professor in the Lenguajes y Sistemas Informáticos Department at Universidad Nacional de Educación a Distancia

"I have seen and done so much that 10 persons could split it and still be happy, so if you ask me if I am happy... Hell Yeah!."

—Tom S. Englund.

Acknowledgments

Institutional Acknowledgements

The research presented in this work has been in part funded by the Regional Government of Madrid under the Research Networks MAVIR (S-0505/TIC-0267) and MA2VICMR (S-2009/TIC-1542), the Regional Ministry of Education of the Community of Madrid, and the Spanish Ministry of Science and Innovation projects QEAVis-Catiex (TIN2007-67581-C02-01) and Holopedia (TIN2010-21128-C02-01).

Abstract

Keeping information organized is an important issue to make information access easier. Although the information we need is sometimes available on the Web, this information is only useful if we have the ability to find it. With this aim, it is increasingly frequent to use automatic techniques for grouping documents.

In this thesis we are interested in document clustering, that is, grouping documents based on the similarity of their contents. In this regard, document representation plays a very important role in web page clustering and constitutes the central point of research of this dissertation. Web pages are commonly written in HTML language, that offers explicit information (tags, in this case) about their visual representation, the typography of the text or its structure, among others. It is also a widely used format on the Internet. The main goal of this thesis is to perform a deep study with the aim of making the most of a fuzzy model to represent HTML documents for clustering tasks.

Our study deals with the idea of discovering whether any part of the system could be exploited in a different way to improve clustering results. We begin our work analyzing the parts of the system where there is room for improvement and then we study different alternatives to do so. Thereby, we do not propose a document representation from the beginning, but we build it trying to understand its different parts during each step.

To evaluate our results and compare the different representation proposals, we use different web page collections previously gathered to be used as gold standards. Clustering is performed by using state-of-the-art algorithms and our proposals are validated in environments of plain and hierarchical clustering. Lastly, we also test the usefulness of our approaches in two languages: English and Spanish.

Contents

List of Figures

List of Tables

1
Introduction

In this chapter we motivate the study on the use of a fuzzy combination of criteria for web page representation oriented to clustering tasks. We present the problem from an intuitive point of view and briefly comment the open issues. First, in Section 1.1 we generally describe the problem of organizing documents and the key role of document representation in this task. Next, in Section 1.2 on page 24 we briefly introduce some ideas about how a document can be represented for clustering tasks. In Section 1.3 on page 29 we detail the scope of this thesis. The goals of this thesis are summarized in Section 1.4 on page 30 and finally the structure of this dissertation is described in Section 1.5 on page 32.

1.1 The importance of document organization

Keeping information organized is an important issue to make information access easier. It is easy to figure out how a pile of unordered documents on the desk can help waste time. But ordering documents demands an additional effort, or better said, a tradeoff between this initial effort and the subsequent benefit of having the documents organized. This simple idea has become more and more important as the number of documents to order has increased.

When we manually group documents, we focus our attention on finding common patterns. These patterns could be based on different characteristics of the documents, from textual content to structural similarities, depending on the concrete aim of our organization task. For instance, in this thesis we are interested in automatically grouping documents that share some thematic similarity, so we could rely on their textual content to find these similarities among them. Besides, manual organization of large amounts of documents is not only difficult, but also expensive in terms of time and money, because we need a huge human effort to perform this task.

If instead of papers on the desk we think of electronic documents in a hard disk, the problem becomes harder. Thinking of web documents highlights the problem scale. Therefore, organizing information is a fundamental problem because, although the information we need is sometimes available on the Web, this information is only useful if we have the ability to find it.

For these reasons, it is increasingly frequent to use automatic techniques for grouping documents. On the basis of the previous knowledge employed in each case, in this dissertation we divide these document grouping techniques in:

- Supervised: this is the case of supervised and semi-supervised document classification. In both cases a training stage is carried out, where a set of sample pairs are presented to the system. Each pair consists of a document, usually represented as a vector, and a desired output value for this input, that will be a class in the ideal solution. The difference among supervised and semi-supervised approaches is the number of preclassified instances. Besides, in the semi-supervised approach some not preclassified instances are used in the learning process. From this training, the system tries to infer a function that is called a classifier. The aim of the classifier is assigning a document to one or more classes from a set of predefined classes. In this thesis, we will employ the term *Classification* to refer to this kind of supervised and semi-supervised approaches.

- Unsupervised: this is the case of document clustering, that can be defined as the unsupervised classification of documents (represented as feature vectors in our case) into groups, that are called clusters (Jain et al., 1999). Thus, related documents are organized into clusters. The clustering process is performed without the use of previous knowledge of the group definitions. In this thesis we will employ the term *Clustering*, in contrast with *Classification*, defined above, to refer to this kind of unsupervised approaches.

For clarity, in this dissertation we never use the expression *unsupervised classification* to refer to clustering, though it would not be incorrect. This way we use the term classification only for supervised approaches and clustering for unsupervised ones.

In our case we are interested in grouping documents based on the similarity of their contents, that is, grouping together documents that share a common thematic area, e.g. astronomy, soccer, chemistry, etc. Our research is directed towards grouping documents in absence of category[1] information. This can be particularly useful in environments where new categories could appear, so the system should fit these new categories with the information contained in the

[1] Each group of documents in the ideal solution or gold standard.

documents themselves. For instance, grouping news based on predefined categories could not be always appropriate. In the case of news related with economy, it could happen that at some point, one concrete theme, like *real state bubble*, becomes more important. In this case a new category could be desirable, but in a classification system these documents would be assigned to one of the initially predefined classes. In this sense, a previous clustering process could be used to detect new categories, as this process is not restricted by predefined classes. The same could happen with information like blog entries, forum posts or tweets, where new trends may become very popular in a particular time period. In those cases, clustering techniques could help organize information before having the knowledge about the new categories and it also could help detect their emergence.

Clustering techniques are particularly useful in web mining tasks, where there is a large amount of resources available. These techniques could be utilized as a preprocessing step, allowing to reduce the initial set of pages to a much smaller number of clusters. In this way, clusters can be used instead of documents in the mining process. For instance, clustering is applied in some web search engines in order to organize the large amount of results returned by a broad query. In these cases, clustering methods are applied to automatically create groups of related documents from the set of retrieved documents, in order to ease the browsing of the results. There are some commercial search engines that employ these techniques, like Yippy[2], Helioid[3] or PolyMeta[4]. These systems show that web page clustering can help effectively organize web documents, making navigation and information access easier for users. These techniques are also useful for other tasks such as taxonomy learning, similarity search, and a number of web mining related tasks.

On the other hand, the document clustering process basically involves two steps: document representation and clustering algorithm. Document representation plays a very important role in web page clustering and constitutes the central point of research of this dissertation. Before applying a clustering algorithm over a set of documents, we must represent them, i.e., transforming the documents so that they can be processed by the clustering algorithm in order to obtain the desired clusters. In the document representation step we choose what characteristics of the document we consider useful and how this information could be used to represent the document for our concrete task. For instance, if we wanted to group documents containing similar number of words, we could represent each document using just an integer number, i.e., the number of words in that document. However, our aim is to represent web pages based on their textual content

[2] http://yippy.com/

[3] http://www.helioid.com/

[4] http://www.polymeta.com/

in order to group related documents. In order to evaluate this kind of document clustering process, it is frequent to employ benchmark datasets as gold standards to compare the ideal solution with the answer of the system. In these cases, the goal is evaluating how well the answer of the clustering system matches that gold standard.

Finally, in our case, when dealing with text, each different word can be considered a feature and then, the whole set of features can be used to find related documents. This way, there are several features involved in the representation process. But not only words are available in web pages so, in addition, other information could be employed in the representation process. Finding out the most useful information and the best way to exploit it are unresolved problems so far and both of them focus our attention in this thesis.

1.2 How to represent a document

In order to represent a document for clustering tasks, we can split the process in three stages: selection of feature sources, weighting of those features and dimension reduction. The first stage has to do with the information we want to use to represent each document, e.g., plain textual content, titles, hyperlinks, etc. The weighting process involves assigning a weight to each feature on each document. This weight tries to express the importance of that feature to represent the document in the collection. Finally, the dimension reduction process is needed for computational reasons[5]. To compute the clustering with the whole set of features for a given corpus could not be possible in a reasonable time. Reduction methods aim to remove useless features, keeping those that are more representative for the documents in the collection. Throughout this thesis we deal with these three stages in order to represent web pages for clustering tasks.

In this thesis we focus on the representation of web pages written in HTML format because it is the most usual format on the Internet. The most common approach for document representation is trying to capture the importance of the words in the document by means of term weighting functions following the Vector Space Model (VSM) (Salton et al., 1975). The VSM is an algebraic model that has been used to represent texts in a large number of systems in Information Retrieval (IR), document classification and clustering. It is characterized by assuming that words appearing in the same text are independent, i.e., there is no relation among them, and therefore they can be quantified individually. It also does not take into account the order in which words appear in the text. The use of this model is not expensive in terms of computational cost and it also leads to

[5] The number of features in a dataset composed of about $10,000$ documents could be greater than $200,000$.

reasonably good results. The VSM is described in more detail in Chapter 2.

In order to define a weighting function within the VSM, there are plenty of document representation alternatives relying on word frequency. In the decade of the fifties, H.P. Luhn developed his work about automatic indexing. Based on Zipf's law, he concluded that words with high number of occurrences in the same document are too general and, therefore, their importance to characterize the document should be considered low. On the other hand, the same happens with words that appear seldom in a document, because they are very specific and their presence in the document could be merely anecdotal.

Among the term weighting functions based on word frequency, TF-IDF is one of the most widely used (more details on this weighting function can be found in Section 2.2.2). It was firstly introduced in IR and then moved to classification and clustering tasks, becoming a de facto standard. It combines word frequency in a particular document with the number of documents containing that word in the collection, penalizing the words that appear very frequently in the corpus because they are considered too general to represent a document. This function works with plain text, and it does not directly exploit other additional information that some kind of documents contain. This additional information is frequently used in other weighting functions in combination with word and document frequencies, in order to establish the importance of a word in a document. In particular, HTML tags provide additional information that could be employed to evaluate the importance of document words in addition to word frequency. As we will see in Chapter 2, there are works using HTML elements like lists, headers or tables, in order to enrich the representation. The possibility of identifying these document characteristics within the HTML document leads us to think about representing the documents taking into account how a human being understand that information.

According to the previous idea, in order to select the words that best represent document contents, one of the initial hypothesis of the present work is that a good representation should be based on how humans have a quick look at documents to extract the most important words. We usually search for visual clues used by authors to capture our attention as readers. These visual clues are included in some kind of documents as format information—e.g., in HTML tags—that is visually represented in order to attract the reader.

Therefore, there are different page elements that can be used to judge the importance of a word in a document. Over these elements it is possible to define a set of criteria in order to establish the importance of words. For example, some words are explicitly highlighted by web page authors, like title texts and emphasized words or text segments. In this dissertation we use the term *Criterion* to refer to information that offer some clues about the importance of a word in a document. For instance, to establish the importance of a word in a document, in addition to word frequency in that document, we can also look at the frequency of that word in the title of the document, which is called `Title` criterion hereafter. Therefore, whenever the word `Title` is written in this dissertation, it refers to the frequency of a word in the title of a document, i.e., how many times that word appears in the title, normalized to the maximum number of occurrences of a word in the same title. It is called criterion because it is one of the aspects that are used to estimate word importance.

However, this information is ignored by representations like TF-IDF, based only on word frequencies. For this reason, some researchers have explored the possibility of representing documents by combining different criteria within the VSM. In some cases, each word within a criterion is weighted by means of TF (Term Frequency), TF-IDF or similar alternatives, and then the weights of that word in the different criteria are combined by a linear function (Wang and Kitsuregawa, 2002; Fresno and Ribeiro, 2004; Liu and Liu, 2008; Hammouda and Kamel, 2008).

In this thesis we are interested in the use of fuzzy logic in order to express the knowledge about how to perform the combination of criteria. A fuzzy logic based representation system offers different interesting features:

- First of all, fuzzy systems allow to combine knowledge and experience in a set of linguistic expressions based on words instead of numeric values (Isermann, 1998; Hansen, 2007). This fact increases the expressiveness of the system, making the system easier to understand and analyze. In this regard, the knowledge is expressed by means of a rule base, composed of IF-THEN rules. The general form of a rule in the system is:

  ```
  IF A THEN B
  ```

  where `A` and `B` are propositions that contain linguistic variables. The former, *A*, is composed by one or more *premises*, combined by means of *AND*, *OR* or *XOR* operators. Each premise associates an input variable with a fuzzy set corresponding to a linguistic variable (e.g. `Title IS High`). The latter, `B`, is the *consequence* of the rule where a fuzzy value is assigned to the output linguistic variable (e.g. `Importance IS Very High`).

- Thus, each rule allows to establish a set of related conditions associating some input values to a particular output value (that, in our case, corresponds to the importance of a word in the document and will be referred to as `Importance`). For example, one rule could say that given a *high* value of a word in `Title` and `Emphasis` criteria, then that word is very important and it could be expressed as:

  ```
  IF Title IS High AND Emphasis is High
     THEN Importance IS Very High
  ```

As it can be seen, these rules are close to natural language, and therefore they are easy to understand. Thus, the use of fuzzy logic allows to declare the knowledge without the need of specifying the calculation procedure.

- More than one rule could be fired at the same time by the system. In these cases, the inference engine evaluates all the fired rules taking into account the truth degree of their antecedents. More details about this process can be found in Section 4.2.1.
- The criteria and the rules can easily be modified, introducing new criteria, removing some of the employed ones, changing the definition of some rules or even changing the information capture process (when we define the possible input values for each criterion).
- An important feature of these systems is the non-linearity in the combination of criteria. It allows to set the importance of a word by considering relations among different criteria, that is, to condition the contribution of one criterion to some of the rest of the criteria.

These features are particularly important to ease the task of expressing heuristic knowledge about web page representation, task that in other kind of systems requires an additional effort to understand how the system works in a deeper level of detail.

There are situations where the expressiveness of a fuzzy system can be very useful. For instance, we could consider that emphasized words are important to represent document contents because the author highlighted them. However, in the World Wide Web it is common to write company name in bold font every time it appears in the document, although it will not usually provide useful information to describe document theme. Something similar happens with titles. Titles are sometimes rhetorical. In those cases, they could contain words that are important to represent the document and others that are only used to embellish the language[6]. Because of that, the words that this kind of titles contain do not always help to properly describe the content of the document.

As an illustrative example of such situation, a newspaper published on its website a page with the title: "Call to arms"[7]. This page contains an article about the new trades made by New York Yankees baseball team and how this trades affect to Boston Red Sox, their main rival in the Major League Baseball[8]. Obviously, it would be impossible to describe the document by using only the words in its title, because these words do not summarize the page main topic. More than that, none of these words reflects anything explicitly related with the content of

[6] the term *rhetoric* refers especially to language that sounds impressive but is not actually sincere or useful, according to Longman Dictionary of Contemporary English.

[7] Publication date: Sunday, January 22, 2012

[8] http://mlb.mlb.com/

the page. Nevertheless, with a linear function considering frequency in titles as a criterion for the combination, these words would get a high importance value, which would not correspond with their real importance to describe the content of the page. In these systems, when a word is important in a single criterion, the corresponding component will have a value which will always be added to the importance of the word in the document, regardless of the importance corresponding to the rest of the components. On the contrary, by using fuzzy logic it is possible to define related conditions, e.g., a word should appear in the title and emphasized or within concrete parts of the document to be considered important. In the same way, if a word appears in the title but not in other criteria, then we could consider that word less important. This type of conditions allow us to try to detect whether words in the title refer to page contents and therefore we should consider them important to represent the document.

Along these lines, fuzzy logic was successfully used to represent web pages in classification and clustering (Fresno, 2006). We selected this approach as our starting point, given the benefits offered by fuzzy logic to express heuristic knowledge and its good clustering results. Besides, this approach also produces a vectors within the VSM, so it allows to use this representation with a wide range of clustering algorithms that are also based on the VSM. The criteria included in the combination are:

- `Title`: word frequency in the title of the document.
- `Emphasis`: word frequency in highlighted text segments.
- `Position`: word positions in a document. Basically it refers to whether a word appears more frequently at the beginning, in the middle, or at the end of a document, considering more important the beginning and the end, that are called *preferential* positions, because they usually contain summaries or conclusions highlighting the important information. The rest of the document are considered *standard* positions.
- `Frequency`: word frequency in the document.

Looking at these criteria, another issue appears: the use of knowledge based on assumptions that are not always true. For example, `Position` criterion in the above mentioned fuzzy system is based on the assumption that documents sometimes contain an introduction at the beginning and a summary at the end. Actually, this not always happens. However, by using `Position` conditioned to other criteria could help alleviate its effect on documents where this structure is not strictly followed, as `Position` will only contributes positively to the combination when there are clues that some word is important in other criteria. Thus, if a word is important attending to its `Position` value and also to its `Title` and

Emphasis values, then it could be considered important at all by means of a rule like the following[9]:

```
IF Title IS High AND Emphasis IS Medium
    AND Position IS Preferential THEN Importance IS High
```

In the same way we have the opposite example:

```
IF Title IS Low AND Emphasis IS Low
    AND Position IS Preferential THEN Importance IS Low
```

This way, Position would be conditioned by other criteria, allowing to give more consistency to the final result. In general, this allows to ensure more consistency to the representation, as it makes possible to detect and alleviate cases where a single criterion is not working for some reason, as in the examples exposed above.

As we have seen, fuzzy logic allows us to deal with document representation in a very intuitive manner. Due to all the aforementioned reasons, we believe that fuzzy logic could be an appropriate tool to represent the heuristic knowledge associated to web page representation process. The fuzzy representation presented in Fresno (2006) is a very interesting option to start our research in order to study the combination in different ways, from analyzing its original definition, to propose new ways of exploiting the system to perform the combination, just as to explore the possibility of adapting the system to the input we want to represent.

1.3 Scope of the Thesis

The main goal of this thesis is to perform a deep study with the aim of making the most of a fuzzy model to represent web pages written in HTML language for clustering tasks. We chose HTML documents as an example of documents that contain tags with information related to their structure and visualization.

The framework presented in Fresno (2006) is a very good starting point to explore the possibilities of a combination of criteria based on fuzzy logic to help apply expert knowledge to document representation. In our opinion, the most interesting part of the work is the framework they presented together with the representation. This framework allows to include new criteria to the combination, to modify the existing ones, to analyze the contribution of each one, and to combine these document-level criteria with collection-level information, like anchor texts or document frequencies.

Our study deals with the idea of discovering whether any part of the system could be exploited in a different way to improve clustering results. We begin

[9] These rules are simplified examples to illustrate the text and do not exactly correspond to the systems studied in this thesis.

our work analyzing the parts of the system where there is room for improvement and then we study different alternatives to do so. Thereby, we do not propose a document representation from the beginning, but we build it trying to understand its different parts during each step.

To evaluate our results and compare the different representation proposals, we use different web page collections previously gathered to be used as gold standards. Clustering is performed by using state-of-the-art algorithms, basically the Cluto toolkit[10] and the Self-Organizing Map (SOM). In fact, there are a large number of clustering algorithms in the literature that can be used to tackle different clustering problems. Because of this, we consider important to preserve the independence between the algorithm and the representation. In this way the same representation could be used with different clustering algorithms and therefore it could be applied to different clustering problems.

1.4 Problem Statement and Main Goals

The main goal of this thesis is to study and improve a web page representation based on fuzzy logic applied to clustering tasks. The fuzzy system we use as starting point (Fresno, 2006) provides us with the initial aforementioned framework, but it does not deal with different aspects in depth. There are some interesting issues that were not covered, and from this point of view we divide our main goal in the following subgoals:

1. To compare the fuzzy system representation method with TF-IDF, that is a standard method to represent documents in clustering. Moreover, in previous works it was compared against several state-of-the-art methods, being TF-IDF the strongest alternative (Fresno, 2006; Pérez García-Plaza et al., 2008). Within this comparison, we include to study different dimension reduction techniques. The high dimensionality of the input space, that is, the features available to represent web pages, needs to be reduced to make the clustering process computationally feasible. Different methods are evaluated and compared. Among these methods we include Latent Semantic Indexing (LSI) in the comparison, that is a well known method in the literature.

2. To analyze the initial combination of criteria we use as starting point. This objective is composed of the following more concrete goals:

    - To study the combination of criteria. Each criterion should be isolated from the rest and individually tested. The comparison of the results of these individual tests and the results of the fuzzy combination of

[10] http://glaros.dtc.umn.edu/gkhome/views/cluto

criteria will allow to extract conclusions about the behavior of the combination and the usefulness of each criterion in this combination. This analysis is very interesting in order to clarify what information is more useful to represent a document and to learn about how to improve the combination of criteria.

- To propose alternatives to improve the combination on the basis of the previous analysis and the characteristics of the fuzzy system.

3. To assess the possibility of adding new criteria to the representation beyond the document contents. The original representation was presented as self-content, i.e., it only includes information from the document that is being represented. Therefore, elements like hyperlinks or weighting functions like IDF, that uses collection information, were discarded in the combination. We will consider the possibility of using information external to the page: collection-level information and hyperlinks. In the latter case the information could come from documents external to the collection.

4. To propose a method to adjust the fuzzy logic based representation to concrete collections. The original fuzzy sets employed to capture input information were fixed by default, regardless the collection we want to represent. In contrast to this approach, we study whether the importance of a word in a concrete document is conditioned by the document collection we want to group. For instance, the use of emphasis could be different from one document collection to another. We should take this into account to be more or less restrictive assigning the importance of a word when it is emphasized. Thus, a collection with many emphasized words could suggest to be more restrictive establishing the importance of those words than other collection where emphasis is less commonly used. In this regard, different document collections can reflect different uses of the criteria utilized by the fuzzy system. Thus, given a set of documents with no category information, this objective can be divided in:

    - To identify particular dataset features from dataset statistics that should be treated in a specific way.
    - To adjust the system to fit to those particular dataset features in order to improve our document representation proposal.

5. To evaluate and to validate our representation proposals in an environment of hierarchical clustering. Up to now, the fuzzy representation method has been tested in plain clustering only. Our proposals will be applied in a problem of taxonomy learning with the aim of validating our results in a totally different clustering environment. In the selected problem, web pages represent concepts, and clustering is used to discover a hierarchy

among these concepts. We consider a taxonomy a simplification of an ontology and, therefore, a reference ontology will be used to perform a gold standard based evaluation.

6. To evaluate and to validate our methods with a collection written in other language different from English. In previous works the experiments were always carried out for documents written in English and it would be interesting to extend them to other languages. This process will be carried out following a similar experimental approach than in the previous case.

1.5 Structure of the Thesis

This thesis consists of 7 chapters. Below we provide a brief overview summarizing the contents of each of these chapters.

Chapter 1 on page 21
Introduction
We present the motivation for the study on the use of fuzzy combinations of criteria for document representation in clustering tasks. First we describe the problem of web page clustering and the importance of document representation. Then we talk about the benefits of using fuzzy logic to specify the knowledge about the combination, giving some practical examples. Finally, we formalize the problem and briefly present the open issues and the goals of the thesis.

Chapter 2 on page 35
Related Work
We provide a survey of previous works in the field. We summarize the advances in related fields and put our work in context. We divide the web page representation process in three steps: selection of feature sources. term weighting function and dimension reduction. We also pay attention to the different type of clustering algorithms, briefly describing some of the most relevant approaches.

Chapter 3 on page 75
Selection and Analysis of Web Page Datasets
We describe and analyze in detail the datasets we use for the experimentation carried out along this work. First we describe how they were created and the categories they include. Next we perform an statistical analysis on the composition of their categories and the term distributions on each criterion.

Chapter 4 on page 103
Fuzzy Combinations of Criteria: An Application to Web Page Represen-

tation for Clustering
We describe in detail a fuzzy system oriented to the representation of web pages for clustering tasks. On the basis of this framework. We begin with the evaluation of the system with different dimension reduction methods, one of them proposed in this thesis. Next we analyze the contribution of each criteria to the combination. From our conclusions, we propose and evaluate different fuzzy rule-based alternatives to exploit the system in a different way. Finally we study the possibility of adding some new criteria from the context of the document (collection information and anchor texts) to the combination.

Chapter 5 on page 143

Fitting Document Representation to Specific Datasets by Adjusting Membership Functions
We propose a new way of adjusting a web page representation method based on fuzzy logic to concrete document collections. Different datasets could have different features. In this chapter we investigate whether these features can be identified from collection statistics and employed to tune the representation method. In particular, we study the modification of membership functions taking into account the frequency distributions of the terms.

Chapter 6 on page 165

Test Scenario: Hierarchical Clustering Applied to Learn a Taxonomy from a Set of Web Pages in Two Languages
In previous chapters we evaluate our proposals in a fixed clustering environment. In this chapter we change this environment: we use a comparable corpus written in English and Spanish, a different clustering algorithm and even a different evaluation measure, because we deal with a different problem. In particular, we deal with the task of learning a taxonomy from a set of web pages in two languages: English and Spanish. In this new context, we aim to validate our representation proposals.

Chapter 7 on page 181

Conclusions and Future Research
We discuss and summarize the main conclusions and contributions of the work. We also list our most relevant publications derived from the work presented in this dissertation. We also present an outlook on future directions of the work.

Additionally, the thesis contains the following appendices at the end, with complementary information and summaries in other languages:

Appendix A on page 209

Publications
We list our main publications related with parts of the work presented in this thesis.

Appendix B on page 213
Key Terms and Definitions
We list the most relevant terms related to web page representation and clustering, providing a detailed definition of them.

Appendix C on page 215
List of Acronyms
We provide a list of the acronyms used along the work, and what they stand for.

Appendix D on page 217
Anchor Stop Word List
We provide the list of stop words used in the experimentation with anchor texts of this thesis.

Appendix E on page 219
Resumen (Summary in Spanish)
We summarize the contents of this work in Spanish language.

Appendix F on page 221
Conclusiones (Conclusions in Spanish)
We present the conclusions of this thesis in Spanish language.

2 Related Work

This chapter is a review of previous work we found in the literature. In Section 2.2 on the following page we introduce the problem of web page representation for clustering tasks. As it is a wide research area, we first summarize in Subsection 2.2.1 on page 38 the different ways of selecting the most representative features of a document. Then, we explore in Subsection 2.2.2 on page 54 the works related to term weighting functions. To conclude the section, in Subsection 2.2.3 on page 60 we present the works about dimension reduction, another important step for document representation in clustering. Next, in Section 2.3 on page 66 we present a summary of commonly used clustering algorithms. Finally, in Section 2.4 on page 71 we summarize the contents of this chapter.

2.1 Introduction and Scope

There are several approaches to web page representation for clustering in the literature. Documents—web pages in our case—are represented in several ways depending on the information employed and how this information is utilized. Using a concise document representation is one of the main problems, not only in document clustering, but also in classification, information extraction or IR (Huang, 2008). Because of the representation step is somewhat similar in all of these fields, we also review here some approaches belonging to classification or IR tasks related with this work. However, we focus our attention on representation methods employed within the clustering field. On this basis, in this chapter we analyze the existing problems that web page representation methods try to solve. We also review their drawbacks and the open questions in this field. Different from other works like Oikonomakou and Vazirgiannis (2005) or Patel and Zaveri

(2011), where the attention is focused on the clustering algorithms, we present the related work from the point of view of web page representation.

2.2 Web Page Representation for Document Clustering

Representation is an essential stage for automatic document organization. The way in which documents are organized depends on how they are represented. Different representations can lead to different groupings. In this thesis we focus on web page clustering, that is organizing a dataset in groups of related documents. This relation is based on the similarity among documents. For instance, a group could contain web pages related to sports, another could group others talking about astronomy, etc.

The first thing we have to do is selecting the features or attributes of the documents to be used for representing them. We call features to the elements employed to characterize the documents. These features are used by clustering algorithms to find similarities among documents. In this dissertation, these elements are mainly the words that compound the documents. However, a word is not a feature as it is. First, we preprocess words to convert them to features or terms. A term is basically a preprocessed word. This preprocessing essentially consists of removing punctuation marks, removing stop words, and stemming the words in order to reduce each word to its main part by removing affixes. This preprocessing is also explained later in this dissertation as a part of the experimental settings. It is worth noting that, in this thesis, term and feature are employed as synonyms, because the features we use to represent web pages are basically the terms that such page contains. In this thesis we refer to feature selection as the process of selecting those elements that characterize the pages to represent them. By extension, clustering algorithms will rely on these features. In this section, we first analyze the most employed alternatives to select the most representative features for clustering.

Once we have decided the features we will use to represent documents, the importance of each feature is established by using a weighting function. These functions can sometimes be used with features from different information sources, e.g., textual content of a document, text from titles or anchor texts. Term weighting functions are sometimes independent of feature selection, in the sense that the same weighting function could be applied to features coming from different information sources, as we will see further on in this section. However, both processes are closely related.

The most common way of representing text documents is using the Vector Space Model (VSM) (Salton et al., 1975), where each document is represented as a feature vector, which length corresponds to the number of unique attributes used for representing documents in the collection. Each vector component, that is, each feature, has an associated weight which indicates the importance of that attribute to characterize or represent the document. The underlying idea of VSM was commented in Section 1.2, but more formally, this model can be seen as an instatiation of the Semantic Space Model (SSM) proposed by Lowe (2001). He defined the model as the following quadruple:

$$SSM = \langle A, B, S, M \rangle \tag{2.1}$$

In this model, A is the weighting function. Different weighting functions are described in Section 2.2.2. B is the set of basic elements that determines the dimensionality of the vector space and allows to interpret each vector dimension. In our case, B will be the set of features used to represent a document and they will be selected from different sources as we will see in Section 2.2.1. S is a similarity measure employed to compare pairs of vectors, in our case, pairs of documents. We deal with this aspect in Section 2.3, because it is strongly related with the clustering algorithm. Finally, M is a dimension reduction function. Several dimension reduction techniques are reviewed in Section 2.2.3.

As we said in Section 1.2, the VSM is based on the assumption of word independence, that is to say, there is no relation among words appearing within the same text and therefore they can be quantified individually. The order in which those words appear in the text is not taken into account. Thus, the semantics of a document is reduced to the sum of the individual meanings of the words in that document. Despite these assumptions are incorrect, they allow us to drastically reduce the computational complexity of the problem. By representing documents as vectors, it is not necessary to calculate the dependencies among each possible pair of words and the similarity—or distance—between pairs of vectors is calculated instead. Besides, in many cases, this simple approach is widely used for document clustering tasks and it does not make the results substantially worse (Dhillon et al., 2001).

Another problem in clustering is the high number of features available to represent the documents in a collection. To make the problem computationally feasible, it is needed to reduce the initial number of features in order to alleviate the computational cost, which could make our task impossible to solve, at least in a reasonable time period. There are numerous algorithms for dimension reduction. All of them are directed at removing the features that are less representative for the document topic, keeping those that better characterize it. The ideal result would be the smallest set of features being enough to represent the different documents in the collection as accurately as possible. This would allow to cluster the documents in the desired sets. In the last part of this section we summarize several dimension reduction methods frequently used in the literature.

Summarizing, this section is divided in three subsections: feature selection sources, term weighting functions and dimension reduction techniques. These three steps allow us to represent a document is the VSM for clustering tasks.

2.2.1 Feature Selection Sources

Depending on the information sources employed to extract the features, the different approaches can be considered as:

- **Content based.** They are based on the textual content of documents. This kind of approaches were initially developed for document retrieval in static collections, but with the popularity of the Internet, their use has also been moved to the Web. Thus, in many cases the textual content of the documents has been enriched by the information provided by HTML tags about document formatting, page structure, visual aspects, etc.

- **Link based.** They are based on the link structure among the collection pages. The basic idea is to consider hyperlinks as cites. When two documents have in common many inlinks from other documents, or both documents have outlinks to the same documents, then could be a semantic relation between them.

- **Hybrid.** They combine features from textual content of the document and others from the context of the page. As context we refer not only to hyperlinks or anchor texts, but also to other information sources, such as collection information or definitions from Wikipedia.

We pay special attention to document contents because in this thesis we take a self-content web page representation as starting point. In other words, this representation uses only information available within the document itself. It is worth mentioning the difference between self-content and content-based. Different from the former, the latter could use information external to the document, for instance information related with the whole collection, like IDF (see Equation 2.5). Therefore, both of them are based on the content of the document, but the expression *self-content* implies to utilize only that content. Thus, both concepts are not mutually exclusive and the same representation could be content-based and self-content at the same time, but some content-based functions like TF-IDF (see Equation 2.6) can not be self-content by definition.

Throughout this dissertation we also study the possibility of adding features from page context, like anchor texts, converting our approach from self-content to hybrid, though in both cases our proposal will be content-based.

2.2.1.1 Features from Document Content

Our work is mainly focused on this approach, with the idea of taking full advantage of document contents. Document content is a good starting point for our research and it does not close the door on the employ of additional context information in cases where it is available.

In addition to clustering, there are other tasks, like IR and web page classification, that have also a representation stage for documents. Given this common stage, we take into account works from these fields in our review. In Qi and Davison (2009) we find a summary of features and algorithms employed for web page classification tasks. The authors summarize the works using web page content to select features in two groups: works based on textual content and HTML tags, and works based on visual analysis. We follow the same approximation to structure the rest of this section.

Textual Content and HTML Tags

It is clear that textual content is one of the easiest features we can find in a web page. However, a representation based on a bag of words approach could not lead us to obtain the best results. Furthermore, HTML documents have tags that affect their visual presentation (Qi and Davison, 2009).

For instance, the idea of using HTML tags employed by web page authors to attract reader's interest is explored in (Kwon and Lee, 2003). Their goal is classifying web sites by means of a set of pages linked with the home page of each web site, instead of using only the home pages. Their weighting scheme to establish term importance takes into account different HTML tags as title, headlines, bold font, etc. to find the most representative words in a web page. The authors divide these tags in several groups, and they assign different weight to each group. To clarify the explanation, we will refer here to the weight corresponding to a concrete tag as w_{tag}, while the term weight will be w_{term}. These w_{tag} weights are manually established, according to the estimated expressive power of a tag. The overuse of some tags in concrete web pages is detected by comparing tag frequency to the total frequency of all tags in that page and the proportion in the collection. Thus, they decrease the weight of the overused tag by dividing the percentage of occurrence of the tag in the collection by the percentage of occurrence of that tag in the web page. The way of combining this tag weight (w_{tag}) with term frequency to establish the weight of a term (w_{term}) in a web page follows a linear approach: when a term appears in a tag whose weight is w_{tag}, they count its frequency w_{tag} times instead of 1. Then, the weight w_{term} for a term in a document is calculated by using TF-IDF weighting function, that will be described more in detail in Section 2.2.2. Before weighting terms, they also apply a feature selection process based on the use of category information. Then, they employ a web page classifier over the web pages linked to the home pages as a previous step of a web site classifier. Their results show an improvement in terms of micro-averaging breakeven point—which is the first point at which recall equals precision—than using an ordinary classifier using home pages only.

Another approach related to classification tasks was presented in Golub and Ardö (2005). The aim of their study was to find out the importance of different parts of a web page for automated classification. The authors classified a set of $1,003$ web pages based on titles, headings, metadata and text. They performed an initial classification for each single case, obtaining the best F-measure results for title, followed by headings, metadata and finally text. They derived significance indicators for each part by applying results gained in evaluation by using the F-measure, semantic distance, and multiple regression. Then, the combination of titles, headings, metadata and text is performed in different ways, employing linear combinations based on the term frequency within each element and the corresponding significance indicators. They conclude that the use of all the elements is necessary because not all of them occur on every page. The most interesting conclusion of their work appears when comparing the results of all these different linear combinations. The authors claim that the best combination of significance indicators leads only to an improvement of a 3% over the baseline, where all the elements are considered of equal significance.

The work by Fresno and Ribeiro (2004) presented an Analytical Combination of Criteria (ACC) to represent web pages. It is based on a linear combination of different heuristic criteria within the VSM. These criteria were selected taking into how a human reader have a quick look at a document to extract the most representative words. The criteria used by ACC are:

- `Title`: word frequency in the title of the document.
- `Emphasis`: word frequency in highlighted text segments.
- `Position`: word positions in a document. Basically it refers to whether a word appears more frequently at the beginning, in the middle, or at the end of a document, considering more important the beginning and the end because they could contain summaries or conclusions highlighting the important information.
- `Frequency`: word frequency in the document.

At this point, it is enough to know that ACC allows to set different weights for each criterion in the combination. This way, the result is a vector representing each collection document in the VSM. The set of vectors from a given dataset could be used to feed a clustering algorithm.

Based on the same criteria, Fresno (2006) proposed an alternative way of combining them in a non-linear way. In this case, a fuzzy logic based system is employed to define the expert knowledge about how to combine these criteria. The output is also a single vector within the VSM, representing the estimated importance of each term in a concrete document. Similar to the previous work, this vectors could be used as input for a clustering algorithm.

Visual Analysis

Every web page has at least two representations: its source code written in HTML and its visualization in a web browser (Qi and Davison, 2009).

In Kovacevic et al. (2004) the authors argue that using visual information is a more generic approximation than using HTML tags only, because different tags can produce the same visual result. They use a graph to represent HTML objects and their spatial relation within the page. They apply heuristics to identify different logical areas corresponding to meaningful parts of the page. These logical areas are defined as link lists—vertical or horizontal—, titles, subtitles, paragraphs and other texts. In our opinion, all of these areas can also be identified with HTML tags, although the same effect, e.g. emphasis, can be obtained with different tags. Thus, this approach could be considered similar to other ones based on HTML tags. Nevertheless, their work presents another interesting issue for our research: they classify the text of each logical area individually—using a bag of words approach with term frequencies—, and then they combine these individual results according to the importance of each area by means of a neural network. To establish this importance, they use training information about document categories in the dataset. In clustering, this step is not possible, because we do not have sample data for training. So, strictly, we are not able to set the importance of each area in each category in the same way they do.

Another work with a similar approach to classify web pages is Shih and Karger (2004), focused on recommending interesting links for users and detecting and blocking web advertisements. Their approach is based on the use of the visual placement of links on the referring page. Basically, they use the tree layout of a web page to learn about the position of concrete links. For instance, those links clicked by users that considered them interesting. Then, this learned information can be employed to decide whether a link is interesting or not for users in recommendation systems. In the same way, the authors apply a similar process to detect and block web advertisements. Although it is an interesting approach, it is based on a training stage that uses category information.

We can also find works following this idea in IR, like Yu et al. (2003), where the authors divide a page in sections depending on its visual layout instead of its DOM tree. By using the visual presentation they search for visual cues as lines, images or different font sizes to segment the content. They build an alternative tree from the visual point of view, aiming to find hierarchies that DOM tree misses, since tags are often distributed within the `<BODY>` part without any hierarchies. This approach is applied on pseudo-relevance feedback. They assume that each single segment contains semantically related content and then the term correlations within a segment are higher than the rest. Thus, expansion terms can be extracted from segments and used to improve information retrieval performance. In this case the problem differs from document clustering because in IR we need to find relevant documents for a query, while in document clustering we need to extract the aboutness of the document without any given term to guide us.

Along these lines, Chen et al. (2009) present a method for extracting news from the Web, based on the visual perception of human users. They try to simulate how human beings understand the information they found in web news by using a function based object model. The objects of this model can be of four main types: information object, navigation object, interaction object and decoration object. These objects are extended to other more concrete ones, as text/media information object, that is an information object containing text. The idea is representing a page using a hierarchy of objects with their functions associated (by identifying the concrete type of each object). Then, a merging process based on several axioms and corollaries is applied over objects. This process is oriented to discover adjacent areas whose contents are related. As evaluation measure, they employ F-measure and their experimental results show the feasibility of these ideas for automatic information extraction. However this technique aims to extract news content from pages and it does not deal with the problem of finding the more representative words in the page or, from a more general point of view, finding relations between different news. It is important to mention that their heuristics try to identify news. In this sense they use the visual representation of the page to identify concrete page parts, and not to represent the page in order to perform any kind of comparison to other pages, that would be the case of our work.

Concluding Remarks on Features from Document Content

Taking everything as a whole, the authors Qi and Davison (2009) conclude that using visual information is more effective than using HTML tags. They refer that HTML tags are more oriented to representation than to semantics. Due to the inconsistent formation of web documents, web page authors could generate different tag structures with the same visual result. Nevertheless, and as far as we know, visual analysis approach may also suffer inconsistencies, since different web browsers, and even different versions of the same browser, do not always show the same visual representation for the same HTML code. In other words, web pages are not always rendered the same, but depending on the engine.

We have seen several works in different tasks that combine different page elements, as title, headings, meta-data, or plain text. All of these elements are available in web pages, thanks to HTML tags.They are frequently used in web page representation in order to exploit the information contained in documents beyond plain text. These works show that the combination of some of them can help improve other alternatives based in plain textual content only.

2.2.1.2 Features from Link Structure

These methods are based on the assumption that when two documents are connected by a hyperlink, then there is a semantic relation between them. This way, it is possible to employ these relations to divide the collection in clusters. However, the employ of link structure is not frequent in clustering works. This kind of information has been traditionally more used in the IR field. Because of this, the most relevant works which have successfully exploited link structure are focused on improving IR systems. This structure has also been used in works related to clustering and classification of HTML documents, using techniques frequently inherited from previous ones developed in the IR field, as shown in Getoor (2003), where a review about link mining techniques is presented.

The use of link structure to group document collections comes from citation analysis (Garfield, 1979), a commonly used bibliometric method, where it is assumed that if the author of a document cites another two documents, then these two documents should be somehow related from the author's point of view. By applying the ideas from bilbiometrics to the Web, Larson (1996) represents information by means of an input matrix where its element (i, j) contains the number of documents citing both documents i and j. After that, the raw co-citation matrix is converted to a correlation matrix. Finally the author applies multidimensional scaling techniques to represent the data in a two-dimensional map where it is possible to analyze the relations and grouping of the documents.

There are numerous works in the IR field using this structure and it is worth mentioning them because of their impact in the last decade and because they may contribute some ideas about what kind of information is worth to employ. One of the most popular applications that uses this information is the PageRank algorithm (Page et al., 1999), that generates a ranking of search results where the importance of a page depends, among other factors, on the pages pointing at it. In the original version, they took into account the number of pages pointing at the page we are interested in and their importance. Later works tried to improve this approach from a similar perspective. For example, in Massa and Hayes (2005) the authors propose to identify trusted websites with a semantic extension of PageRank algorithm by using additional information from hyperlinks. Following this trend, Borgs et al. (2010) propose a number of principles to extend PageRank aiming to establish relations of trust and distrust in recommender systems.

Another very well known method in IR is the HITS algorithm (Kleinberg, 1999), that identifies two types of pages or communities: authorities and hubs. The former are considered very important because they receive a huge number of inlinks. The latter are pages pointing at a number of important pages. As in the case of PageRank, there are many works in the literature whose research is focused on alleviating HITS drawbacks or analyzing some of its aspects or applications. For instance, Hung et al. (2010) propose a new way of addressing the topic drift problem, that appears when a page not related with the original topic has high link density and leads to authoritative but not relevant pages. Furthermore, the HITS algorithm showed low performance in some experiments, but Peserico and Pretto (2009) warns about the possibility of mistaken insufficient iterations for an intrinsic deficiency of the algorithm.

Combining the idea of HITS with citation analysis, Kumar et al. (1999) present a method called *trawling* in order to discover new emerging cyber-communities by means of web page clustering. The two ideas behind this method are: (1) relevant web pages are frequently cited together even before their authors realize that there is a community, and (2) these communities consist of hubs and authorities mutually reinforced.

More recently, Garza Villarreal et al. (2009) presented a framework that uses hyperlinks for topic detection by means of clustering techniques. They consider the corpus as a directed graph and document clusters as topics. Of course, though we can consider the fact that clusters have some implicit semantic properties, a topic detection stage would be needed to add semantics to the clusters. As one could expect, documents are represented as graph nodes, while hyperlinks are represented by edges between nodes. For the experiments, they use a small subset of Wikipedia articles, composed of $7,041$ documents and $452,702$ hyperlinks. With respect to the clustering algorithms, three were tested: Principal Direction Divisive Partitioning (Boley, 1998), a hierarchical divisive algorithm; K-means, introducing the link information using an adjacency matrix; and Graph Local Clustering (GLC) (Virtanen, 2003), that is based on the use of a local search method to maximize a graph-theoretic fitness function. The evaluation is performed in terms of internal measures: cosine similarity, semantic relatedness and Jaccard index. All of them are applied to compare intra-cluster and inter-cluster similarity achieving favorable results, particularly for K-means and GLC. Nevertheless, there is no comparison against other methods that allows us to conclude about the usefulness of this work. In the same manner, further research is needed to evaluate the applicability of these techniques in a large scale dataset.

In this section we do not go into any further detail because, in web page clustering, link structure is usually combined with document contents or other criteria. Attending to the specific characteristics of clustering problems, information from document contents or hybrid approaches—which we will see in section

2.2.1.3—may be more appropriate. We believe this combination is needed in order to establish thematic relations among documents in a most solid way.

2.2.1.3 Hybrid Approaches

In this case, features come from different information sources aiming to exploit the information contributed by each one. Actually, it can be seen as a way of extending document contents with external information sources. The main sources we consider in this review are: link structure, anchor texts and Wikipedia. Among them, anchor texts are related to links and to page textual contents also. For this reason we separate these approaches from link structure based approaches. In the case of Wikipedia, it is a publicly available source of encyclopedic type information that has become popular among researchers in recent years.

Combination of Content and Link Structure

There are several works that combine document contents with link structure in order to represent documents. On the one hand, although hyperlinks may be understood as author's recommendations to other pages, actually they might not imply any similarity among that pages. Besides, the number of available hyperlinks could affect the result, either we could not have enough hyperlinks or, on the contrary, we could have a too dense link structure. On the other hand, text based methods may suffer problems in some scenarios, for example, when dealing with different languages, or when the author uses synonyms, etc. Apart from this, web pages usually contain other type of multimedia information different from text. Because of this, the use of features from both sources might be suitable for concrete cases. In this context, Fisher and Everson (2003) analyzed the usefulness of links for web page classification tasks. They conclude that links may be either useful or harmful depending on link density and quality.

In Yang et al. (2002) a study of hypertext regularities is presented. They use web page content as the main information source in order to explore where it is possible to find additional information that sometimes is not explicit in the text itself. They focus their research on web page classification, but their conclusions are also valid for clustering works. After a comprehensive study using different classification algorithms, the authors highlight the importance of the identification of different hypertext regularities in the data, as well as the selection of appropriate representations for web pages. The regularities they define and study are the following:

- **No hypertext regularity:** when the only useful source of information is the document itself. In these cases, it is better to represent the document by means of document contents only, since the use of external information could even be harmful for the system performance.

- **Encyclopedia regularity:** when a document of a given category only contains outlinks to other documents belonging to the same category. This restriction makes very difficult to find this regularity outside collections composed of encyclopedia articles.

- **Co-referencing regularity:** like the previous one, but linked documents have other relation in common with the page from which they receive the inlinks, instead of belonging to the same category.

- **Preclassified regularity:** when a single document contains outlinks to other documents that share the same category. Thus, identifying this document would help to group all the documents that are linked from it. It is common to find this regularity when there is a preclassification of the documents, as occurs in web directories like Yahoo! Directory or the Open Directory Project.

- **Meta data regularity:** meta data information is often available from external sources on the Web, and it is also present in an implicit way in the page itself, within some HTML tags like `<META>` or `<ALT>`, which are not shown to the users by web browsers. The problem is the small percentage of pages including this type of elements.

Encyclopedia, Co-referencing and Preclassified regularities are impossible to detect in clustering problems when we do not have category information. However, they give us an idea of the type of information we can expect to find when using links to represent web pages. Finally they conclude that the use of a type of information or another should depend on the particular document collection and on the problem, i.e., it does not exist a unique solution for all the cases. These regularities are difficult—some of them even impossible—to identify in clustering tasks and assuming the existence of the wrong regularities could be harmful for the clustering results.

Regarding web page clustering, the work presented by Takahashi et al. (2005) combined results from two different clusterings, one using textual content and another using link structure. For the textual content side, they adopted a binary representation within the VSM instead of using term frequencies. The link structure is represented by means of a graph, where pages are nodes and arcs between nodes are hyperlinks between pages. Then, both representations are expressed as dissimilarity matrices where each component (i, j) represents the dissimilarity between the documents i and j. For the graph, the dissimilarity is computed as:

$$d_{i,j} = 1 - \frac{2 \cdot |\text{From}(a_i) \cap \text{From}(a_j)|}{|\text{From}(a_i)| + |\text{From}(a_j)|} \tag{2.2}$$

where a_i and a_j are graph nodes, and $|\text{From}(a)|$ denotes the out-degree of the node a. For document vectors, the dissimilarity is calculated as follows:

$$d_{i,j} = 1 - \frac{(\vec{p}_i \cdot \vec{p}_j)}{|\vec{p}_i| \cdot |\vec{p}_j|} \tag{2.3}$$

where $\vec{p}_i$ and $\vec{p}_j$ are document vectors. Each dissimilarity matrix is clustered separately by means of complete link hierarchical clustering and then both alternatives are combined in a single clustering solution which corresponds to the intersection of both single solutions:

$$C_{i,j} = A_i \cap B_j \tag{2.4}$$

where A_i and B_j are the clustering results for both approaches respectively. However, the main drawback of this method comes from the absence of a quantitative evaluation. The authors just performed a qualitative analysis without any conclusion comparing different methods. Besides, this method is focused on finding a way of combining two clustering outputs, instead of unifying the representation process.

In order to overcome this limitation (merging two different clustering processes), Fersini et al. (2010) presented a method that models both document contents and link structure in an unified manner, though they employed different representations for each one. Their approach relies on the assumption that the probability of a web page to belong to a category can be determined not only by its contents, but also by analyzing the contents of other pages referencing it and the strength of relations with them. On the one hand, textual contents were represented by means of TF-IDF. On the other hand, the link structure was employed by using a directed graph where nodes are documents and arcs are probabilistic links between them. These links express the probability of jumping from one document to another, calculated as the arithmetic mean of internal and external coherence. The former is defined as the coherence of the semantic area in which the link is located in the source page. The latter is the same but with respect to the target page. The semantic area around a link is extracted by dividing the document in different visual blocks based on the layout information embedded into the HTML tags.

They used the same approach as Cai et al. (2003) to detect visual blocks, where the DOM tree structure is combined with visual clues obtained from a web browser. Some of the visual separators used to identify blocks are `<HR>` tags, different background colors, several text nodes with no other tags surrounding them (that would correspond to a visual block) or different text sizes, among others.

Once we have the intuitive idea of what a visual block is, the internal coherence can be seen as the relation between the terms belonging to the visual block that contains the link and the terms in the rest of the document. When a subset of terms that appears in the visual block are frequent throughout the rest of the document, the internal coherence tends to be high. The external coherence follows the same idea. It can be seen as the relative frequency of terms that occurs in both the target page and the link block, with respect to the number of occurrences of the most important term in the target document.

The proposed algorithm is based on assuming that the jumping probability is able to estimate the joint probability that both source and target pages belong to the same cluster. But during a clustering process we neither know the category of documents, nor the possible document category labels. The authors estimate the probabilistic evidence that a document is similar to a representative element of the cluster by means of TF-IDF representation of the documents and the calculation of cosine similarity between them. As a single document can be linked from different pages, a Bayesian approach is followed to calculate the probability of the document to belong to the different possible categories. This approximation does not clearly separate web page representation and clustering process. In order to evaluate their proposal they do not compare representations but algorithms, improving results obtained by K-means and expectation maximization in terms of F-measure, entropy and corrected rand coefficient (Hubert and Arabie, 1985).

Combination of Content and Anchor Texts

Besides the link structure, anchor texts are among the most commonly employed elements to represent web pages. In the work of Noll and Meinel (2008) the authors state that anchor texts provide meaningful information for IR tasks. Nevertheless, they also state that this information is not so good for capturing the aboutness of web documents. They agree with Eiron and McCurley (2003) about the similarity among anchor texts and search queries with regard to term distribution and length. They also found that anchor texts are generally less likely to appear in document content. In the case of the study of Noll and Meinel (2008), they found anchor texts also in web page content in a 51% of cases. Result of a similar study, Eiron and McCurley (2003) found a 66, 4% of cases.

Within the IR field, Xu and Zuo (2004) compared three ways of using information from context:

- Anchor texts.
- Anchor context by following the approach proposed by Pant (2003), where HTML DOM tree is used to extract this context.
- A technique, proposed by the authors, based on natural language processing as an alternative to the method of Pant (2003).

The proposed technique relies on two steps: location of the cohesive text region around the anchor text, and use of an English parser to extract the relevant sentence fragments in the text region and the nearest heading text. The former step is based on the DOM tree, searching for block elements as <P> or <DIV> to identify areas that can be considered cohesive due to their formatting, that is, following the assumption that the author employed these HTML tags to group cohesive texts. The latter step uses the syntactic tree of the sentence where the anchor text appears to extract its context by searching for some halting tags. These halting tags are determined by the following assumption: the anchor texts are almost always noun phrases, so the authors define the halting tag as the node in the parsing tree that is not a noun phrase, such as verb phrase and preposition phrase. This assumption tries to guarantee the inclusion of as much relevant modifiers as possible, avoiding noise by stopping just before the first halting tag found in the parsing tree.

In order to evaluate the different approaches they use WebKB dataset (see Section 3.3), but ignoring category labels and grouping documents in four subdatasets corresponding to the four Universities. Then, they use the words in a concrete link context as query terms and the documents in the same subdataset as potential retrieval results, computing the relevance of each page with respect to the extracted link context. Their metric employs the rank of the page as performance measure. Their results show no significant differences between the former two alternatives, anchor texts and anchor contexts based on DOM tree. The latter approach improves the others, but with a much higher computational cost. This leads us to believe that the use of anchor texts is a light an efficient alternative to more complex methods for extracting anchor context.

In Liu and Liu (2008) the authors consider that document title, textual content and anchor texts have different importance levels and decide to represent each one with a separate feature vector. Thus, each document is represented by a tuple of three vectors and each vector is represented within the VSM with TF-IDF weighting function. One disadvantage of this approach is that it requires a particular clustering algorithm, so it cannot be used with other algorithms we find in the literature, at least as it is proposed (though it would be possible to combine the vectors or just to concatenate them in a single document vector to adapt the representation to the VSM). Because of that, it was not compared to other representations, but to other algorithms. Another important disadvantage of this model is that it does not allow to include new criteria to represent documents without changing the whole system (input format and algorithm). Furthermore, they do not analyze the contribution of each criteria to the combination. About the evaluation, a manually classified dataset containing 1,600 documents divided in 11 categories is used, but the manual process is not explained. Finally, their experiments are based in average precision only, improving the results of K-means algorithm, but not those obtained by hierarchical K-means (Chen et al., 2005). It is worth noting that including recall could lead to different conclusions.

Combination of Content, Link Structure and Anchor Texts

It is also possible to take into account all the previous sources to represent web pages. For instance, in Wang and Kitsuregawa (2002), a study about how to combine textual content and link analysis is performed. They use inlinks and outlinks in order to improve clustering applied to search results. The terms are extracted from four different parts: text snippet from search results, anchor texts, meta-content and anchor window of the inlinks to the page (there is no clear information about which inlinks are considered in this task). These four parts are merged for each page in search results and a stemming process is applied to extract terms. Thus, each page is represented by means of three vectors. Two of them are composed of binary values indicating whether the *i*th inlink or outlink, depending on the vector, is present for the web page being represented. This way, checking whether two pages are cited by the same page (coupling) or checking whether both cite the same page (co-citation) can be done easily by comparing the corresponding vectors. The last vector contains the frequency of the terms extracted following the above mentioned method. The combination is performed with a linear function. In their experiments they vary the coefficients in order to find the best way of combining them. They use a corpus composed of 200 URLs from search results, corresponding to 8 different topics arbitrarily selected and they extract 100 inlinks per each URL. This experimental corpus was created by using Google search engine. The evaluation measures they employ are precision, recall, entropy and distribution. The interpretation of the latter one is explained by the authors as follows: "High distribution means low quality of the clustering since similar pages are scattered into several groups instead of grouping together". As clustering algorithm they propose a variation of K-means, utilizing a similarity threshold instead of pre-defined k centroids to control the clustering process. Their empirical results suggest that the combination of both, textual content and links can improve web page clustering from search results, and this improvement come from the way they are combined, giving more weight to one or another. In their work they obtain the best results when 50% of the total weight comes from links and the other 50% from textual content. However, the authors do not propose a way to calculate these coefficients from the data, they just analyze their empirical results where those coefficients are manually modified to cover a range of possible combinations.

Combination of Content and Wikipedia Information

In addition to link-related information and anchor texts, there are other hybrid approaches that use other kind of information external to the document. Among the most recent works, we find the use of Wikipedia as external information source to extract concept and category information. For instance, this information is combined with document content in works like Hu et al. (2009), where instead of using the typical bag of words approach, the authors represent each document by means of three vectors: document content vector, concept vector and category vector. All of them use TF-IDF weighting applied over different elements (content, concepts and categories). To associate documents to Wikipedia concepts and categories, they utilize the connection between concepts and categories which is explicit in Wikipedia. Then, the document-concept matrix is built through two matching schemes: exact-match and relatedness-match.

For the exact match case, they first build a dictionary using Wikipedia concepts. Each topic is described by only one article in Wikipedia, that will correspond to a dictionary entry. A preferred phrase is chosen to be the title of the article, but there are no details about how to choose it. Each dictionary entry also contains all redirected concepts representing the same topic, that are gathered by using the redirect links of Wikipedia. Thus, both preferred concepts and redirected concepts are retrieved from the documents we want to cluster. The concept vector is built only with preferred concepts for each document. The weight of each preferred concept corresponds to its frequency added to the frequencies of all the corresponding redirected concepts. Once they have built the document-concept matrix, composed of the frequencies of each concept appearing in a document, the authors calculate the document-concept TF-IDF matrix.

The relatedness-match case aims to cover the cases of concepts that do not explicitly appear in a document. To do that, they build a document-concept matrix from Wikipedia article collection. First, the word-concept matrix is used as an intermediate step to associate documents with Wikipedia concepts, which is done by calculating the relatedness between a term and a concept. The TF-IDF weight of each document term in the concept and in the document itself is multiplied, and then the values corresponding to all the terms in the document are combined by summation. Finally they take the 200 top concepts with highest relatedness score to represent each document.

In both cases, exact and relatedness match, the document-category matrix is calculated by combining concept-category matrix and document-concept matrix. For evaluation, they present an approach based on the content as a baseline and then the different possible combinations between content, concepts and categories. The metrics employed are purity, F-measure, and normalized mutual information. For the combination of content with category, and content with concepts, different values from 0 to 1 with intervals of 0.1 are tested as coefficients in a linear combination of each couple. Then, the three vectors are also linearly combined and the coefficients used correspond to the best results achieved in the previous experiments where content and concepts, and content and categories were combined. Thus, the authors present the best possible results of a linear combination, but there is not a way to get these results without the supervision introduced to extract the coefficients. It is also worth mentioning the need of preprocessing Wikipedia articles. In this case, the whole wikipedia dataset contains $911,028$ articles and about $29,000$ categories after pre-processing and filtering (both processes are not concretely specified). The clustering experiments were performed over three datasets: TDT2[1], LA Times[2] (from TREC), and 20-newsgroups[3]. They compare three type of document vectors and their combinations: word vectors, concept vectors and category vectors. The word vector approach is used as baseline. The combination of category and document content achieved the best results in most of the cases. Between exact and relatedness match, the former outperformed the latter in two out of three datasets. However, it is worth mentioning that the whole datasets are not utilized. For efficiency reasons, they select five subdatasets of 100 documents from each selected category of a given dataset. Although the authors do not offer details about computational cost, it seems that the cost of the whole process constitutes an important drawback of this system.

A similar approach is presented in Huang et al. (2009), where a different way of combining vectors in a linear fashion is presented. The way of selecting Wikipedia relevant concepts is also different from the above mentioned work by Hu et al. (2009). They build a vocabulary of phrases extracted from Wikipedia anchor texts and then, they match those phrases with the textual content of the documents they want to cluster. However, the same anchor texts could appear in different Wikipedia concepts. To solve this ambiguity, they use a classifier whose input are the possible targets for a given anchor text and all unambiguous anchors from the surrounding text, that are used as context. The output of the classifier is the probability of each sense to be the intended one. Then, the sense with the highest probability is selected. The weighting function for concepts is also TF-IDF, and the similarity between documents is calculated with a linear combination of text and concepts. As in the previous case, the coefficient utilized in the combination is set based on preliminary results, and there is no analysis about the influence of modifying it.

In Hu et al. (2009) they based their representation on three types of vectors, each one corresponding to a different source of features: document, concept and category vectors. In Huang et al. (2009) approximation, concepts and categories are unified by means of a single semantic relatedness measure. The evaluation is performed by means of purity, inverse purity and micro-averaged F-measure. The datasets to cluster are a subset of Reuters-21578 containing $1,658$ short news arti-

[1] http://www.gabormelli.com/RKB/TDT2

[2] http://www.gabormelli.com/RKB/LA_Times_Dataset

[3] http://qwone.com/~jason/20Newsgroups/

cles in 30 categories and OHSUMed, composed of 23 categories and 18, 302 documents. The snapshot of Wikipedia they used contains five million distinct phrases linking to almost all of the two million articles. They also experiment with Latent Semantic Indexing and Independent Component Analysis over concepts (we will comment both techniques in Section 2.2.3), but only for Reuters dataset, because of the computational cost of applying these techniques on OHSUMed, due to its high number of documents. Their results show that both techniques do not offer any advantage in this scenario, arguing that both are applied globally to the Reuters dataset without using any information related to the categories in the collection. From the point of view of the authors, this fact is the reason why the latent semantic structures that are found do not retain sufficient information to differentiate the categories. They also compare their approach to other similar works, particularly they use the traditional bag of words as baseline, and Hu et al. (2009) approach as the reference to be improved. However, the comparison is performed by means of purity and inverse purity only and they do not show the results of F-measure in this case. The authors claim an improvement about a 12% and a 14% in both collections in terms of purity, comparing their method against a bag of words approach. The authors also estate that their approach and the approximation of Hu et al. (2009) achieve comparable results. Nevertheless, parsing the whole Wikipedia corpus could be a expensive task that, added to the classifier used to disambiguate senses. It would be interesting to perform a complete complexity analysis to determine the applicability of this approach. Finally, the linear combination of criteria rely on a coefficient that is not clearly defined, so changing this value could lead to different results, even using the proposed value could be not suitable when working with other datasets. On the whole, the use of Wikipedia as source of information could be positive in terms of clustering results, though the associated computational cost and the selection of the coefficient for the combination should be taken into account depending on the task we deal with.

Concluding Remarks on Hybrid Approaches

Summarizing, we have reviewed different options for adding contextual information to document representation. Link structure, anchor texts and Wikipedia are commonly used sources to enrich the content of web pages in order to represent them. The use of any of these methods depends, of course, on the availability of the needed information.

We have seen that links could not always be a suitable way to represent the contents of a given page, since their quality and density varies, and could even be harmful. Anchor texts suffer a similar problem. In both cases, their usefulness will depend on the dataset. Therefore, we ought to take much care to apply them to general problems.

Wikipedia approaches have the problem of parsing Wikipedia corpus prior to the representation stage. However the results showed by different researchers are encouraging. Although in this thesis we do not employ Wikipedia as external information source, it could be interesting to explore this possibility as future work.

In general, in the three cases—links, anchor texts and Wikipedia—, an important factor is the computational cost needed to collect the information before representing the documents. It is also worth noting that different from document contents, collecting context information can be much more expensive in terms of time, and this information sometimes can not be available.

2.2.2 Term Weighting Functions

In this section we describe different term weighting functions found in the literature. As we saw in the previous section, it is common to fix a standard weighting function when testing different feature selection sources. For this reason, we could not avoid to mention weighting functions like TF-IDF in the previous section, although they are described in more detail in the present section.

We review different alternatives from the point of view of the feature sources they use. First we talk about functions that are commonly applied to plain text. Next we describe methods that combine different feature sources by means of linear combinations. Finally we review other alternatives to combine features from different sources.

Term Weighting Functions Applied to Text

First of all, we center our attention on weighting functions applied to text. The weight of a feature in the VSM could be calculated using a straightforward method like a binary value meaning whether the feature is present in the document or not, or employing more complex weighting functions. The weighting method will depend on the goal of the representation. One of the most commonly used weighting functions is TF-IDF, where term frequency (TF) in a document is combined with the Inverse Document Frequency (IDF) of that term in the whole collection:

$$\text{IDF}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2.5}$$

where t is a term, d a document, D the whole corpus, $|D|$ is the total number of documents in the corpus and $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears. Then, IDF is linearly combined with TF to calculate TF-IDF:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \tag{2.6}$$

This function comes from the fields of automatic text representation and IR. It does not take into account the additional information one can find in web pages, just plain text. Despite that, there are many works using TF-IDF aimed at web page clustering.

Despite TF-IDF is widely used, there are cases where the term frequency information available is not sufficient to represent texts. When dealing with short texts or snippets, term frequencies can be very low, hindering the use of weighting functions based on them. In this context, the use of textual energy (Fernández et al., 2007) has been proposed. The concept of textual energy was inspired by statistical physics of magnetic systems and it was applied to study fundamental problems of Natural Language Processing. Later, in Molina et al. (2010) it was applied as a distance measure between short texts, corresponding to definitions in this case, represented within the VSM. To do that, they divide the document in Lexical Entities (LEs). The list of all unique LEs appearing in a corpus is called here a dictionary. This way, definitions are represented as vectors with as many dimensions as different lexical entities exist in the dictionary. These vectors contain binary values, 0 or 1 depending on whether the LE appears or not in the document. To obtain the textual energy, the elements of the document-LE matrix are considered as the neurons of a Hopfield network (Hopfield, 1988), where the documents are defined as neuron chains. A neuron is active (its value will be 1) when the LE appears in the document or inactive (0) otherwise. Thus, the textual energy is calculated as:

$$E_{text} = -\frac{1}{2}(X \times X^T)^2 \tag{2.7}$$

where X and X^T are the document-LE matrix and the transpose of that matrix. As E_{text} contains negative numbers or zeros, the absolute value can be considered:

$$E = |-E_{text}| \tag{2.8}$$

And finally this matrix E can be represented as an array of a single dimension containing the distances between each pair of vectors:

$$D_e = [e_{1,2}, e_{1,3}, \ldots, e_{1,n}, e_{2,3}, e_{2,4}, \ldots, e_{2,n}, \ldots, e_{n-1,n}] \tag{2.9}$$

These distances are normalized using the maximum energy value. As the array represents the distances between all pairs of documents, it can be used in combination with a clustering algorithm to create groups of documents. The authors use a hierarchical agglomerative algorithm and compare their results with a baseline using Hamming distance (Hamming, 1950). Their results are encouraging in terms of precision and recall, but to the best of our knowledge, this kind of approaches have been applied to short texts and not to complete documents, neither compared with TF-IDF or other term frequency based functions.

Term Weighting Functions Based on Linear Combinations

Nowadays, TF-IDF has become a de facto standard in document clustering. Because of that, some researchers have presented new representations based on variations of TF-IDF. In Hammouda and Kamel (2008) the authors propose to employ keyphrases instead of words with a function derived from TF-IDF, introducing some changes like rewarding, instead of penalizing, keyphrases that appears in many documents:

$$\text{score}(p) = (\sqrt{w \cdot pf} \cdot d^2) \times -\log(1 - DF) \tag{2.10}$$

where p is a phrase, w the average weight of the phrase over all documents, pf the average number of times this phrase appears in one document, normalized by the length of the document in words, d is the average location of the first occurrence of the phrase in the document, that is calculated on the basis of the number of words before and after the first phrase occurrence, and DF is the document frequency of the phrase in the whole collection. The weight w of a phrase is calculated taking into account whether it appears in titles or headers by means of a linear combination. They multiply keyphrase frequency by a fixed value in each case, but they do not specify the exact value, or the way of calculating it, or even the way of calculating the weight itself. The clustering evaluation is performed by means of F-measure, and a two-sample t-test is applied to evaluate statistical significance. However, they do not compare document representations, but distributed clustering algorithms, so they do not compare their methods with other ones not based on keyphrases.

Directly focused on web page representation we found the work of Fresno and Ribeiro (2004), commented in section 2.2.1.1. Their representation proposal is called ACC and it is based on a linear combination of four criteria: `Title`, `Emphasis`, `Position` and `Frequency`. The weight of a term is calculated by taking its frequency in each criterion (normalized to the maximum frequency of a term in the corresponding criterion) and multiplying these frequencies by a set of coefficients. The coefficients for the linear combination were set on the basis of a statistical analysis over a set of heterogeneous web pages. However, the results of this study could change for specific sets of web pages, so the validity of these coefficients is not clear for all the cases. The main drawback of this approach is common to all the linear approaches and it was explained in Chapter 1: the fact that the contribution of one criterion is independent of the rest of the criteria. Then, it is not possible to express related conditions to establish term importance.

There are also works exploiting HTML tags in combination with anchor texts and link structure. We previously review the work by Wang and Kitsuregawa (2002) in Section 2.2.1.3, when we talked about hybrid approaches to select features. Focusing on the weighting side, the authors combined textual content and links. The textual content was extracted from text snippet from search results, anchor texts, meta-content and anchor window of the inlinks to the page. Terms were extracted from these four parts and merged in a single term frequency vector. They used two more vectors, one for inlinks and another for outlinks. They combined links and text by means of a linear combination where each component had an associated coefficient. Their best results were achieved when 50% of the total weight came from links and the other 50% from textual content. However, there is no proposal to calculate these coefficients from the data. As we seen previously, in their study those coefficients were manually modified to cover a range of possible combinations.

It is also worth mentioning here the works by Hu et al. (2009) and Huang et al. (2009) that were also reviewed in Section 2.2.1.3. Both of them employed Wikipedia as a external source of information to enrich document representations. They extract information about concept and categories and then they linearly combine it with document contents. Both of them relies on the use TF-IDF as weighting function. The coefficients for the combination are set on the basis of preliminary tests and there is no analysis on their influence on the clustering results.

Finally, there are other works exploiting HTML tags and using TF-IDF as term weighting function, where the combination is done by means of the algorithm. These works have the drawback of breaking the independence between representation and clustering stages, and then the whole system need to be modified to introduce changes on either side. In our particular case, this is an important negative aspect, as it does not allow to test different web page representations with the same algorithm. These kind of tests are needed to compare different representations in the same conditions, that is the only way to be sure of the effect of the representation in the clustering process.

In this line, other work we commented before in Section 2.2.1.3 is the approach of Liu and Liu (2008). Document title, textual content and anchor texts were individually represented by means of TF-IDF. Then, the algorithm performed the combination of the three vectors. It was done by taking into account the weighted average of the three vectors for computing the cosine similarity between documents. However, the coefficients for the weighted average were neither fixed nor data driven. These coefficients were manually established. In fact, the authors used different combinations to test the system. Their approach was not compared to other representations, but to other algorithms, improving K-means, but not hierarchical K-means in terms of average precision.

Other Combinations

In addition to linear approaches above mentioned, other works also focus on the combination from the point of view of the algorithm, instead of the representation. For example, Takahashi et al. (2005) proposed to perform the clustering separately for document contents and link structure. We reviewed their approach in Section 2.2.1.3. They clustered both separately by means of complete link hierarchical clustering and then, both clustering results were combined in a unique clustering solution which corresponds to the intersection of both individual solutions (see Equation 2.4). Nevertheless this method did not propose a new way of weighting or representing, but only a way of combining clustering results.

We also talked about the proposal of Fersini et al. (2010) as an example of modeling both document contents and link structure in an unified manner. They represent document textual content by means of TF-IDF and link structure by means of a directed graph, where nodes are documents and arcs express the probability of jumping from one document to another. We also saw that their approximation does not clearly separate web page representation and clustering process. However, to the best of our knowledge, at the same time they keep content representation and link structure clearly separated, so other functions different from TF-IDF could be used with this algorithm. They evaluated their proposal by comparing it to other algorithms, improving results obtained by K-means and expectation maximization in terms of F-measure, entropy and corrected rand coefficient (Hubert and Arabie, 1985).

These two works above mentioned did not present a proper term weighting function, but methods to combine some term weighting functions during the clustering process. Different from them, in Section 2.2.1.1 we mentioned the approach presented by Fresno (2006): a self-content representation based on a fuzzy combination of criteria. It was called Fuzzy Combination of Criteria (FCC) and has been successfully applied in clustering and classification. FCC works within the VSM by using a fuzzy system to heuristically combine criteria. Concretely, four criteria are used:

- Term frequency.
- Term frequency in document title.
- Term frequency in emphasized text segments.
- Term positions in the document.

Besides, the fuzzy logic engine provides the possibility of adding new criteria and modify the rules easily, which allows to study the contribution of each criterion. As the result of FCC will be a vector within the VSM, it could be used with different state-of-the-art algorithms. However, the fact that FCC is self-content may be questionable in the sense that it could be analyzed whether adding external information, like IDF or anchor texts, helps improve the representation. In Chapter 4 of this dissertation we analyze FCC in depth, explaining its basics and covering these issues among others.

Concluding Remarks on Term Weighting Functions

Among term weighting functions employed in web page clustering, TF-IDF is frequently used to represent document textual content. It is sometimes used as a part of linear combinations that try to capture features from different sources (title, headers, anchor texts, emphasis, etc.), assigning different importance levels to the features coming from each source. However, in linear combinations this importance is set by means of coefficients. In most cases these coefficients are not data driven, but manually or empirically established. Some authors detected that these coefficients needed to be changed for different datasets to make the most of the combination. However, none of the reviewed approaches proposes a way to calculate the coefficients based on the dataset we want to cluster without using category information.

We found other methods where the combination of features from different sources was performed by means of the algorithm. In those cases, the representation methods do not combine anything. In fact, those methods relies on other representations, e.g., TF-IDF for the content side or a directed graph for the link structure, that are used without any specif modifications. Thus, the algorithm is specifically designed for the concrete input. For this reason, other representations cannot be tested with the same algorithm and the tests to validate this kind of approaches are oriented to compare them to other algorithms.

In this thesis we want to investigate about web page representation. We want to employ a representation independent of the algorithm, in such a way that it can be used with different state-of-the-art algorithms. Different problems may require different clustering algorithms, e.g., the most intuitive case could be the difference between plain clustering and hierarchical clustering. A web page representation independent of the algorithm does not need to deal with the clustering algorithm applied after. In this sense, as we said in section 2.2, we focus on the VSM because is the most common way of representing documents and there are a number of state-of-the-art clustering algorithms relying on this model.

As we said before, we want to take advantage of HTML language to extract some visual clues that authors use to attract the interest of the reader. We are interested in exploring the usefulness of different criteria, coming from different feature selection sources. Most of the works on clustering HTML web pages employ linear combinations of criteria. First, these approaches suffer the above mentioned limitations related with the way of establishing the coefficients for the combination. Also, this kind of approximations has the problem of fixing the contribution of each component to the combination, regardless the rest of the components (we explained this problem intuitively in Section 1.2 and we will review it in more detail in Chapter 4). Finally, despite of the drawbacks of linear combinations, they show that the knowledge about how to calculate the importance of a term in a document is not exact and the possibility of employing heuristic knowledge in the process seems natural. Thus, approaches based on non-linear combinations could be more suitable for our task.

For these reasons, we have decided to focus on a fuzzy combination of criteria. Our decision of using a representation within the VSM—to grant independence from the algorithm—and based on non-linear combinations of criteria, led us to FCC representation. Utilizing fuzzy logic as a tool for the combination allows to express the knowledge by means of rules written in natural language, improving their legibility. This constitutes another important factor when it comes to working with heuristic combinations, as approaching to natural language should ease the expression of knowledge.

On the other hand, as we have seen in previous works, there are several approaches based on a combination of textual content and different sources of external information. Therefore, any improvement in one of the sides will also benefit the combination. For textual content, TF-IDF is usually employed, so any other function working within the VSM could be used instead. Our work, though focused on document representation using its contents, could also be used in these hybrid approaches. In this sense, the better the results of each individual part of the combination, the better the results we could expect from the combination. Besides, anchor texts from inlinks are sometimes used in the representation in addition to document content. They have been used in the combination as text of the web page being represented, as well as separately. It would be interesting to analyze the effect of the anchor texts in a non-linear combination of criteria.

On the whole, the combination of web page content and links could lead to good clustering results if we have suitable collections having a sufficient number of quality links. But actually, this does not always happens. In those cases the representation will only rely on the content of the page. That's why we consider so important having a good content-based representation, in order to guarantee the best possible grouping quality, whether we have context to combine or not.

2.2.3 Dimension Reduction

First of all, it is worth explaining some terms we start using from this section. We employ the word dimension because we represent documents within the VSM, where each document is a vector. So dimension reduction implies reducing the number of vector components, that is, the number of features employed to represent a document. Thus, we call vocabulary to a set of features we use to represent the documents in a particular collection. Therefore, in this thesis we refer to the size of the vocabulary as the dimension of the document vectors.

When we work with a large number of web documents, we also find the problem of having a very much larger number of features available. So basically, dimension reduction implies reducing the initial set of features, e.g., composed of all the different terms that appear in the documents of the collection, to a smaller one. Although it sounds quite simple, dimension reduction plays an important role, because it is used to reduce computational cost. At the same time, it should remove useless features by selecting those more representative to find relations among documents in the collection we want to cluster. In other words, dimension reduction is very important to select those features that lead us to a good grouping, and furthermore, it is necessary to reduce the computational cost of clustering algorithms, otherwise too high so that the problem could be unapproachable. To give an idea about the magnitude of the problem, for example, a collection of 10,000 documents could have an associated vocabulary from 200,000 to 700,000 features or dimensions. These numbers clarify the importance of dimension reduction.

There are many different approaches in the literature, from techniques based only on term frequency in the collection, to more complex methods coming from the classification field and adapted to document clustering. The use of one or another depends on the problem we work on and its requirements.

Approaches Traditionally Employed for Clustering Tasks

In Tang et al. (2005) a comparison of several alternatives of dimension reduction techniques for document clustering is presented. Concretely they compare three methods based on feature transformation[4]:

- Latent Semantic Indexing (LSI) (Landauer et al., 1998): this method is also called Latent Semantic Analysis. The underlying idea is reducing dimensionality by mapping related terms to the same dimensions. To do this, LSI assumes that words with similar meanings occur close together in the text. Basically, LSI projects the initial space of documents and their words into another vector space of reduced dimensions, where terms are independent on the basis of the previous assumption: they do not co-occur and then they are mapped to different dimensions. The input term-document matrix is reduced by using a mathematical technique called Singular Value Decomposition (SVD) (Wall et al., 2003), a matrix decomposition technique. SVD can be seen as a method for data reduction, a way to find a good ap-

[4] These methods modify the initial features trying to find independent components. When applying reduction methods based on feature transformation, the features in the reduced set have no direct correspondence with the features in the initial set, since they were transformed in the reduction process.

proximation to the original data, yet using fewer dimensions by ignoring data variations below a threshold. A more detailed explanation about LSI and SVD including practical examples can be read in Van de Cruys (2010).

- Random Projection (RP) (Bingham and Mannila, 2001): it is a lightweight alternative to LSI. The original data with d dimensions is projected to another subspace with lower dimensionality k by using a kxd random matrix with unit length columns. The idea behind RP comes from the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984): "if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved".

- Independent Component Analysis (ICA) (Hyvärinen and Oja, 2000): this method tries to transform the input data to obtain statistically independent components. To be used as a dimension reduction technique, the data is preprocessed using principal component analysis, where the new dimensions are ordered by their importance. These data are then transformed into independent components by using ICA. To do this, ICA maximizes the statistical independence of the estimated components. There are different ways to define independence, though two of the most widely used in ICA algorithms are minimization of mutual information and maximization of non-gaussianity. Intuitively, ICA is based on the assumption that the input data come from unknown linear combinations of some hidden variables. These variables are the above mentioned independent components, and the algorithm aims to find them. More formally, the process can be seen as the decomposition of the data into a linear sum of non-orthogonal basis vectors with hidden variables being statistically independent (Choi, 2009).

Besides, three methods based on feature selection[5] are used in the comparison:

- Document Frequency (DF): it is based on the idea that terms appearing in few or many documents should be removed because they could not be useful to represent those documents. Then, DF technique involves removing terms having very high or low document frequency. In order to decide what is considered very high or low document frequency, the corresponding thresholds are established. Thus these thresholds are used to decide the final size of the feature set.

[5] These methods reduce the input space by selecting a subset of features that satisfies a given criterion. Therefore, the features in the reduced set have direct correspondence with the features in the initial set.

- Mean TF-IDF (TI): it is based on using the TF-IDF mean value for each term over the whole dataset as a measure of term quality. Thus, the higher the mean, the higher the quality of the term. To reduce dimensionality, the terms are ranked on the basis of their means and then the best n ranked terms are selected, being n the number of terms we want to use for the reduced vocabulary. This method is proposed by Tang et al. (2005) and it is not a standard process in clustering works.
- Term Frequency Variance (TFV): the quality of a term is measured based on the variance of its frequency:

$$\mathrm{TFV}_i = \sum_{j=1}^{n} \mathrm{TF}_j^2 - \frac{1}{n}\left[\sum_{j=1}^{n} \mathrm{TF}_j\right]^2 \tag{2.11}$$

  where i is the term, j the document and n the total number of documents in the collection. Then, the idea is to rank the terms based on its TFV and reducing the vocabulary in the same manner than in the TI case.

The weighting function used is TF-IDF for all cases. The evaluation is performed using three datasets containing from 500 to 9,000 documents: Reuters 21578[6], CSTR[7] and WebKB[8], being the last one formed by web pages. The comparison is performed by using a paired Student's t-test and classification accuracy. They conclude that LSI and ICA outperform RP, in particular with aggressive reductions—that results in smaller number of features—, but they are computationally more expensive methods than RP. Between them, they lead to similar results in CSTR dataset, while ICA performs slightly better in Reuters and LSI slightly outperforms ICA in WebKB. Among the three feature selection based alternatives, TI and TFV outperform DF in Reuters dataset, but they perform similar in the others. The results over WebKB are particularly interesting for us because it is composed of web pages and we also use this dataset to evaluate our proposals in this thesis. In this sense, LSI seems to be the best alternative in WebKB among the feature transformation methods, while among the three feature selection methods there are no significative differences. Finally, to alleviate computational complexity of LSI and ICA, the authors suggest to combine them with any other previous—and lighter—technique of dimension reduction. They experiment with TI and TFV as first stage reductions before applying LSI and ICA, obtaining results comparable to the ones from the original methods, with the benefit of a lower computational cost.

A similar work was presented by Mohamed et al. (2006). The authors propose a new technique and compare it against different state-of-the-art alternatives: DF,

[6] http://www.cs.cmu.edu/~TextLearning/datasets.html

[7] http://www.cs.rochester.edu/trs

[8] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

RP, LSI and ICA. Their proposed technique needs a previously classified collection to build a dictionary, where term importance is calculated in the different categories. Then, this preclassified collection should cover all the categories and topics that are present in the collection we want to cluster. The dictionary will contain the terms that appear in the preclassified dataset and the dictionary building process involves assigning a weight to each term in each category. The weighting function employed is TF-IDF, normalizing the weights on each category. The result will be a single vector for each term, containing the weights of that term in the different categories. For each term in a document we want to cluster, its corresponding vector is retrieved from the dictionary and the weights of the term in each category are then multiplied by the TF-IDF weight of the term in the document, ignoring those terms that do not appear in the dictionary. The final feature vector is created by adding the weights of each term for all the possible dictionary entries. Therefore, this technique could not be used in an unsupervised environment, where we do not have any other information than the documents themselves. In addition, the preprocessing step requires to represent another collection, and this collection should be chosen based on the categories we want to create during the clustering process, which again is not possible when the categories are unknown beforehand. Another drawback of this approach is the size of the reduced vocabulary, because it is not possible to select it as it depends on the number of categories in the collection used to build the dictionary. Anyway, the rest of the techniques in their comparison are unsupervised. They use WebKB dataset in their experiments and purity as evaluation measure, concluding that LSI is the best among the unsupervised methods, followed by ICA, DF and finally RP.

In addition to the above mentioned, there are several previous works that apply DF, RP or LSI for document organization tasks, though they are not focused on dimensionality reduction techniques. For instance, in Kohonen et al. (2000) a SOM is used to organize large document collections based on textual similarities. As the number of documents is very large, the authors decided to use RP instead of LSI after testing both methods, due to the much lower computational cost of RP. Comparing both techniques in terms of accuracy, they found a good trade-off between computational cost and accuracy, despite the worse accuracy results achieved of RP. In Correa and Ludermir (2006) another approach to organize large document collections by means of a SOM is presented. In this case, DF reduction is applied by removing terms occurring in more than half of the dataset and less than five documents. They do not compare this reduction to other ones.

Classification Approaches Adapted for Clustering Tasks

There are also feature selection methods inspired by supervised classification alternatives, like Park and Kwon (2007). Nevertheless, their approach is not applicable in unsupervised environments, because it requires to have half of the collection pre-classified. They represent the documents by means of TF-IDF weighting function and then reduce vector dimensionality by applying mutual information (Wang and Lochovsky, 2004) and information gain (Yang and Pedersen, 1997) techniques. These techniques need category information, reason why a half of the dataset, previously classified, is used in the dimensionality reduction process.

To solve this issue, in Huang et al. (2006) the authors propose the use of partial clustering solutions, obtained by an iterative clustering process. After one iteration of clustering, each input vector is assigned to a cluster. Then, they assume that each cluster corresponds to a real class. This approach is similar to the ones presented by Liu et al. (2003) and Li et al. (2008), but in those cases only one type of feature is used, while in the approximation proposed by Huang et al. (2006), in addition to textual content, they also use information about the URL, from anchor texts and from users' access logs.

For this reason, we will comment here the work by Huang et al. (2006) as an example of adapting feature selection methods from classification to clustering. First, for each source of information they conduct a different iterative clustering with K-means algorithm producing different feature spaces. The data from textual content, URLs and anchor texts is weighted by means of TF-IDF. Regarding users' logs, they refer to web pages and queries, indicating which web pages have been visited by users after a query. For each web page, a feature vector is created to represent the number of times each query is associated with the page we want to represent. The authors then consider each query as a feature and apply TF-IDF weighting over the data. Several feature selection methods are used on each feature space: information gain, chi-square, correlation coefficient, relevance score, odds ratio and GSS coefficient. As each clustering iteration produces different results on each feature space, all of them are combined by means of a fusion function. They tested five fusion models including voting, average value, maximum value, average rank, and max rank, all of them defined in their paper. In order to combine the features from the different spaces, the authors establish a selection percentage. They empirically test the values 0.9, 0.8, 0.6, 0.4, 0.2, and 0.1 for each type of feature, and then they choose the one by which their iterative method has the best performance. They evaluate their approach with error rate, F-measure and entropy. They used two subsets of WebKB (Section 3.3), both of them smaller than the one we use in this dissertation. In particular, they remove *other* category and use only two and three categories respectively (there is no concrete information about the categories they are using). Additionally, they also utilize a dataset based on the Open Directory Project, composed of 15 categories and $8,071$ documents, including user access logs from the MSN search engine. They also employ clustering with DF feature selection as the baseline, improving it in most of the cases.

All in all, we find again the problem of manually (or empirically) determining the coefficients to linearly combine features from different sources. In this case the authors provide a table with the values they set for the different datasets they experiment with. It is worth mentioning they use different values for each collection, which points out two different problems: the previously mentioned issue of linearly combining features from different sources and the fact that different datasets would require different methods for information capture.

Conclusions on Dimension Reduction Techniques

Summarizing, among the dimension reduction techniques commonly applied in clustering, RP is used in the literature as a lightweight alternative of LSI, although RP always lead to slightly worse results. Besides, DF is also widely used, due to its simplicity: terms appearing in few or many documents are removed because they are not useful to distinguish among those documents. These three reductions are employed and evaluated in this thesis in Chapter 4.

The works applying feature selection techniques adapted from classification to clustering tasks show interesting approaches and results. However, as far as we know, none of them has been adopted as a standard method in clustering tasks. Other issue about these methods has to do with the way in which the clustering is performed. Traditionally, dimension reduction is applied before the clustering process, aiming to reduce the computational cost of clustering vectors with a large number of dimensions. In the above mentioned works it is not clear how the dimensionality affects the algorithm, as the whole feature set should be used for feature selection and thus, on each iteration, the clustering algorithm has to deal with a large number of features. Then, they require a first clustering step to select features. Besides, Huang et al. (2006) used different values to combine the different feature spaces for each collection, which points out two different problems: the previously mentioned issue of linearly combining features from different sources, and the fact that different datasets would require different methods for information capture. We deal with the first of these problems in Chapter 4, while the problem of adjusting the information capture process to concrete datasets is studied in Chapter 5 of this dissertation.

2.3 Clustering Algorithms

Document clustering is the process of automatically group a collection of documents, web pages in our case, in such a way that elements within the same group are somehow similar. This similarity will depend on how the documents are represented and the measure used to calculate the similarity among them. This thesis is focused on finding groups of thematically related pages, where each group would correspond to a major category, e.g. astronomy, soccer, chemistry, etc.

In this section we review some well known clustering algorithms. There are several ways to classify clustering algorithms, in this work we divide them following a similar approximation as Oikonomakou and Vazirgiannis (2005) and Patel and Zaveri (2011):

- **Partitive Algorithms:** they carry out a flat, non hierarchical, partitioning of the input space, dividing the collection into a predefined number of clusters. Their input is a matrix composed of feature vectors and they are based on the optimization of a criterion function (Zhao and Karypis, 2001, 2004). One of the most popular algorithms using this approximation is K-means, which also has several variants like Forsati et al. (2008); Liu and Liu (2008); Mahdavi et al. (2008) or Carullo et al. (2009). Among the advantages of these kind of methods we find their simplicity and their low computational cost. In contrast, they depend on several parameters that may lead to different clustering solutions. Some of these parameters are the number of desired clusters, the input order to process the documents, the similarity measure, or the selection of the criterion function.

- **Hierarchical Algorithms:** these algorithms merge two clusters or split a cluster into two during each step, producing, as a result, a tree structure called dendrogram[9]. Although there are some divisive algorithms within this group, most of them are agglomerative. Depending on the way of calculating the similarity between pairs of clusters, they are divided in:

  - *Single link:* the similarity between two clusters is the greatest similarity between two documents, belonging one to each cluster, i.e., the similarity between their most similar members. This tends to produce long clusters, which is called chaining effect[10].

  - *Complete link:* the similarity between two clusters is the smallest similarity between two documents, belonging one to each cluster, i.e., the similarity between their least similar members. This tends to produce compact clusters with small diameter.

  - *Group average:* it is an intermediate approximation between the previous ones. The similarity between two clusters is calculated as the av-

[9] A tree-like diagram that organizes data in subcategories which are divided until the desired level of detail is reached.

[10] Two clusters are forced to merge themselves due to the proximity between two of their elements, even though many of the remaining elements of each cluster are far away from each other.

erage similarity among the elements of both clusters. This method is suitable for obtaining clusters composed by homogeneous elements.

  - *Ward's method* (Ward Jr., 1963)*:* this method is different from the rest because it uses variance analysis. At each step the sum of squares within the clusters is minimized over all possible combinations obtained by merging two of the clusters from the immediately prior step. That is, this method iteratively merges the most homogeneous clusters. It is less sensitive to outliers[11] and tends to create tightly bound spherical clusters. As a counterpart, though it is considered a very efficient method, it also tends to generate clusters of small size (Quan et al., 2003; Hill and Lewicki, 2007).
  - *Centroid/Median methods:* the clusters to be merged at each stage are those which centroids or medians are more similar. Different from the centroid, the median is not weighted proportionally to the size of the cluster. This method is considered appropriate when there is a significant variation in group sizes.

- **Graph-based Algorithms:** documents are modeled as nodes of a graph and its edges represent relations between documents. These edges have a weight that indicates the similarity degree between the connected nodes, i.e., between the corresponding documents. The idea behind this is that clusters contain documents (nodes in the graph) connected by edges with greater weights than edges between clusters. The differences among this kind of algorithms stems from the way of creating the graph, as well as the method employed to divide it in clusters. The use of graphs allows to capture the structure of data and to work in an effective way in spaces with a large number of dimensions.

- **Neural Networks:** the Self-Organizing Maps (SOM) (Kohonen, 1990; Kohonen et al., 2001) are unsupervised neural networks that use competitive learning for producing a spatial-topological relation among the vectors that characterize its neurons. This is achieved by means of a training process based on a set of input vectors without external information (in contrast with other machine learning approaches where the training phase needs category information to be carried out). The whole process may also be seen as a projection of an initial space with a large number of dimensions to another one with usually 2 or 3 dimensions. This eases the graphical representation of the input data. This method usually gives considerably good results without high computational complexity, but representing the output in a space with so few dimensions may result in information loss. It

[11] Atypical values, observations that are numerically distant from the rest of the data.

is worth mentioning the works of Honkela et al. (1997) and Honkela (1998) because they were the first in applying the SOM to massive organization of document collections. Recently in Liu and Liu (2008) the authors rely on the SOM to develop a modification by using a tuple of vectors to represent each document and neuron. Their work was previously commented in Section 2.2.2. In a prior step within the process of this thesis, we experimented with the SOM and our results were presented in Pérez García-Plaza et al. (2008). Later we explored new ways of organizing tag clouds also employing this algorithm in Zubiaga et al. (2009a) and Pérez García-Plaza et al. (2012).

- **Probabilistic Algorithms:** the basic idea is to assign probabilities for expressing the membership degree of a document to a cluster. They assume that data can be partitioned into clusters that are characterized by a probability distribution function. The algorithm Expectation Maximization (Dempster et al., 1977) is an example of this kind of methods.
- **Fuzzy Algorithms:** a single document could be assigned to more than one cluster, that is, it takes into account the possibility of overlapping between clusters. The solution is usually obtained by means of the optimization of a criterion function. One of the most widely used of these algorithms is the so-called Fuzzy C-means (Bezdek et al., 1984), a modification of K-means.

All of these methods are not exclusive, that is, there are alternatives based on combining some of them in order to find a solution for clustering problems.

We are interested in a clustering process where there is no overlapping between categories. Besides, the number of clusters, also known as k, strongly affects the clustering quality. In most cases, we cannot easily determine this value. To make the evaluation process more intuitive, we decided to use the number of categories. This way, a perfect clustering it is possible and the clustering process does not introduce additional bias to the representation step, in such a way that when we modify the representation, we can be sure that the variation on the clustering results will come from that modification.

Within the framework of this thesis where most of the experiments involves plain clustering, we chose an state-of-the-art clustering algorithm called *rbr* (k-way repeated bisections globally optimized), that belongs to the Cluto package[12] (Karypis, 2003), a software toolkit for clustering high-dimensional datasets. This algorithm computes a first solution by means of a sequence of $k-1$ repeated bisections of the initial input set—where k is the desired number of clusters—and then, it tries to globally optimize the clustering criterion function. It is a widely used algorithm with good results in the literature (Fresno, 2006; Montalvo et al., 2007; Zhuhadar and Nasraoui, 2008; Giannopoulos et al., 2008; Ghani and Kumar,

[12] http://glaros.dtc.umn.edu/gkhome/views/cluto

2010; Dredze et al., 2010; Tan and Mitra, 2010), although not all the works used the globally optimized version. In our case we selected this alternative because, in practice, the global optimization process ensures the algorithm computes the same clustering solution for the same input data (Van de Cruys, 2010).

In terms of results, Zhao and Karypis (2004) state than depending on the criterion function, the use of *rbr* or its version without global optimization does not differ very much, in particular with functions I2, H1 and H2[13]:

> *"[...] as we optimize either I2 , H1 , or H2 , the overall cluster quality changes only slightly (sometimes it gets better and sometimes it gets worse)".*

In our case we use the I2 criterion function, set by default for the *rbr* algorithm, given the good performance of this combination.

Furthermore, in Zhao et al. (2005) an empirical evaluation of nine agglomerative and six partitive algorithms applied to solve problems of hierarchical clustering is presented. The experiments show that partitive methods works better than the agglomerative ones in plain as well as hierarchical clustering.

These works also demonstrate that partitive methods are suitable to produce clustering solutions from document collections with plain or hierarchical organization, in an effective and efficient way.

At the same time, we need to calculate the similarity or distance between documents or clusters. Not all measures works well in any case, because, in many cases, the features that define the clusters depend on the concrete data or problem we want to solve. Thus, there is not any particular measure that performs better than the rest for any kind of clustering problem. Despite that, there are several measures that are commonly used, such as cosine similarity between two vectors—this is probably the most widely used—, the Jaccard correlation coefficient, Euclidean distance or Kullback-Leibler divergence, though reviewing the literature it is easy to find a number of alternatives (Van Rijsbergen, 1979; Strehl et al., 2000; Huang, 2008). In the latter two cited works the authors agree that cosine similarity, Jaccard correlation, Pearson coefficient and Kullback-Leibler divergence lead them to really close results, and significantly better than Euclidean distance.

In this thesis we use cosine similarity between documents for plain clustering using Cluto *rbr*, because is the most widely used within the VSM (Zhao and Karypis, 2002). This measure guarantees independence with respect to document length.

Lastly, in this dissertation we propose a new test scenario for our proposals. Different from the rest of the experiments, we modify the clustering framework to deal with a different algorithm, a different dataset, a different goal and, therefore,

[13] I2 , H1 y H2 are three of the criterion functions available within the Cluto package.

a different evaluation procedure. In other words, we just keep the representations untouched and we modify the rest of the clustering framework. Basically, we propose to evaluate our proposals in a taxonomy learning process by means of hierarchical clustering. To this end, we employ an agglomerative hierarchical binary clustering method applied on a SOM. This way, a hierarchy is built with different SOMs on each level. The root will contain the whole document corpus that will be divided in several clusters on each tree level. Then, each level will contain several SOMs with different subcollections, one per each cluster on the immediately upper level. Thus, the result of merging all the SOMs at the same tree level would correspond to the whole dataset. We have presented this method in Paukkeri et al. (2010) and Paukkeri et al. (2012) and it is explained in detail in Chapter 6.

2.4 Conclusion

Throughout this chapter we have summarized the related work we found in the literature, covering different alternatives for document representation. Mainly, we have focused our review from the point of view of web page clustering tasks.

The web page representation models we found in the literature usually differ from each other in the information sources they use, the weighting functions they apply over such information, and the dimension reduction techniques they employ. In spite of many of them belong to the group of hybrid representations, a detailed analysis of each approach—content based and context based—separately could allow to obtain better combinations. As Fresno (2006) previously stated, obtaining a good representation for web pages is a task very dependent on the processes we want to apply after. This way, a representation oriented to IR tasks should not consider the same elements as in the case of automatic text classification or document clustering, where document content may result more relevant.

Several issues emerge from our review:

- Among the term weighting functions, TF-IDF or sometimes just TF are applied in most of the works—even in classification tasks, where category information is available for training—, although these functions do not exploit other information than textual content.

- In order to improve the results of TF-IDF as document representation method, several alternatives have been proposed to represent web pages taking advantage of different page information. Most of the works rely on criteria as document titles, emphasized words, headers or hyperlinks to enrich TF-IDF.

- In most cases, the way of combining criteria follows a linear approach, where the importance of a term in a single criterion is calculated regardless the rest of the components. As we stated from the beginning of this thesis, we consider fuzzy combinations of criteria a more suitable way to perform this task. This thesis establishes its framework around a fuzzy logic system oriented to web page representation. Such system allow non-linear combinations of criteria, in addition to a high level definition of the heuristic knowledge.

- To the best of our knowledge, most of the linear combinations of criteria are based on manually or empirically selected coefficients. These coefficients strongly affects the results but, however, there are no proposals to automatically determine their values when category information is not available. In fact, in some cases we have seen that these coefficients need to be empirically adjusted to each single collection in order to achieve better results. This point out the fact that each different collection could need different adjustments for the combination. Other works fix their values beforehand, but most of them do not explain the reasons for that selection. Our proposals are directed towards finding a web page representation method allowing to easily express expert knowledge about the combination of criteria, having the ability to adapt those criteria to the concrete characteristics of a particular dataset.
- In other cases, the combination of criteria is carried out by the algorithm, losing the independence between the representation and clustering processes. Then, to introduce some modification on either side, representation or algorithm, the whole system has to be changed. Besides, this approach does not allow to perform a direct comparison of different document representations. This way, these kind of approximations make difficult to analyze whether the benefits or drawbacks of an approach come from the representation or from the algorithm. Lastly, in this thesis, we aim to propose a web page representation that can be applied in different clustering scenarios. For these reasons, we consider the independence between representation and algorithm very important.
- Dimension reduction can also affect the effectiveness of the representation process, as it should remove useless features by selecting those more representative to find relations among documents in the collection. There are works comparing different approaches as DF, RP, ICA or LSI. Besides, LSI, RP and DF are widely used to reduce dimensionality when dealing with clustering problems in the literature. Nevertheless, these works do not analyze how each of them behave with different weighting functions.
- The use of link structure has been combined with content information in most cases. Most of the combinations used a standard weighting function

within the VSM for content side. In this sense, this function could be substituted by other VSM-based alternative. Thus, by improving either the link structure side or the content side, the improvement should be reflected in the final combination. There are also works using anchor texts linearly combined with document contents. In this linear combinations, anchor texts are treated as other elements like titles or term frequency, that come from document contents. In other cases, anchor texts terms are used in a similar manner as terms from the contents of the page, i.e., directly adding them to page contents. However, this combinations usually include other elements as page titles or link structure, and we did not find any study on the usefulness of adding anchor texts in particular to page contents for clustering tasks.

- Wikipedia has also been used as external source of information to enrich document representation. These works suffer two main drawbacks: the use of a linear combination with coefficients based on preliminary results to integrate Wikipedia-based information with document contents, and the need of parsing the Wikipedia corpus to extract concept information as a previous step for document representation. In this thesis we do not consider the use of external information from encyclopedias or ontologies.

- Most of the research works on web page clustering include a quantitative evaluation of their results. This quantitative evaluation is performed by means of a gold standard when it is available, by utilizing precision, recall and F-measure in most of the cases. To analyze the significance of their results, the statistical t-test is also used by some researchers. In some cases, entropy is also employed to evaluate clustering results. On the other hand, accuracy is rarely used, and finally, in few cases, the evaluation is totally performed from a qualitative perspective.

- The datasets employed in evaluation differ from one work to another. There is no a standard set of web page collections for evaluating clustering tasks. In this sense, even when the same dataset is used, like is the case of WebKB, each work utilizes it with a different preprocessing. For example, it is common to use only some of its categories, or even just a part of these categories. Moreover, the filtering process is not always well described. In this dissertation we describe the collections we employ and the considerations we make on each one.

Taking all the above mentioned issues into account, we perceive the lack of a standard methodology to compare web page representations. Each work establishes its own framework and, though some aspects are shared through different works, they do not follow a common process that allow to obtain results comparable with previous works. In this thesis we try to make the representation process independent from the rest, centering our research and modifications mainly on this stage, at the same time we try to keep the rest of the framework as standard as possible, by means of employing techniques, algorithms, datasets and measures widely used in the literature.

3

Selection and Analysis of Web Page Datasets

This chapter describes and analyzes in detail the datasets we use throughout this work. The underlying idea is presenting all the information about the datasets in the same place, in order to establish their similarities and differences that will be useful to extract conclusions about the results of our experiments in future chapters. At the same time, this organization aims to grant independence between the concrete experiments we perform in some chapters and the datasets we employ, because not all the datasets are utilized in the same chapters, yet some datasets are used in different chapters.

The chapter is organized as follows. First, in Section 3.1 we describe our requirements and criteria to select and study the datasets we use in this thesis. In Sections 3.2 on page 77, 3.3 on page 83, 3.4 on page 90 and 3.5 on page 94 we comprehensively analyze the features of the selected datasets. For each and every dataset we explore how the documents were collected and the categories created. We also study the term frequency distributions in the whole document, emphasis and titles. Finally, we conclude the chapter summarizing our findings in Section 3.6 on page 98.

3.1 Selection of Web Page Datasets

There are different aspects we want to study in this work. First of all, we are interested in web page datasets composed of HTML documents. As our initial framework is based on the work developed by Fresno (2006), we decide to use the same datasets in order to keep backwards compatibility from the results point of view. These reference collections are Banksearch (Sinka and Corne, 2005) and WebKB (Craven et al., 2000).

Furthermore, in this thesis we are interested in exploring the use of anchor texts as a source of contextual information. Both aforementioned collections were created some years ago and many of their pages no longer exist. Therefore, it is very difficult to find anchor texts outside the datasets themselves. For this reason, we decide to use the Social-ODP-2k9 Dataset (Zubiaga et al., 2009b), a newer collection with most of the pages still available on-line during our research. We later describe the process we follow to extend this collection with anchor text information.

Finally, we employed a collection presented in Paukkeri et al. (2012) composed of Wikipedia documents about animals to perform experiments about hierarchical clustering in two languages: English and Spanish.

All of these four datasets are studied in the following sections in order to characterize them and understand the challenges we have to deal with in order to represent their documents for clustering tasks. It is worth defining category or class as the set of related documents associated under the same group in the ideal solution of a dataset. In contrast, we use the term cluster to refer to a set of documents grouped by a clustering algorithm. Moreover, we refer to domain and subdomain as abbreviations of web domain and web subdomain. Once we have made clear these definitions, we focus our analysis in the following aspects:

- The number of documents belonging to each category, in order to see whether collection categories are balanced or not. Unbalanced datasets are usually more difficult to cluster due to the bias that bigger categories introduce.

- The differences among category themes. These differences could affect to the divergence of the vocabularies used in the documents of each category. The greater the divergence, the easiest should be the clustering process, as categories will be represented by means of different terms.

- The domain distribution within and among categories. When different documents belong to the same domain, then they could share some features regarding the style (structural or formatting similarities). Sometimes documents from the same domain can share some terminology associated to that domain, as text corresponding to menus, headers or footers. However, this terminology is usually not related with the category of that documents. For instance, documents belonging to Wikipedia contain different menus in common for editing, login or navigation tasks. This may affect the clustering depending on the number of documents belonging to the same domain.

When this domain-related terminology is shared within the same category, it could help the clustering, although it is not directly related with the theme of the documents. In this case, not removing this domain-related terminology could even help the clustering. On the contrary, when the documents sharing the same domains are distributed among different categories, then it is important to remove this domain-related terminology, because it could lead to find relationships among documents belonging to different categories, though those documents do not share the same themes. In general, we talk about a good distribution of domains or well distributed domains to refer to uniformly distributed domains among categories.

- The proportion of sets of documents belonging to a same domain with respect to the set of documents belonging to domains that appears only once in the collection. This way, we show the influence of the origin of the documents on the dataset, that is, whether the previous point can have any effect in the collection or not. If most of the documents belong to domains appearing only once in the collection, then there is no possible influence. Yet, this influence grows as the number of documents that share the same domain increases. This aspect is clearly related with the previous one and serves to measure its effect on the collection.

- The frequency of the terms in a document. In general, we should find few terms with high frequencies and many terms with low ones. Variations in this trend may be employed to differentiate among collections. We analyze not only term frequency on the whole document, but also term frequency in titles and emphasis, which correspond to the criteria we use to represent web pages in this thesis.

3.2 Banksearch Dataset

This dataset was compiled by Sinka and Corne (2005), being designed to be a reference collection oriented to unsupervised document clustering. Thus, the authors proposed it as a benchmark dataset for evaluating different techniques. Some previous works using this dataset are: Fresno (2006); Liu and Lu (2008); D'hondt et al. (2010); Matharage et al. (2011) and Liu et al. (2011).

3.2.1 Category Analysis

Banksearch dataset contains a total of $11,000$ documents divided into 11 categories of equal size and two hierarchy levels: 10 main categories at the same level and another one parent of two of them. These categories are divided in four associated themes, as shown in table 3.1.

Table 3.1: Banksearch categories and their associated themes.

Dataset Id	Dataset Category	Associated Theme
A	Commercial banks	Banking and finance
B	Building societies	Banking and finance
C	Insurance agencies	Banking and finance
D	Java	Programming languages
E	C/C++	Programming languages
F	Visual Basic	Programming languages
G	Astronomy	Science
H	Biology	Science
I	Soccer	Sport
J	Motor sport	Sport
K	Sport	Sport

We use the 10 main categories—A to J—, corresponding to 10,000 documents. We removed K category, because it was created to be more general than I and J, containing documents about sports not included in I and J, since we are interested in plain clustering with well differentiated categories without overlapping among them. K is a miscellanea category related to sports that would introduce noise in the process. We also removed some documents because of HTML parsing errors[1], so the final number of documents we used in our experiments was 9,897. Figure 3.1 shows the distribution of documents among categories in Banksearch dataset. The slight difference among categories is due to the aforementioned parsing errors.

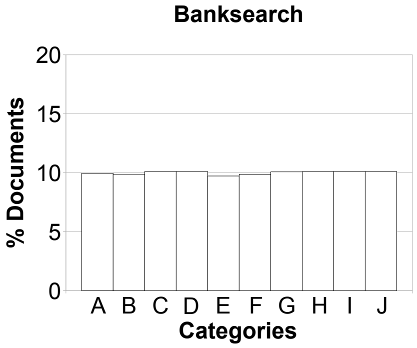
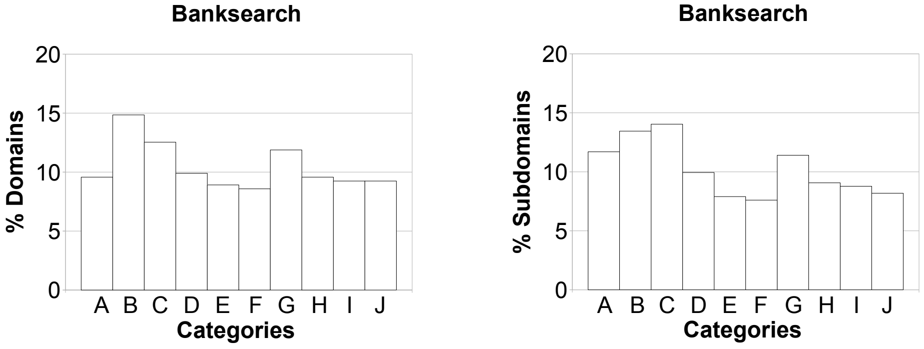


Figure 3.1: Banksearch: Documents in each category.

At this point, we have seen that Banksearch has clear thematic blocks and well balanced categories. In terms of domains, these documents belong to 303 unique

[1] Encoding errors, files that were incorrectly downloaded, etc.

domains and 342 unique subdomains. In this collection, unique domains and subdomains are uniformly distributed among categories. This is a consequence of the restrictions imposed in the creation of the dataset, in particular due to the crawling process where they removed websites containing fewer than 10 web pages (Sinka and Corne, 2005). On the one hand, figure 3.2 shows the proportion of domains and subdomains for each category. On the other hand, figure 3.3 shows the percentage of domains and subdomains that appears in more than one category. Thus, the figures present low dispersion of domains among categories and well balanced number of documents per category. This is a consequence of the above mentioned restrictions in the creation of the dataset. These characteristics should ease the clustering task on Banksearch, as we can expect a reasonable divergence among the vocabularies of the categories.

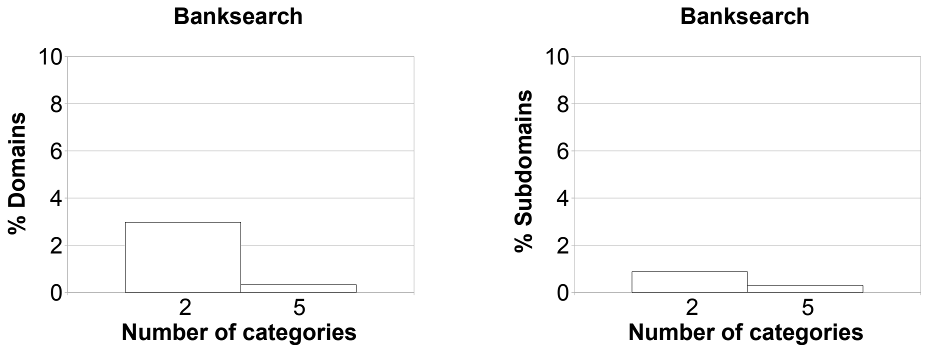


(a) Domains

(b) Subdomains

Figure 3.2: Banksearch: domain and subdomain distributions with respect to dataset categories.

Nevertheless, we do not know at what extent domains could have an influence on categories. Aiming to clarify this, figure 3.4 shows the number of repetitions of each domain or subdomain in the collection along the x axis. A document belonging to a domain is what we call here a domain repetition, so a domain having two repetitions means that there are two documents in the collection belonging to that domain. The same occurs with subdomains in the corresponding figure 3.4b. Then, the white bars represent the percentage of domains or subdomains—depending on the figure, 3.4a or 3.4b—having n documents in the dataset. The gray bar shows the percentage of documents belonging to these domains. For example, if we look at the bars corresponding to 42 repetitions in figure 3.4b, we can interpret that more or less a 7% of subdomains appears 42 times in the collection, that is, there is a 7% of domains that contributes with 42 different documents to the collection each, and the total of documents contributed by that 7%



(a) Domains **(b)** Subdomains

Figure 3.3: Banksearch: domains and subdomains that appears within different categories.

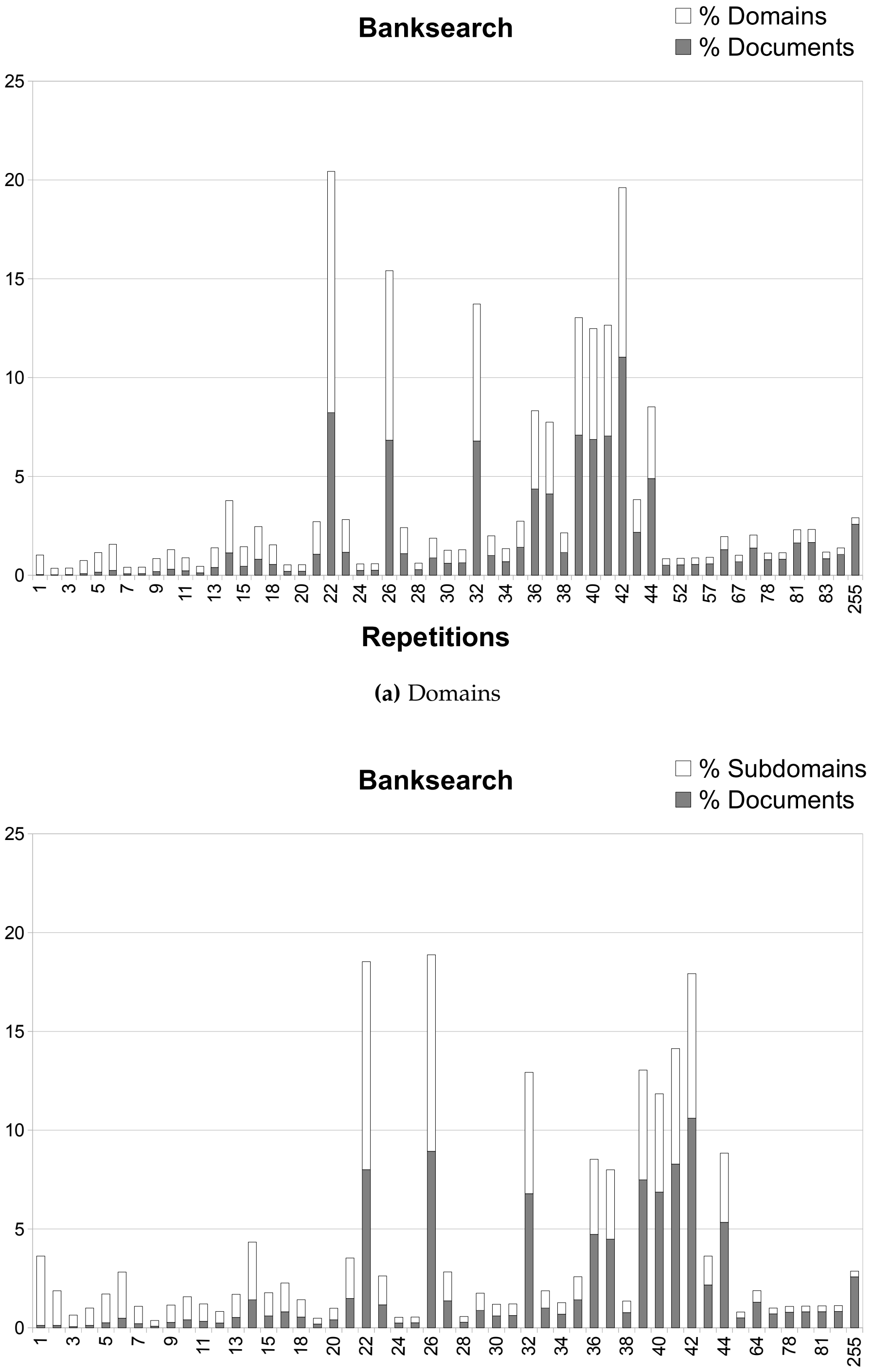
of domains represents about 11% of the total number of dataset documents. It should be noted that the right part of the figure (over 44 repetitions) contains few values, so we group them in order to simplify their visual representation.

Looking at the figure 3.4 different issues emerge. First, the number of documents belonging to domains and subdomains that appears only once in the collection is very low. This also implies that for most of the documents, there is at least another one belonging to the same subdomain in the collection. To give the concrete number, 99.11% of documents belong to domains with 10 or more repetitions, which corresponds to 94.06% of total domains. The 80% of the documents belongs to domains having between 20 and 50 repetitions, for a total of 79% of domains.

Summarizing everything up to now, Banksearch has a good balance of documents and domains per category. Within each category, it has mostly groups of documents belonging to domains and subdomains that are not present in any other categories. Moreover, these groups of documents represent almost the whole collection. The rest of the documents belong to domains or subdomains that provide less than 9 documents each to the collection. Because of these reasons, we conclude that this collection could ease the clustering task as documents within each category are probably very similar among them—attending to the categories defined by the authors of the dataset—, and different from documents belonging to other categories.

3.2.2 Term Distribution Analysis

In this section we analyze the term distributions in the following criteria we use in the combination: term frequency in the whole document, term frequency in the



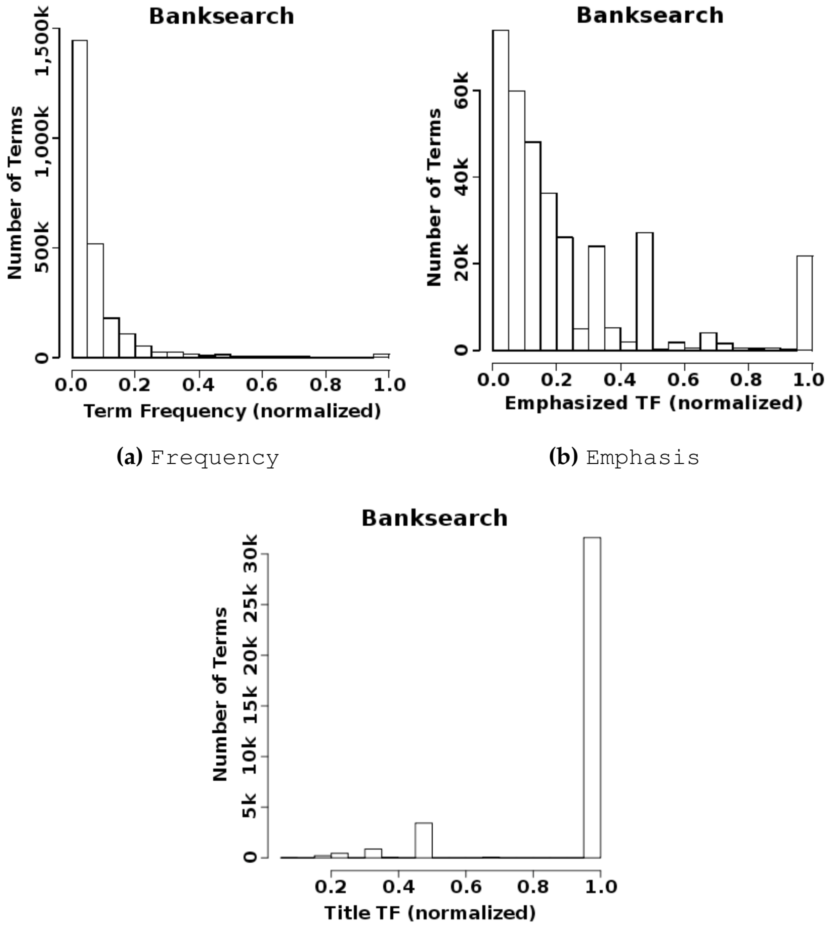
(a) Domains

(b) Subdomains

Figure 3.4: Banksearch: percentage of domains and subdomains that appears *n* times (x axis) in the dataset associated to the number of documents belonging to these domains.

title of the document, and term frequency in emphasized text segments. These distributions are employed in this dissertation to find common patterns or differences among collections, that could be used to characterize them.

Figure 3.5 shows term frequencies in each criterion normalized to the maximum term frequency value in the corresponding criterion. Each bar represents the amount of terms having a concrete normalized frequency for the corresponding criterion. From the point of view of how the terms are used in each criterion, Banksearch shows term distributions where few terms have high frequencies and most of the terms have low ones for `frequency` in the whole document and for `emphasis`, while `title` shows a different behavior.



(a) `Frequency`

(b) `Emphasis`

(c) `Title`

Figure 3.5: Banksearch: normalized frequencies of terms, emphasized terms and title terms in Banksearch dataset.

Focusing on Subfigure 3.5a, we can see a long tail of terms with high frequency in the document. Due to the scale imposed by the normalization process, terms having low frequency values should be considered the same because all of them are far away from document maximum value and there is no way to difference among their representativeness due to the high term density in that area. At the same time, there are few terms with high frequencies, probably because document maximum values are far away from the rest.

Shifting our attention to Subfigure 3.5b that shows normalized frequencies for emphasized terms, we see a variation on the long tail compared to the previous criterion. We can see a larger number of terms having greater emphasis term frequencies with respect to Subfigure 3.5a. When authors emphasize a word, they are explicitly establishing the importance of that word over other words in the document. So, it is normal finding higher frequencies for emphasized terms than for the whole set of terms, because the authors select them explicitly, introducing a bias towards term importance. This fact is reflected in the figure, where the bars corresponding to values like $1/2$ and $1/3$ are larger than in the `frequency` case. This points out that maximums are lower for emphasis, and then the probability of having frequencies that are normalized to these values increases. We also believe that this distribution can vary more than the others as it depends directly on author's opinion about which are the important words and their particular style using emphasis to highlight them.

Finally, Subfigure 3.5c shows normalized frequencies for title terms. In this case, as titles are usually short text strings, there are not much different possible values and most of the terms coincide with the maximum used for normalization. Naturally, most of the maximum values will be low—1 or 2—, so we believe this is the usual situation one can expect to find for frequencies of title terms in most of the collections.

Summarizing, `frequency` and `emphasis` criteria show a decreasing number of terms as the frequency values increase. While `frequency` shows a clear queue following a shape that could correspond to a distribution of exponential type, `emphasis` shows a more relaxed decreasing area, with more intermediate values. We believe that this behavior for `emphasis` could be due to these terms are explicitly highlighted by the authors, as explained above. Lastly, `title` criterion is biased by the short length of titles and shows a different distribution where few different values are possible.

3.3 WebKB Dataset

This dataset was compiled by Craven et al. (2000) to be employed in tasks related to making text information on the web available in computer-understandable form, enabling sophisticated information retrieval and problem solving. These web pages were collected from computer science departments of various Universities: 4, 162 web pages from Cornell, Texas, Washington and Wisconsin and 4, 120 additional pages gathered from other universities. Some previous works using this dataset are Xu and Zuo (2004); Tang et al. (2005); Mohamed et al. (2006); Fresno (2006); Huang et al. (2006) and Ozcan et al. (2012).

3.3.1 Category Analysis

WebKB contains a total of 8, 282 classified web pages divided into 7 categories of different sizes (see table 3.2). *Other* category contains different kind of pages like publication lists, vitae or research interest. Moreover, its size is much bigger than the rest of categories. Because of this, in this thesis we decided to remove this category to avoid the noise it would produce in the clustering process. Thus, the resulting dataset consists of 6 categories and a total of 4, 518 documents.

Table 3.2: WebKB categories and their sizes.

Dataset Category	# of Documents
Student	1,641
Faculty	1,124
Staff	137
Department	182
Course	930
Project	504
Other	3,764
Total	8,282

In order to show this imbalance more graphically, figure 3.6 represents the proportion of documents in each category in the resulting dataset. The figure shows how the proportions go from 3% of documents in the smallest category, to 36% of documents in the biggest one. Besides, categories could contain very heterogeneous documents, which could lead to the use of similar terminology among them.

Precisely, an important difficulty of this dataset, probably the most problematic one, is the heterogeneity of documents within the categories. For instance, web pages with references to the same subject, let's say Java programming or Modern Art, could be found in different categories like *Student*, *Course*, *Department* or *Faculty*. This supposes an important difficulty to find a clustering that corresponds with the categories in which this dataset is divided.

In terms of domains, there are 189 unique domains and 449 unique subdomains in this dataset. Depending on the category, these could contain from about 20% to 85% different domains as shown in figure 3.7. There is no direct relation between the number of documents in a category and the number of different do-

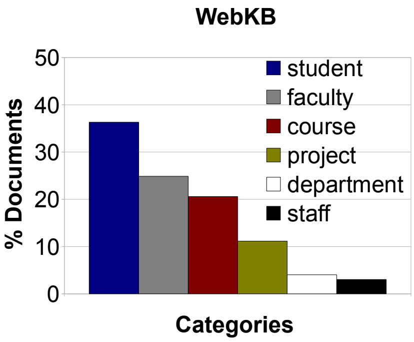
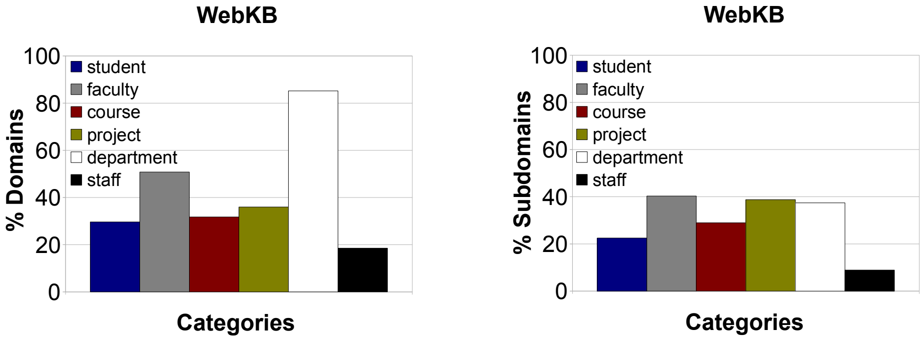


Figure 3.6: WebKB: Documents in each category.

mains that contribute documents to that category, e.g., *Department* category is the second category with smallest number of documents, but these belong to more than 85% of domains.



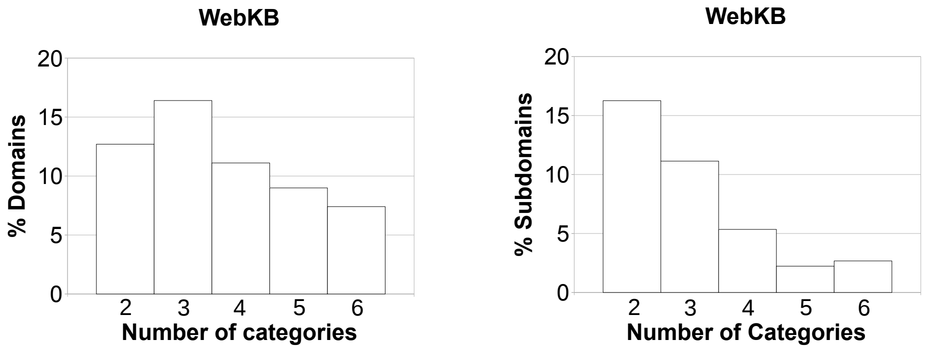
(a) Domains

(b) Subdomains

Figure 3.7: WebKB: domain and subdomain distributions with respect to dataset categories.

In addition, figure 3.8 shows that almost 57% of domains are distributed among several categories, in some cases even among all of them. Looking at subdomains (figure 3.8b) occurs approximately the same: about 38% of subdomains are spread among different categories.

Following the same assumptions we made to introduce this chapter in Section 3.1—documents belonging to the same domain, or better, to the same subdomain are good candidates to share some terminology that has nothing to do with the theme of the documents, as they could share menus, headers, footers, etc.— and taking into account than documents collected from the same domain belong to



(a) Domains

(b) Subdomains

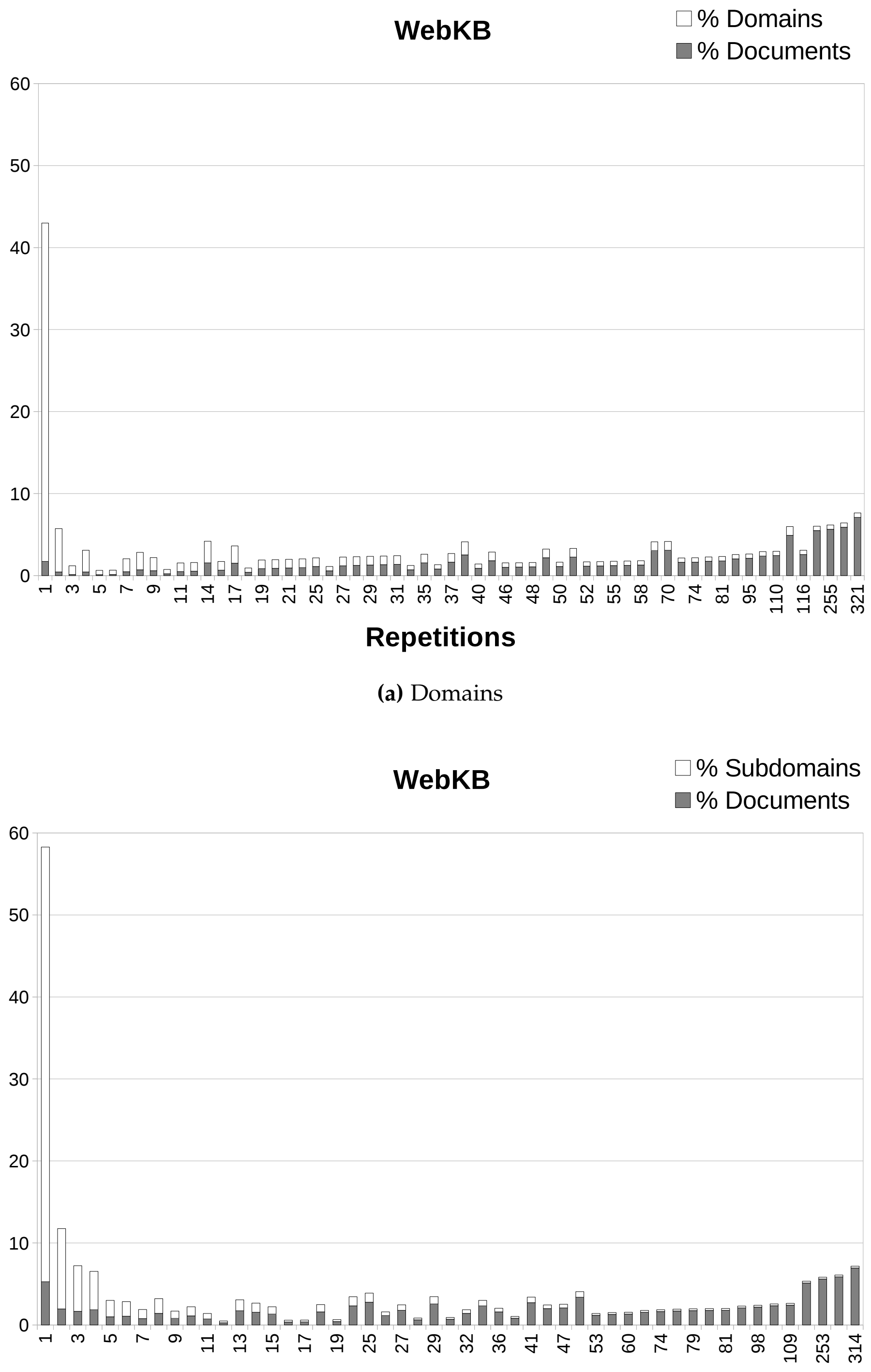
Figure 3.8: WebKB: domains and subdomains that appears within different categories.

different categories, then the terminology associated to the web domain but not related with the theme of the document would be spread over the whole dataset. In terms of clustering tasks, this terminology should be removed because, otherwise, it will increase the difficulty of finding the right categories and, therefore, proper clusters.

On the other hand, the distribution of repeated domains among categories will affect the dataset depending on the percentage of documents these domains represent. In figure 3.9 we can see that most of domains—about 41%—are unique, i.e., they have only one repetition in the collection. Nevertheless, this fact hardly influences the dataset because these domains represent less than a 2% of documents. The case of subdomains is similar: about a 53% of subdomains appearing only once in the collection and representing about a 5% of documents. In addition, domains with less than a 10 repetitions—about 57% of domains—correspond to less than a 5% of documents, so about a 95% of collection documents belong to domains appearing 10 times or more. Looking at subdomains, there are about a 83% of documents belonging to about a 18% of total subdomains and having more than 10 repetitions.

To sum up, there are many domains and particularly subdomains with low representativeness and few of them that contribute most of the documents to the collection. We refer to representativeness as the number of documents that a particular domain contributes to the collection. The greater the number of documents a domain contributes, the higher its representativeness is.

Therefore, there is a considerable influence of domains and subdomains in this collection. Together with the distribution of domains among categories, this fact emphasizes the importance of removing web domain associated terminology



(a) Domains

(b) Subdomains

Figure 3.9: WebKB: percentage of domains and subdomains that appears *n* times (x axis) in the dataset associated to the number of documents belonging to these domains.

mentioned above.

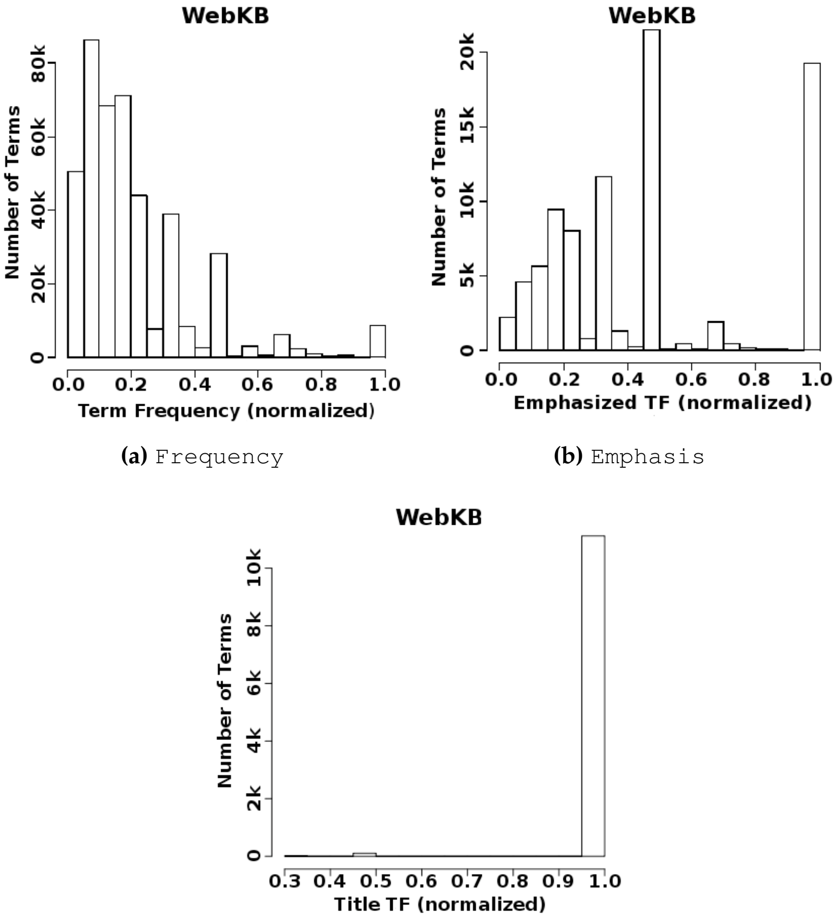
In contrast to Banksearch, few domains are repeated within the same category. Concretely, as we saw before, a 41% of domains appear only once in the collection, so that a 59% of domains appear more than once. On the other hand, we also found a 57% of domains appearing in more than one category. Thus, as each and every domain appearing in more than one category must belong to the group of domains appearing more than once in the dataset, it is a fact that only about a 2% of domains appear more than once in the same category.

3.3.2 Term Distribution Analysis

Taking a look at Figure 3.10 and comparing WebKB results with other collections like Banksearch (see Section 3.2.2), it is possible to realize at what extent the way of creating a collection can influence its composition.

Subfigure 3.10a shows a tail of terms with high frequencies, but WebKB seems to have a different distribution under 0.5 value than Banksearch. In WebKB there are more terms with intermediate frequencies, that is, not as far from document maximum values than in Banksearch. Thus, the long tail we found in Banksearch from frequencies above 0.2 is not clear here, since the bars in the distribution do not always decrease with respect to the immediately previous one. However, if we divide the space between 0.2 and 1 in quartiles, we found almost the exact same values for Banksearch and WebKB. Then, the number of terms between 0.2 and 1 seems to follow a similar proportion. In the case of WebKB the tail could be not so clear due to smaller maximum values per document, that lead to a smaller set of possible values after normalization. With less possible values, we would find the term frequencies more concentrated in concrete points, like in Figure 3.10a. So, tails look different in the figures, yet splitting them in quartiles shows that they are not so different at certain degree of detail.

But actually, Subfigure 3.10b shows the most surprising data. In this case WebKB dataset shows a totally different distribution with respect to Banksearch (see Subfigure 3.5b), which implies that emphasis has been used by web page authors in a different manner. It is also worth noting the size of the bars corresponding to frequencies of 1/2 and 1/3, that are larger than the rest. As we mentioned in the case of Banksearch, we believe that the reason are the low maximum values, that lead to a limited number of possible frequency values. However, in WebKB, the number of emphasized terms grows with the frequency for values from 0 to 0.5. This demonstrates the fact that in the case of emphasized terms, their frequency distribution can be totally different depending on how authors use them. We believe this distribution could be due to a more restricted and meaningful use of emphasis. More than that, this is an example of how the distribution of frequencies in a collection does not always follow the Zipf's law, where the most



(a) `Frequency`

(b) `Emphasis`

(c) `Title`

Figure 3.10: WebKB: normalized frequencies of terms, emphasized terms and title terms in WebKB dataset.

frequent term will occur approximately twice as often as the second most frequent term, three times as often as the third, and so forth. In the particular case of emphasis, the reason is the need of an explicit act of the author to highlight words, that both restricts the use of emphasis and provide visual information about the importance of a word.

Lastly, there is not much new to say about the distribution of title frequencies shown in Subfigure 3.10c, because it is similar to Banksearch dataset that was already explained in Section 3.2.2.

3.4 Social ODP 2k9 Dataset

In order to explore how anchor texts could be employed to enrich web page representation for clustering, we needed to employ a recently crawled collection, in such a way that it was able to find other web pages with hyperlinks to collection documents. We decided to use the dataset Social ODP 2k9 (SODP) presented in Zubiaga et al. (2009b). SODP consists of 12,616 documents retrieved from social bookmarking sites. They were classified by extracting the category for each URL from the first classification level of Open Directory Project (ODP[2]). The collection and its categories were presented as a gold standard with no overlapping among categories. We removed documents that caused problems with the HTML parser, resulting in 12,148 documents we use in this thesis.

In addition to the documents themselves, we collected the anchor texts corresponding to a maximum of 300 unique inlinks per each document in the collection. We queried Google search engine for links pointing at collection pages. So, for each web page in SODP we also gathered the anchor texts corresponding up to 300 other pages that had a link to the dataset page. To give an idea of the information retrieved, 2,704 web pages have less than 50 inlinks, 4,717 have less than 100, so the rest, approximately 60% of collection pages, have more than 100 inlinks.

3.4.1 Category Analysis

SODP collection is divided in 17 unbalanced categories, having from 39 to 3,217 documents each (see table 3.3).

There is a clear bias towards *Computers* category. This fact is even more important given the high number of categories: almost 26% of documents belong to *Computers* category, so the other 74% is divided among the remainder 16 categories. Nevertheless, each category has different number of documents. Figure 3.11 shows the proportion of documents in each category. This kind of document distribution among categories makes the clustering task more difficult. Terminology from bigger categories has more influence than that from smaller ones. It will favor the division of documents belonging bigger categories making more complicated to find smaller ones.

Looking at the domain and subdomain distributions shown in figure 3.12, we find that these distributions are almost the same than document distribution. This points out the small number of documents belonging to the same domain in the collection. In other words, most of domains and subdomains are unique in this dataset. To confirm this fact, there are exactly 10,240 different domains and 11,242 different subdomains, both numbers really close to the total number

[2] `http://www.dmoz.org/`

Table 3.3: SODP categories and their sizes.

Dataset Id	Dataset Category	# of Documents
1	Reference	558
2	Society	923
3	Business	494
4	World	1,804
5	Regional	1,295
6	Arts	1,060
7	Kids and Teens	383
8	Shopping	479
9	Computers	3,217
10	Home	248
11	Games	267
12	Science	736
13	Recreation	299
14	News	138
15	Sports	102
16	Health	106
17	Adult	39
Total		12,148

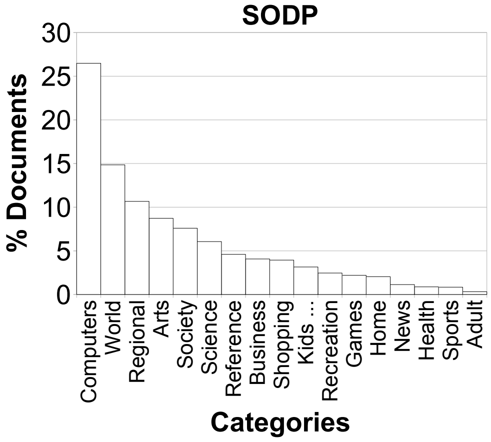
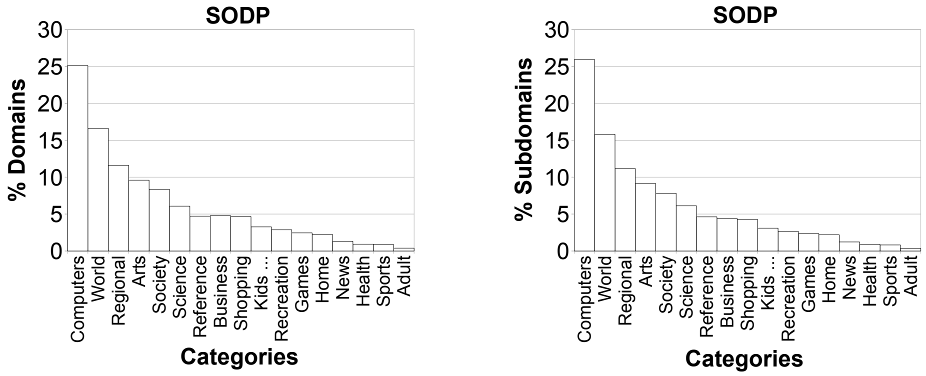


Figure 3.11: SODP: Documents in each category.

of documents in the collection, that is 12, 148.

Clearly, in this case, domains appearing more than once in the collection are not representative enough and they will not have any influence in the collection. Figure 3.13 shows domains and subdomains appearing in more than one category. Although there are domains and subdomains belonging to many cat-

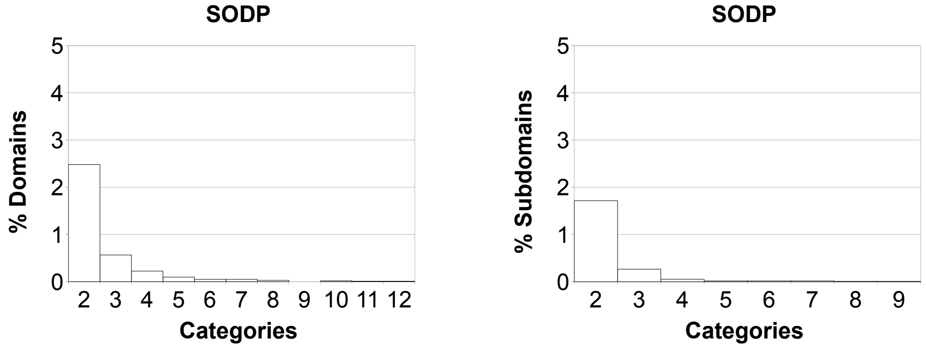


(a) Domains

(b) Subdomains

Figure 3.12: SODP: domain and subdomain distributions with respect to dataset categories.

egories, their number is not representative enough to produce any effect on the collection terminology.



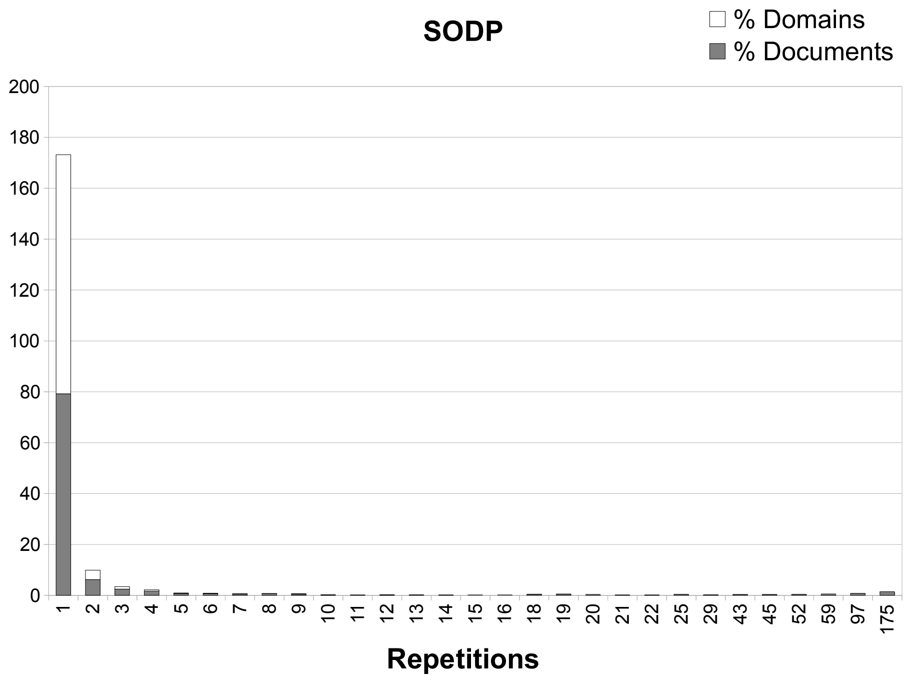
(a) Domains

(b) Subdomains

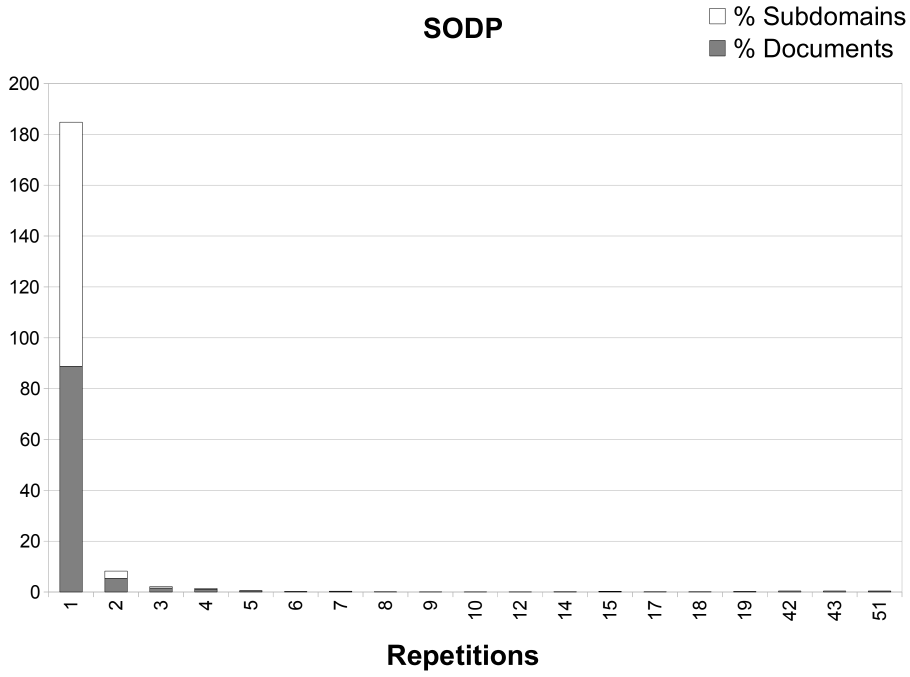
Figure 3.13: SODP: domains and subdomains that appears within different categories.

Finally, with the help of figure 3.14 we can easily check that most of the documents belong to domains and subdomains appearing only once in the collection. In the case of domains (see figure 3.14a), about 79% of documents corresponds to 94% of domains that are unique in the dataset. Looking at subdomains (see figure 3.14b), 89% of documents belong to 96% of subdomains that appears only once in the collection.

On the whole, this dataset is clearly unbalanced in terms of the number doc-



(a) Domains



(b) Subdomains

Figure 3.14: SODP: percentage of domains and subdomains that appears *n* times (x axis) in the dataset associated to the number of documents belonging to these domains.

uments classified in each category. In addition, domains are not representative enough to introduce any kind of bias because of the small number of domains appearing more than once in the collection and their associated number of documents.

3.4.2 Term Distribution Analysis

Attending to Figure 3.15, this dataset looks very similar to Banksearch (see Figure 3.5). Thus, we find again the long tail for term frequency in the document (see Subfigure 3.15a) and a less pronounced curve for emphasis (see Subfigure 3.15b), where again there is bias about 0.5 due to the different use of emphasis with respect to term frequency in the whole document. Besides, frequency of title terms (see Subfigure 3.15c) has the same behavior of previous datasets, as we could expect.

Consequently, the conclusions and ideas we set forth before in Section 3.2.2 are perfectly valid here.

3.5 Wikipedia Animal Dataset

Wikipedia Animal Dataset (WAD) was presented in Paukkeri et al. (2012)[3]. It is composed of documents about animals gathered from Wikipedia, a multilingual, online and free-content[4] encyclopedia that follows a collaborative model allowing anyone edit its contents. Different from the previous datasets, WAD is hierarchically classified.

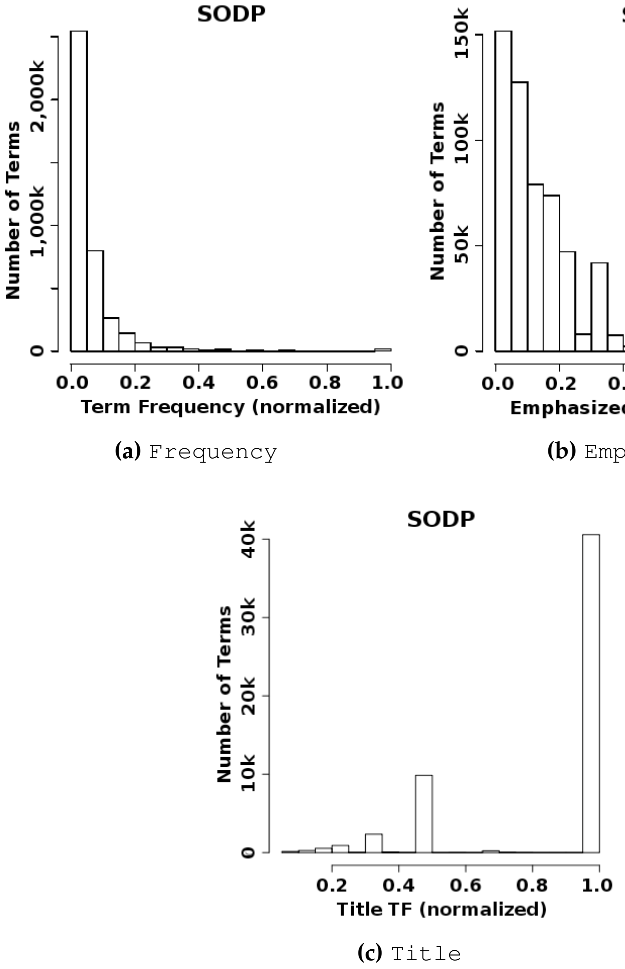
The data were collected manually. Information about categories was used to collect groups of animals with at least 3 articles fulfilling the restriction of containing more than 100 words each. Taking advantage of the multiple languages of Wikipedia contents, documents were collected in two of them, English[5] and Spanish[6], in order to employ this dataset to explore the behavior of different web page representations with different languages. Each article corresponds to a single animal and the whole dataset is composed of 166 articles in each language, for a total of 332 Wikipedia articles, what makes WAD a bilingual comparable corpus. It is worth mentioning that we use directly the HTML pages from Wikipedia in this thesis.

[3] WAD is available for research purposes at `http://nlp.uned.es/~alpgarcia/wad.php`

[4] `http://en.wikipedia.org/wiki/Free_content`

[5] `http://en.wikipedia.org`

[6] `http://es.wikipedia.org`



(a) Frequency

(b) Emphasis

(c) Title

Figure 3.15: SODP: normalized frequencies of terms, emphasized terms and title terms in SODP dataset.

3.5.1 Category Analysis

On the one hand, the WAD is different from the other cases presented previously in this chapter, because the whole collection belongs to Wikipedia domain. Thus, domain and subdomain analysis have no sense for this dataset.

On the other hand, articles were classified by following the scientific classification schema extracted from the information box available in the corresponding Wikipedia articles. The reference ontology was also manually built using this information. The articles gathered for each language correspond to the same animals, so the reference ontology is exactly the same for both languages.

The hierarchy was slightly simplified to three levels of the scientific classification. These levels are: Kingdom, Class, and Order. For example, Figure 3.16 shows a part of the reference ontology.

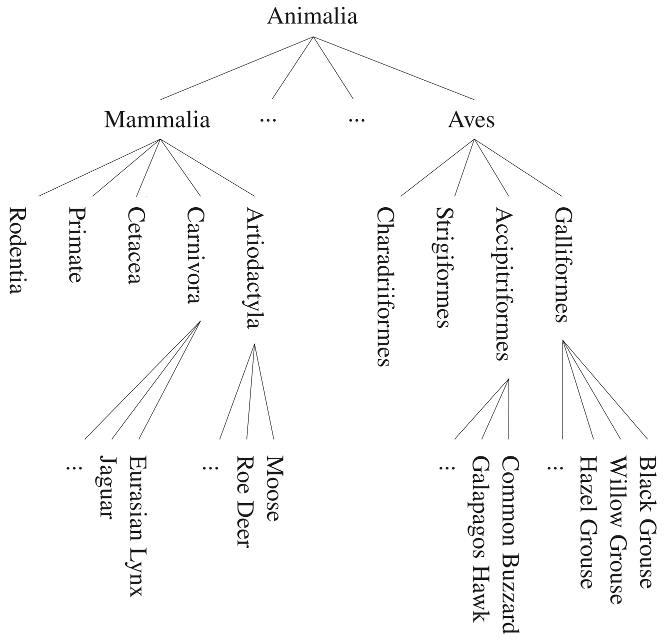


Figure 3.16: A part of the reference ontology, from Paukkeri et al. (2012). For example, Jaguar belongs to Order *Carnivora*, Class *Mammalia* and Kingdom *Animalia*.

This simplification put together Families, Subfamilies and Species in the third hierarchy level, making no distinction among them. The hierarchy begins with the Kingdom *Animalia*, that is the root concept and comprises the whole set of animal documents. Then we find 4 first level concepts, corresponding to animal Classes: *Actinopterygii* (Ray-finned fishes), *Aves* (Birds), *Mammalia* (Mammals), and *Reptilia* (Reptiles). Each of these classes are divided in several Orders, for a total of 17 different animal Orders. Finally, we found 166 concepts corresponding to the animal documents in the third level of the ontology. Then, an example of how to read the schema would be: *Moose* belongs to the Order *Artiodactyla*, within the *Mammalia* Class of Kingdom *Animalia*.

The document collection was not modified to produce groups of equal size in order to keep the original distribution of data as it is in Wikipedia (Paukkeri et al., 2012). The sizes of the groups on the first and the second level are shown in Table 3.4. As we already mentioned above, the first level corresponds to animal Classes, located in the first table column, that are divided in different Orders on the second level, that corresponds with the second table column. Finally, Wikipedia articles—that is, the animals—are the leaf nodes, located on the third level of the reference ontology and in the last table column.

Table 3.4: Wikipedia articles from different animal Classes (in bold text) and Orders. The number of articles within each Class is shown below the corresponding Class name. The whole set corresponds to the Kingdom *Animalia*.

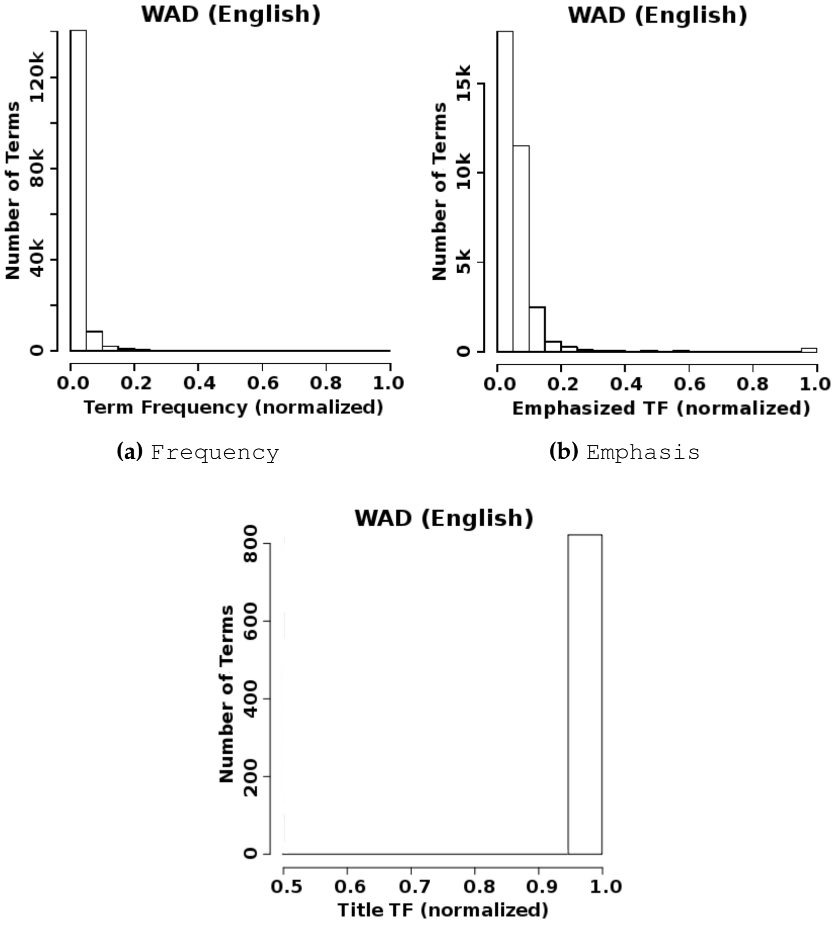
Class	Order	# articles
Actinopterygii **(26 articles)**	Cypriniformes	5
	Salmoniformes	5
	Gadiformes	4
	Tetraodontiformes	3
	Perciformes	9
Aves **(34 articles)**	Charadriiformes	8
	Galliformes	5
	Accipitriformes	13
	Strigiformes	8
Mammalia **(80 articles)**	Artiodactyla	16
	Carnivora	37
	Cetacea	11
	Primates	8
	Rodentia	8
Reptilia **(26 articles)**	Crocodylia	8
	Squamata	15
	Testudines	3
Total		**166**

Looking at the number of articles in each Class and Order, we can see the differences among them. For example, *Mammalia* class show a clear imbalance over the rest of classes. The same occurs with the order *Carnivora*. In conclusion, WAD is clearly unbalanced in all hierarchy levels, so finding a hierarchical clustering solution taking the Wikipedia articles as input is a challenging task.

3.5.2 Term Distribution Analysis

First and different from the other datasets, WAD collection was compiled in two different languages. We show here the distributions of each language separately: Figure 3.17 for English and Figure 3.18 for Spanish.

Nevertheless, results look very similar in both languages in terms of the term distributions and, despite of the smaller size of this collection, they also look very similar to Banksearch and SODP datasets. It is worth noting that in Subfigure 3.17c all the terms have the same frequency value, that is 1, and because of this the histogram just show one bar covering the whole area. The main difference we can observe between languages is the smaller number of terms in Spanish, due to English documents are longer and more detailed than Spanish ones. This



(a) `Frequency`

(b) `Emphasis`

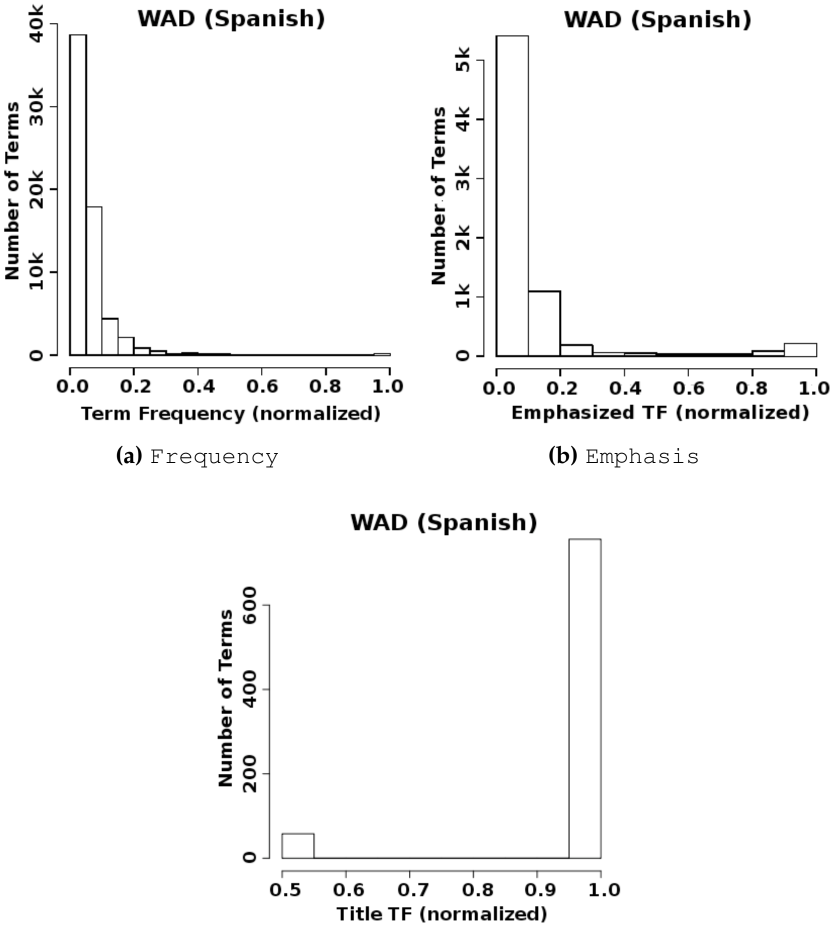
(c) `Title`

Figure 3.17: WAD (English): normalized frequencies of terms, emphasized terms and title terms in English WAD dataset.

difference is not representative to compare different datasets because they contain totally different documents, but it is important in WAD because documents represent the same concepts in two languages. Thus, the number of terms used to describe the concepts could affect the representation and, by extension, the clustering.

3.6 Conclusion

In this chapter we have presented the main characteristics of the datasets we have employed in the experimentation carried out for this thesis. We have also talked about the influence of dataset specific aspects that can help or hinder web page



(a) `Frequency`

(b) `Emphasis`

(c) `Title`

Figure 3.18: WAD (Spanish): normalized frequencies of terms, emphasized terms and title terms in Spanish WAD dataset.

representation and clustering.

Consequently, the idea behind this chapter is reflecting the degree of challenge of each dataset. The analysis we have carried out shows that Banksearch should be the easiest collection to group. Banksearch has very good domain and subdomain distributions within and among categories. In addition, it has a good balance among categories, which are also composed of homogeneous documents. On the other side, we have SODP dataset, with big differences in the number of documents belonging to each category and hardly any domain or subdomain in common among documents, regardless whether they belong to the same category or not. It has a clear bias towards *Computers* category, the biggest one that could lead the algorithm to find subgroups of this category instead of finding the smallest ones in the gold standard. In an intermediate point we find WebKB, whose main difficulty come from the way in which their documents were collected from a restricted environment (University domain, mainly from 4 Universities) and the heterogeneity of its documents within its categories. The last one, WAD, is very specific in the sense that the terminological domain is clearly bounded. For this reason, we believe that the initial difficulty derived from its imbalance among categories could be alleviated thanks to the more concise terminology used to write its documents.

Apart from this, we have studied the frequency distributions of terms in different criteria we use to represent pages: `frequency`, `emphasis` and `title`. As a result of this study, we have found that most of the datasets introduced here follow similar distributions. For terms in the whole document, they tend to follow distributions approaching to Zipf's law. At the same time, the use of emphasis is more restricted and it highly depends on authors' writing style. This way, term distribution for `emphasis` may vary from one collection to another. We have seen most of the collections tends to show a decreasing term distribution for emphasis, with a more relaxed curve than term frequency in the whole document, because as a consequence of its different use, the maximum values are much smaller for emphasis than for term frequency. Nevertheless, in WebKB we have found a good example of a totally different distribution for `emphasis`, that has nothing to do with a power law. Actually, the distribution of term frequency in the whole document corresponding to WebKB dataset is not approaching to these kind of distributions as much as the distributions from other collections, though we found a similar proportion of terms in the queue of the distribution. Finally, distributions of term frequency in titles are similar in all collections we have studied. Titles are usually short texts, and therefore term frequency cannot vary too much. Then, most of the terms appears the same number of times than the maximum used for normalization, because most of the maximum values are low, that is to say 1 or 2. So, as we stated before in this chapter, we believe this is the usual situation one can expect to find for frequencies of title terms in datasets composed of web pages.

The differences among the distributions corresponding to `frequency` criterion and particularly to `emphasis` criterion show that different collections can reflect a different use of each criterion. Thus, the importance of a given term in a concrete criterion could depend on the general use of that criterion in the particular dataset. For instance, WebKB `emphasis` distribution could suggest a more restricted utilization of this criterion than in Banksearch or SODP, since the emphasized terms in WebKB have higher frequencies. Then, the information capture process could be adapted to take this fact into account. The same term frequency of an emphasized term in WebKB could imply more importance for that term than in Banksearch or SODP, because we consider the use of `emphasis` more restricted in WebKB. Thus, term distributions could help establish the importance of a term in a particular criterion, regarding the use of that criterion in the dataset. This would allow to automatize the information capture process at the same time this process is adjusted to the concrete dataset, since we have seen different distributions among datasets. Therefore it is interesting to analyze whether this kind of adjustments influence the clustering and their impact on the results.

4

Fuzzy Combinations of Criteria: An Application to Web Page Representation for Clustering

Web pages are usually written in HTML, offering useful information to try to select the most important features to represent them. In this chapter we investigate the use of a fuzzy combination of criteria by means of a fuzzy system to find those important features. We start our research from a self-content document representation called Fuzzy Combination of Criteria (FCC) that relies on term frequency, document title, emphasis and term positions in the text. Next, we analyze its drawbacks, proposing and applying some ideas to overcome them. We also explore different methods to include contextual information in the representation. As a result, we propose a new fuzzy combination of criteria based on our findings.

The chapter is organized as follows. First, in Section 4.1 we briefly introduce the chapter. In Section 4.2 on page 106 we describe in detail FCC method to represent web pages. Next, the experimental settings employed for evaluation are detailed in Section 4.3 on page 114. In Section 4.4 on page 116 we explore different dimension reduction techniques and propose a new one, based on the weighting function itself. Then, FCC representation method is analyzed in Section 4.5 on page 124 and two alternatives to overcome its drawbacks are proposed and evaluated. Section 4.6 on page 132 is focused on the study of new criteria for the combination beyond the pages themselves. Finally, a robust evaluation of our proposal is performed in Section 4.7 on page 136 and our conclusions and findings are summarized in Section 4.8 on page 139.

4.1 Introduction

As we saw in Chapter 1, document representation is an essential step in web page clustering. Throughout Chapters 1 and 2 we showed that the most common approach is trying to capture the importance of the words in the document by means of term weighting functions, most of them based on the VSM. Among them, TF-IDF is one of the most widely used. It has become a de facto standard applied to text documents and web pages, among others.

We also stated one of the initial hypothesis of the present work: a good representation should be based on how humans have a quick look at documents in order to extract the essential information, that is, to determine the words that better represent document contents. We usually search for visual clues used by authors to capture our attention as readers. These visual clues are included in some kind of documents as format information, e.g., in HTML tags. However, representations like TF-IDF, based only on word frequencies, directly ignore this information.

Thus, the HTML tags provide additional information about those visual clues that can be employed to evaluate the importance of document terms in addition to term frequency. Regarding the way of combining different criteria within the VSM, we talked about linear combinations of criteria (Wang and Kitsuregawa, 2002; Fresno and Ribeiro, 2004; Liu and Liu, 2008; Hammouda and Kamel, 2008), allowing to set different weights for each criterion. We previously explained the main drawback of this kind of representations: when a term is important in a single criterion, e.g. in title, the corresponding component will have a value which will always be added to the importance of the term in the document, regardless of the importance of the rest of the components. From our point of view, it is questionable whether a human reader would evaluate the influence of these criteria in this way. In Equation 4.1 we show an example of a linear weighting function aiming to calculate term importance (Fresno and Ribeiro, 2004):

$$I_k = C_f f_f(k) + C_t f_t(k) + C_e f_e(k) + C_p f_p(k) \tag{4.1}$$

where I_k is the estimated importance of term k in the document, C_f, C_t, C_e and C_p are coefficients established for each criterion—frequency in the whole document, title, emphasis and word position in the document, respectively—, and $f_f(k)$, $f_t(k)$, $f_e(k)$ and $f_p(k)$ are the frequencies of the term k in each criterion, normalized to the maximum frequency value found for that criterion in the document. These criteria were introduced in Section 2.2.2 on page 54 and they are explained in detail later in this chapter (see Section 4.2).

To give a concrete example, when a term k is important in the title t, then the corresponding component $C_t f_t(k)$ will have a value that is always added to the importance of the term in the document. We show an example with numbers in equations 4.2 and 4.3 where a term appearing only in the title is considered more important than another one appearing in the document body (with frequency half times the maximum and with different positions distributed throughout the document) and also emphasized:

$$I_k = 0.2 \times 0 + 0.4 \times 1 + 0.3 \times 0 + 0.1 \times 0 = 0.4 \tag{4.2}$$

$$I_k = 0.2 \times 0.5 + 0.4 \times 0 + 0.3 \times 0.5 + 0.1 \times 0.5 = 0.3 \tag{4.3}$$

Notice that it is easy to get a frequency of 1 for a term appearing in the title of a document because of the small number of words that titles usually contain. For this reason, it is unusual to find terms more than once in the same title and, in these cases, all the terms in the title will have a frequency value of 1.

In order to allow the definition of related conditions for establishing term importance—e.g., a term having high frequency in the document should appear in the title or emphasized to be considered important—, in this thesis we are interested in fuzzy combinations of criteria. The use of fuzzy logic allows to declare the knowledge without the need of specifying the calculation procedure. The knowledge is specified by means of a set of IF-THEN rules. These rules are close to natural language and ease the understanding of the system. Fuzzy logic also allows to specify the combination in such a way that the final importance of a term could depend on the relations among the criteria and their values. Thanks to these characteristics, fuzzy logic ease the task of expressing heuristic knowledge about web page representation, task that in other kind of systems requires an additional effort to understand how the system works in a deeper level of detail. In this context, we also provided a brief explanation of the main ideas of FCC document representation in Section 2.2.2 on page 54, an approximation to fuzzy combination of criteria in order to calculate term importance in a document. In the present chapter we focus our research on studying in detail the framework behind FCC and evaluating its advantages and disadvantages. As a result, we present a representation resulting of our findings and its empirical evaluation, showing significant improvements over FCC.

4.2 FCC: a previous Web Page Representation Based on Fuzzy Logic

In this section we describe FCC from the main high level ideas behind the representation to a detailed description of the fuzzy system that implements them. We also include the description of PF, a dimension reduction technique created to be used together with FCC.

4.2.1 Description of the Fuzzy System

In a high abstraction level, we deal with the initial hypothesis of the present work: a good representation would try to extract the most important terms to represent a document and therefore it could be based on how humans have a quick look at a document in order to determine the words that better represent document contents. From a psychological point of view, reading can be seen as the process by which a reader extracts visual information from a piece of written text and makes sense of it (Garrod and Daneman, 2006). But reading texts on a computer screen is different from reading them on paper. The use of hypertext is a basic difference that could influence this difference in the reading processes. Morkes and Nielsen (1997) found in their tests that only a 16% of users read web pages in a sequential manner, against a 79% of users who read the pages moving their attention among different parts of the page and not word by word, like usually occurs with printed texts. Other studies like Spyridakis (2000) and Isakson and Spyridakis (2003) have suggested some key aspects of HTML document creation in order to ease their understandability. Thus, for a human reader, title and emphasized words in a text document have a bigger role than the rest of the document in understanding its main topic. Moreover, the beginning and the end of the body text could contain overviews, summaries or conclusions with essential vocabulary. A more detailed description of psychological and psycho-linguistic processes involved when humans read documents is included in Fresno (2006).

The goal of FCC is to define the importance level of each word in a document by using a set of heuristic criteria:

1. Word frequency in the whole document.
2. Word frequency in document title.
3. Word frequency in emphasized text segments.

4. Word frequency in the beginning and the end of the document.

On HTML web pages, titles and other special texts are encoded with HTML tags, reason why a subset of the HTML tags are used in FCC in order to try to collect information of each single criterion aiming at finding the most representative words in the document. It is worth mentioning that FCC approach could be used with any kind of formatted documents, not only with HTML-encoded documents, simply by collecting the corresponding criteria information (`title`, `position`, `frequency`, and `emphasis`).

FCC relies on a fuzzy system to capture and combine this information taking into account expert knowledge. This fuzzy system consists of three stages: fuzzification[1], rule base definition and defuzzification[2]. To give an idea of how the fuzzy system looks like, Figure 4.1 shows a schema of the FCC architecture. The arrows represent system inputs and outputs, that we describe later in this section. The knowledge base of a fuzzy system is composed of the data base, which contains the membership functions associated to the linguistic variables and the rule base, that is a set of IF-THEN rules based on propositions that contain linguistic variables. This way, the fuzzy system is built over the concept of

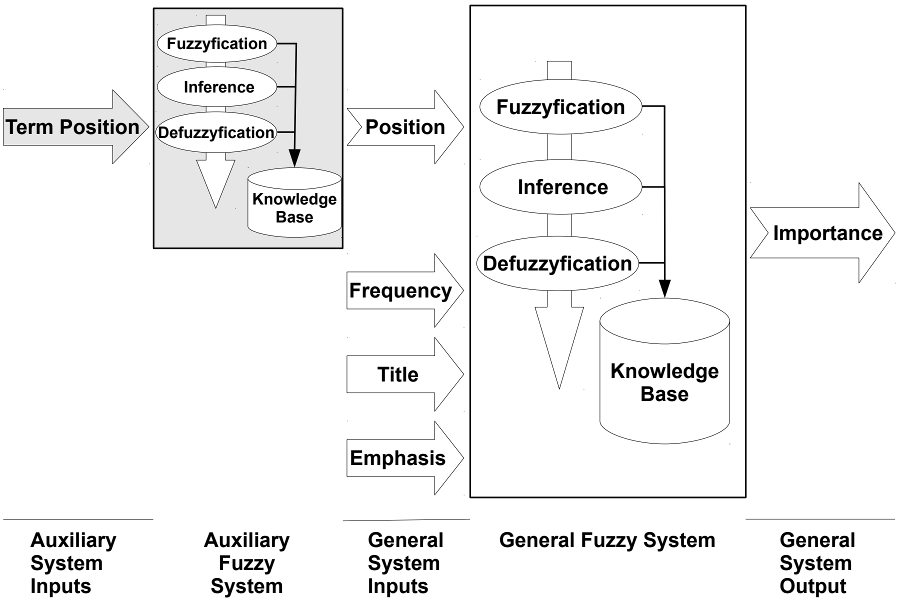


Figure 4.1: FCC system architecture.

[1] The term fuzzification refers to the process of transforming crisp values into fuzzy ones, that is, the membership degrees of the inputs to the fuzzy sets.

[2] Just the opposite to fuzzification, fuzzy values are converted to crisp ones.

linguistic variable. Each variable describes the membership degree of an object to a particular class and it is defined by human experts. This membership degree is defined by a membership function. For each heuristic criterion exposed above, FCC has an associated linguistic variable, as well as for the system output:

1. `Position`: term global position in the document. It is obtained by means of an Auxiliary Fuzzy System that takes as inputs all the positions of a term in a document (captured by means of the linguistic variable `term position` shown in Figure 4.2a) and returns the global position value in terms of two fuzzy sets: *standard* and *preferential* (see Figure 4.2b). It should be noted that the output linguistic variable of the auxiliary system is used as input in the general FCC system, as shown in Figures 4.1 and 4.3d.

2. `Frequency`: term frequency in the document. Its input is calculated by normalizing the number of occurrences of a term in a document to the maximum number of occurrences of a term in that document. This linguistic variable is defined by means of three fuzzy sets: *low*, *medium* and *high*, as shown in Figure 4.3a.

3. `Title`: term frequency in the title (terms between the `<title>` tags). Its input is calculated by normalizing the number of occurrences of a term in the title of a document to the maximum number of occurrences of a term in the title of that document. This linguistic variable is defined by means of two fuzzy sets: *low* and *high*, as shown in Figure 4.3b.

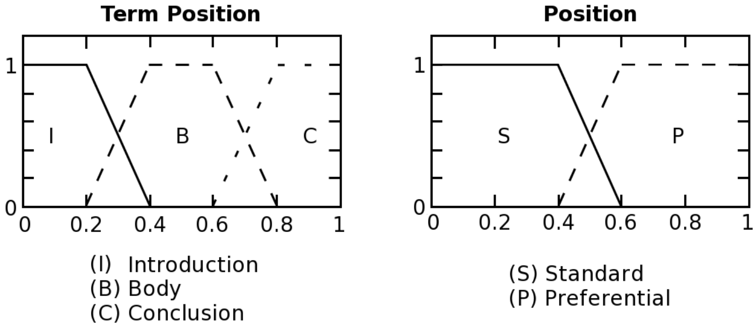
4. `Emphasis`: term frequency in emphasized parts of the text (included in tags like `<em>`, `<h*>`, `<b>`, etc.[3]). Tags like `<h*>`, used for headers, are included in this criterion and not in `title` because it is considered that a document has a unique title, while headers can refer to different sections within the document. Its input is calculated by normalizing the number of occurrences of a term in emphasized text segments in a document to the maximum number of occurrences of a term in emphasized text segments in that document. Like `frequency` linguistic variable, it is composed of three fuzzy sets: *low*, *medium* and *high* (see Figure 4.3c).

5. `Importance`: it is the output of the fuzzy system and corresponds to the estimated importance of a term in the document content. This linguistic variable has five fuzzy sets (see Figure 4.4): *no*, *low*, *medium*, *high* and *very high*.

Note that all these membership functions have trapezoidal shape in FCC, as it can be seen in the corresponding figures. Trapezoids can be seen as an approximation to other function shapes like Gaussian function, that is the density func-

[3] In this work we consider the following HTML tags for emphasis: `<em>`, `<b>`, `<u>`, `<strong>`, `<big>`, `<h*>`, `<cite>`, `<dfn>`, `<i>`, `<blockquote>`

tion of a normal distribution. All the variables except `emphasis` are defined by means of sets of equal size symmetrically distributed along the possible input values. These sets were defined regardless of concrete datasets, aiming to divide the input space in sets of equal size considering the possible input values, that is, frequency values from 0 to 1, since the input values are normalized to the maximum. Nevertheless, `emphasis` is considered separately because when the maximum frequency value for emphasized words in a document is small, the normalization could mislead us about the importance of other emphasized terms. For example, using symmetrical sets and having a maximum of 4 would lead to consider the importance of terms emphasized once as *low*. However, emphasis is used by authors to stress some words they consider important over the rest. In the case of our example we could want to give more importance to this kind of terms. For this reason, the sets for `emphasis` were asymmetrically defined. This way, frequencies that would be strictly *low* can be also considered as *medium*, since we can expect small maximum values in `emphasis`. Figure 4.3c shows how FCC takes this issue into account by defining different sets for `emphasis` than for the rest of the criteria, with bigger *medium* and *high* sets and a smaller *low* set, i.e., the *low* set for `emphasis` will only include terms with small emphasized frequency that also belong to documents with high maximums.

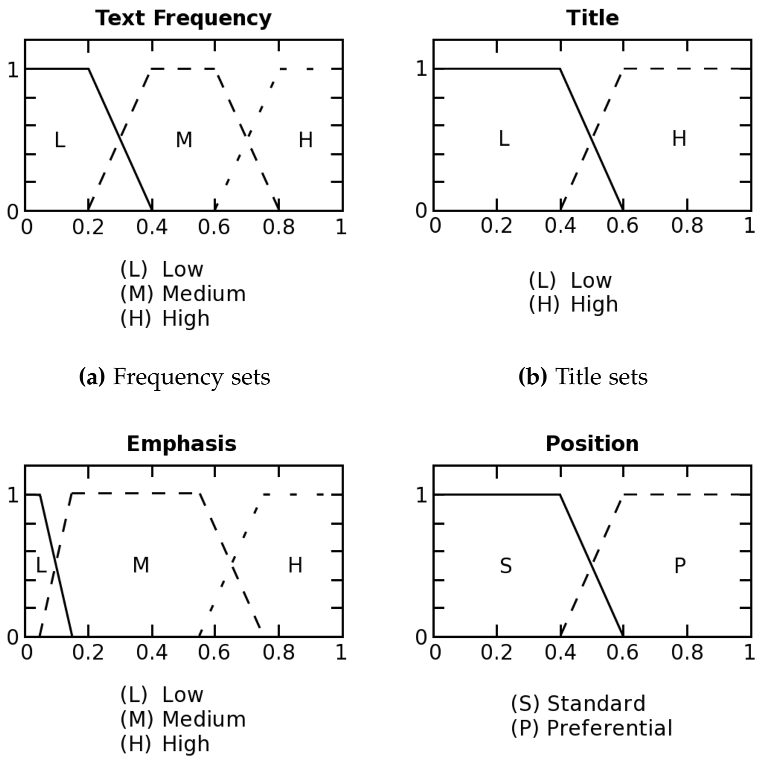


(a) Input linguistic variable: Term Position.

(b) Output linguistic variable: Position.

Figure 4.2: Data base for the Auxiliary Fuzzy System: input and output linguistic variables.

The other part of the knowledge base is a set of IF-THEN rules that combine the variables in order to define the behavior of the system. The aim of the rules is to combine one or more input fuzzy sets (antecedents or premises) and to associate them with an output fuzzy set (consequent). Once the consequents of each rule have been calculated, and after an aggregation stage, the final set is obtained.



(a) Frequency sets

(b) Title sets

(c) Emphasis sets

(d) Position sets

Figure 4.3: Data base for FCC: input linguistic variables.

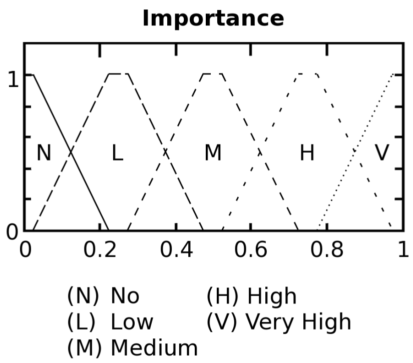


Figure 4.4: Data base for FCC: output linguistic variable Importance.

First, we show the rule base for the Auxiliary Fuzzy system in Table 4.1. This is a small set of rules aiming to establish the `position` value of a term in a document depending on where that term appears more frequently within the document. The rule base assigns preferential position to terms appearing mostly in the first or the last part of the document, and standard position to the rest. To do this, the document is divided in three sections that were called: *introduction*, *body* and *conclusion* (see Figure 4.2a). It is worth noting that this is just a naming scheme to split document contents. However, the contents of each of these three parts could have nothing to do with an introduction or conclusions (the combination of criteria should deal with this fact). Then, each time a term appears in the document, its position—normalized to the total number of terms in the document and previously referred to as `term position`—is used as input for the Auxiliary Fuzzy System. The total number of terms is used for normalization, because the different renderings of the same page depending on the browser, window size, etc., make very difficult to use other references, as the number of lines in the document, in a solid way. Once all the positions of the same term have been evaluated by the auxiliary system, the result is the global position of that term, called here `position` to simplify the naming. More details about how the evaluation process is performed are given below, after explaining the General Fuzzy System. The idea behind the auxiliary system is to assign different degrees of membership of a given word to *preferential* or *standard* sets depending on whether that word appears more frequently at the beginning and at the end of the document, or in the middle.

Table 4.1: Rule base for the Auxiliary Fuzzy System.

IF	Term Position	THEN	Position
	Introduction	⇒	Preferential
	Body	⇒	Standard
	Conclusion	⇒	Preferential

The general system has a bigger set of rules as can be seen in Table 4.2. As these rules are based on expert knowledge, it is advisable to understand the ideas behind the rules before trying to read the rules themselves. Concretely, FCC rule base relies on the following considerations:

1. If a word appears in the title or the word is emphasized, that word should also appear in one of the other criteria in order to be considered important. This aims to alleviate the problem of rhetoric titles or non-informative highlighting, problems that were described in Section 1.2.
2. Words appearing in the beginning or at the end of a document may be more important than the rest of the words, because documents sometimes contain overviews and summaries in order to attract the interest of the reader. Besides, FCC is a general representation, not oriented to a particular type of document, which means that some documents could have introduction and conclusions and others do not. By means of the combination of criteria

we can try to detect which case we are dealing with. When the words in preferential position do not also appear in the title or emphasized, then we could assume that the document does not follow the mentioned structure and we could reduce the importance value of that word.

3. It is possible that there are no emphasized words in a document. In the same way, it is possible that a document does not have a title, or the title does not contain important words. In these cases we have to take care of the penalization it could cause to the combination. We could not always expect to find the most important terms by combining `title` and `emphasis`, other criteria also count, particularly in these cases.

4. If the previous criteria were not able to choose the most important words, the frequency of the words in the whole document may help to find them. Different from the others, `frequency` criterion is always available, so it gives a last chance to establish word importance when the rest of the criteria fail.

Keeping these ideas in mind, now we can concentrate on the rule base for FCC, which is shown in Table 4.2. In this table, each row is a rule that contains the values of different criteria and the resulting output (`importance`). Yet, essentially, the way of reading the propositions does not vary too much. First, blanks in the table correspond with premises that are not present in a concrete rule. That is, the values corresponding to that criterion do not affect the rule. We have also separated rules containing blanks from the rest, to ease table reading. To give an example, the first row below the title header should be read as:

```
IF Title IS High AND Frequency IS High AND Emphasis IS High
   THEN Importance IS Very High
```

Going into more detail, the rule set must be complete, that is, given any possible input, at least one rule must be fired. It is important to realize than more than one rule could be fired at the same time. In those cases, the inference engine evaluates all the fired rules on the basis of the *Center Of Mass*[4] (COM) algorithm, that weights the output of every fired rule, taking into account the truth degree of their antecedents. The COM algorithm takes the balance point or centroid of all the scaled membership functions taken together for that variable (Hopgood, 2011).

Lastly, as we mentioned above, the output is a linguistic label (e.g., *low*, *medium*, *very high*) with an associated number related to the importance of a word in the document. Concretely, the output—for each term input to the system—is calculated by scaling the membership functions by product and combining them

[4] It is the most common method for defuzzification and it is also known as the centroid, center of gravity or center of area method.

Table 4.2: Rule base for FCC. Inputs are related to normalized term frequencies.

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
	High		High		High			⇒	Very High
	High		Medium		High			⇒	Very High
	High		High		Medium			⇒	Very High
	High		Medium		Medium			⇒	High
	Low		Low		Low			⇒	No
	High		High		Low		Preferential	⇒	Very High
	High		High		Low		Standard	⇒	High
	High		Medium		Low		Preferential	⇒	Medium
	High		Medium		Low		Standard	⇒	Low
	High		Low		High		Preferential	⇒	Very High
	High		Low		High		Standard	⇒	High
	High		Low		Medium		Preferential	⇒	High
	High		Low		Medium		Standard	⇒	Medium
	High		Low		Low		Preferential	⇒	Medium
	High		Low		Low		Standard	⇒	Low
	Low		High		High		Preferential	⇒	Very High
	Low		High		High		Standard	⇒	High
	Low		High		Medium		Preferential	⇒	High
	Low		High		Medium		Standard	⇒	Medium
	Low		High		Low		Preferential	⇒	Medium
	Low		High		Low		Standard	⇒	Low
	Low		Medium		High		Preferential	⇒	Very High
	Low		Medium		High		Standard	⇒	High
	Low		Medium		Medium		Preferential	⇒	Medium
	Low		Medium		Medium		Standard	⇒	Low
	Low		Medium		Low		Preferential	⇒	Low
	Low		Medium		Low		Standard	⇒	No
	Low		Low		High		Preferential	⇒	High
	Low		Low		High		Standard	⇒	Medium
	Low		Low		Medium		Preferential	⇒	Medium
	Low		Low		Medium		Standard	⇒	Low

by summation. These kind of systems are called additive and their main advantage is the computing efficiency (Kosko, 1998). A more detailed explanation of the fuzzy system can be found in Ribeiro et al. (2003) and Fresno (2006).

4.2.2 PF Method for Dimension Reduction

This method is oriented to term weighting functions that assigns weights that are directly proportional to the estimated importance for a feature in a document. Other lightweight reduction methods are based on frequencies or probabilities calculated from these frequencies. Thus, commonly used reductions like DF do not take into account all the factors that FCC combines, but only frequencies within the corpus. For this reason, together with FCC, a different reduction method called PF was proposed, aiming to avoid penalizing functions that consider other aspects than frequency, like FCC. PF is based in the term weighting function itself, that is employed also as feature reduction function.

The first step of this method is ranking the terms corresponding to each page on the basis of their weight. Then, for each document, the first n ranked terms are selected to compose the reduced vocabulary. Then, the maximum number of terms in the reduced vocabulary will be $n \times |D|$, where $|D|$ is the number of documents in the collection. Actually, the number will be smaller because we can find the same term in different rankings (each ranking corresponds to a different document). Besides, for different term weighting functions applied on the same collection, the term rankings for each document will be different and the reduced vocabularies obtained by using PF could be of different size and contain different terms. Reducing by DF, the reduced vocabularies would be the same, regardless the different weighting functions applied.

It is important to apply methods like PF to term weighting functions that do assign weights directly proportional to the estimated importance for a feature in a document. For instance, by using PF over term weights calculated with TF, the result will be to select the terms with highest frequencies on each page, that could result in a vocabulary composed of common use terms (that should not be the most important ones, following the hypothesis of H.P. Luhn exposed in Section 1.2). Moreover, binary weighting functions could lead to a random selection, as the number of possible values for each term is limited and it is not possible to create suitable term rankings.

4.3 Experimental Settings

In this section we describe the common experimental settings we use in all the experiments of this chapter.

First, in preprocessing, a list consisting of 621 stop words was used to remove common words. Punctuation marks were also removed. Terms were stemmed using a standard implementation of the Porter's algorithm for English[5]. After this preprocessing the vocabulary sizes are:

- Banksearch: 210,785 terms.
- WebKB: 40,258 terms.
- SODP: 625,137 terms

Regarding the clustering process, we chose Cluto rbr (k-way repeated bisections globally optimized) as a state of the art algorithm Karypis (2003). The number of desired clusters (K) was fixed to the number of categories in the dataset in order to make the evaluation process more intuitive. Thus, it is possible to find a perfect clustering and the algorithm does not introduce additional bias to the representation step. That is, when we modify the representation, we can be sure that the variation on the clustering results will come from that modification. The rest of the algorithm parameters were set by default. More details regarding the algorithm and its parameters can be found in Section 2.3 on page 66.

As all the datasets we deal with in this thesis have associated their ideal solution, we decided to employ an external evaluation measure to evaluate the clustering quality. These solutions can be used as evaluation benchmarks or gold standards. Then, we need a measure which evaluates how well the answer of our clustering system matches that gold standard (Manning et al., 2008). Typically the F-measure (Van Rijsbergen, 1974) (Equation 4.6) is the most used for this task which is equal to the harmonic mean of recall (Equation 4.4) and precision (Equation 4.5):

$$Recall(i,j) = \frac{n_{ij}}{n_j} \tag{4.4}$$

$$Precision(i,j) = \frac{n_{ij}}{n_i} \tag{4.5}$$

$$F(i,j) = \frac{2 \cdot Recall(i,j) \cdot Precision(i,j)}{Recall(i,j) + Precision(i,j)} \tag{4.6}$$

where i is the cluster, j the category, n_i the number of documents in the cluster i and n_j the number of documents in the category j. Both terms, cluster and category were defined in Section 3.1. In clustering, recall is the fraction of documents in the category j that are also in the cluster i. Precision refers to the fraction of documents in the cluster i that are also in the category j (Crabtree et al., 2007). The overall F-measure is the weighted average of the F-measure for each category (Equation 4.7):

$$F = \sum_j \frac{n_j}{n} \cdot \max_i \{F(i,j)\} \tag{4.7}$$

[5] http://www.tartarus.org/~martin/PorterStemmer

where n is the total number of documents we want to group, n_j is the number of documents in a category j, and the maximum function is applied over the whole set of clusters. The F-measure values are in the interval [0,1] and larger values correspond to higher clustering quality.

Other measures like entropy, purity or inverse purity have been also used in the literature to evaluate clustering quality. Recently, the BCubed precision and recall metrics have been also applied to clustering evaluation, as they satisfy some constraints that other measures do not (Amigó et al., 2009). However, the most widely used is F-measure based on precision and recall, that is utilized in this thesis and allow us to keep backwards compatibility of our results with previous works.

4.4 Dimension Reduction

To compare different term weighting functions we need to reduce the initial vocabulary to smaller sizes. Using different reduced vocabulary sizes also allows us to analyze the behavior of the different weighting functions when the vocabulary size changes. This is an important factor, as it could allow to establish a compromise between the reduction and the results. Basically, in terms of computational cost, it is interesting to use vocabulary sizes as small as possible without strongly penalizing clustering results.

However, as we saw before, PF reduction does not allow to select the exact number of features in the reduced vocabulary. With PF, vocabulary size will depend on the number of unique terms on the first n positions of each document ranking. To compare different reduction methods we should use the same vocabulary sizes for all of them. There are different dimension reduction techniques that are able to overcome this limitation of PF. In this section we test some of them first and them we propose a new method based on the idea of PF, but overcoming its limitation.

We evaluate the effect of different dimension reduction techniques in document representation. Our goal is understanding how dimension reduction affects clustering results, taking into account how this process benefits or not the different term weighting functions.

4.4.1 Initial Tests

First, we analyze the effect of using different dimension reduction techniques over two different term weighting functions that we will use as baselines in this chapter. On the one hand, FCC weighting function, because we are using the same framework and we think it has to be used as a baseline in this thesis. On the other hand, TF-IDF has been used as a standard term weighting function in clustering. This function exploits only plain text, so it is another good baseline to compare against functions that combine different criteria together with plain text. It is worth noting that both functions, FCC and TF-IDF were evaluated against other functions, like ACC (see 1.2 on page 24), binary, binary-IDF, TF, TF-IDF or weighted IDF, in previous works (Fresno, 2006; Pérez García-Plaza et al., 2008; Pérez García-Plaza et al., 2009). In most cases FCC and TF-IDF outperformed the rest in clustering tasks on Banksearch and WebKB collections (described in sections 3.2 and 3.3 respectively).

In order to reduce the vocabulary size, in this thesis we employ different techniques (these and other techniques were described in Section 2.2.3, here we describe some details related with how we use them in this thesis):

- Document Frequency (DF): the size of the reduced vocabulary is obtained by means of two thresholds. These thresholds are maximum and minimum values for DF, in such a way we select those terms whose DF is between both thresholds. They are usually defined as percentages of the total number of documents. For example, selecting the upper threshold as 60% would mean selecting only terms appearing in less than the 60% of collection documents. Thus, the thresholds are used to decide the final size of the feature set, although with this technique it is necessary to test different possible values to get an approximate number of features in the final vocabulary. It is also possible to obtain two different vocabularies from different thresholds but with approximately the same size, as different thresholds could lead to the same reduction in terms of size, but with different terms. We decided to include it with TF-IDF in this thesis because they are commonly used together in the literature achieving reasonably good results, as we saw in Section 2.2.3.

- Latent Semantic Indexing (LSI) (Landauer et al., 1998): in this thesis, as suggested by Tang et al. (2005), LSI was applied after a previous reducing step to alleviate its computational complexity. We reduced vector dimension on the basis of the corresponding term weighting function by selecting the highest ranked terms for each document from the original size—$210,785$ terms for Banksearch, $40,258$ for WebKB—to $5,000$ features before applying LSI. To compute LSI with vectors having more than $5,000$ dimensions requires too much time for the computation (to compute the algorithm to reduce to $2,000$ features the elapsed time is about an hour, in addition to the initial reduction to $5,000$ features needed to make possible the computation of LSI).

- Random Projection (RP) (Bingham and Mannila, 2001): as Tang et al. (2005) stated, the effectiveness of RP in text clustering is still not clear (as a substitute of LSI), reason why we decided to test this technique by ourselves in this thesis instead of assuming that can be used as a lightweight alternative to LSI. Precisely, as RP is used as an alternative method to LSI, the same previous reducing step was performed in both cases.

The number of features per vector, that is, the vocabulary size, was not fixed to a unique value. We wanted to discover how different each term weighting function behaves depending on the vocabulary size. Thus, in our research for this thesis, vector sizes were reduced to 100, 500, 1,000, 2,000, and 5,000 dimensions. The maximum of $5,000$ was selected because with this size, the computational cost of the clustering begins to be considerable. Moreover, the size of the initial vocabulary corresponding to the smallest dataset, WebKB, is $40,258$ terms, so reducing to $5,000$ features constitutes a reduction of approximately one order of magnitude. We consider this scale reasonable for the reduction process.

Table 4.3 shows the F-measure results for the combinations of weighting functions and dimension reduction techniques mentioned above. These experiments were carried out over two datasets: Banksearch and WebKB (see Chapter 3, sections 3.2 and 3.3 for more information on these collections). Each table row contains F-measure values corresponding to the clustering solution obtained by using the representation specified in the first column with the number of features per vector detailed on top of the remaining columns, being Avg. and S.D. the average and the standard deviation for that row. The average value and the standard deviation give an idea of the global behavior of the corresponding representation method. However, they do not take into account that, for clustering tasks, it is more interesting to obtain good results with as few features as possible. This way, we will reduce the computational cost of the clustering algorithm. On the other hand, as we do not know the perfect number of features to represent a document, a good balance among the results of all the vocabulary sizes is also desirable. So, what we are looking for is a representation with a good average value combined with good results with vocabularies below $1,000$ features. This way, reducing to smaller vector sizes would guarantee good clustering quality at the same time that the computational cost remains reasonable. Besides, the S.D. gives us an idea about the stability of the results among different vocabulary sizes.

Looking at the results, RP shows no real advantage over the rest of reductions in all cases. Although it is computationally less expensive than LSI, its results are worse than the rest in most of the cases, especially when we apply higher dimensionality reductions. It is worth mentioning that in WebKB, FCC RP seems to progressively improve its clustering results from 100 to $1,000$ features, where it achieves its best result. However, with larger vector sizes its results fall again. Then, choosing RP for reducing dimensionality it is not a good option as the probability of obtaining bad clustering results depending on the vector size is high.

The case of DF is particularly interesting, because it is the worst reduction

Table 4.3: F-measure results for dimension reduction experiments

Rep.\Dim.	100	500	1,000	2,000	5,000	Avg.	S.D.
Banksearch							
TF-IDF DF	0.398	0.674	0.723	0.748	0.749	0.659	0.149
TF-IDF RP	0.480	0.685	0.707	0.730	0.741	0.669	0.108
TF-IDF LSI	0.750	0.755	0.756	0.757	**0.763**	0.756	**0.005**
FCC RP	0.484	0.739	0.740	0.746	0.758	0.693	0.117
FCC LSI	**0.775**	**0.763**	**0.785**	**0.763**	0.758	**0.769**	0.011
WebKB							
TF-IDF DF	0.469	**0.521**	**0.530**	**0.534**	**0.532**	**0.517**	0.027
TF-IDF RP	0.313	0.423	0.492	0.499	0.516	0.449	0.084
TF-IDF LSI	**0.516**	0.507	0.505	0.506	0.501	0.507	**0.006**
FCC RP	0.446	0.477	0.528	0.470	0.470	0.478	0.030
FCC LSI	0.449	0.460	0.473	0.474	0.475	0.466	0.011

method in Banksearch, but at the same time it can be considered the best one in WebKB. We strongly believe that the good balance of documents per category hinders DF in Banksearch due to the thematic divergence among categories (see Section 3.2.1). This way, when reducing to small vector sizes, useful terms to represent concrete categories would be removed. Reducing by DF has a clear drawback: the way of establishing the thresholds. There is no automatic way of selecting them, so the performance of the system could be affected by changing them, even in the case of reducing the vocabulary to the same size (but by using different thresholds values). In our case we select the upper threshold about a 60% to remove too common terms, and we change the lower threshold to get the desired number of features. The upper threshold is used to make minor adjustments, as the number of terms with high DF is relatively small, while the lower threshold is used to make major changes in the number of features, as there is a very large number of terms with low DF.

Looking at the row of TF-IDF DF for Banksearch in Table 4.3, when we apply aggressive reductions, like below $1,000$ features per vector, we could probably be removing category related terminology, particularly with the lower DF threshold. It should be taken into account, that the lower threshold needed to get only 100 features is established with values that are bigger than category size (in Banksearch all the categories have the same size, a 10% of the size of the collection). When we relax the thresholds, this terminology enters to our vocabulary, leading to an improvement in clustering results, as occurs from $1,000$ to $5,000$ features. However, the bad result obtained with 100 features strongly penalizes this row, because it implies that choosing a wrong number of features (or the wrong thresholds) would be harmful for the clustering results. By choosing a different lower threshold (below category size), we could probably improve the clustering. However, to do this selection we had needed to know, at least, the size of the categories we wanted to find. Then, in absence of this kind of information, reducing by DF can lead to bad results with aggressive reductions, since it does not provide a clear way of estimating the least important terms.

Nevertheless, the case of WebKB and TF-IDF DF is just the opposite, because WebKB consists of pages from Universities, most of them from only four Universities. In Section 3.3.1 we saw that the difficulty of clustering this collection comes mainly from the heterogeneity of documents within categories, that are also unbalanced in terms of documents per category. In this sense, DF seems to help removing terms shared among different categories. Moreover, four out of six WebKB categories are also bigger than Banksearch ones, with respect to the total size of the dataset. Thus, the lower threshold would have a softer effect, and probably in at least three categories (that are bigger than a 20% of collection size) would have even a positive effect, by removing the least common terms in those categories. Besides, the WebKB has an initial vocabulary smaller than Banksearch (40, 258 terms WebKB, 210, 785 terms Banksearch), so the reduction is less aggressive in the case of the former. For all the reasons exposed above, it can be said that the DF method applied here fits better the problem of filtering too common or too rare terms for WebKB than for Banksearch.

Looking at the performance of TF-IDF LSI in Banksearch we find very good results. They show also good stability regardless the vocabulary size. Its results are only outperformed by FCC LSI in Banksearch. Both, term weighting functions and the dimension reduction technique works well in Banksearch. However, focusing on WebKB, results are different. First, TF-IDF LSI obtained good results, but DF outperformed LSI in 4 out of 5 cases. When using the smallest vector size LSI performs better than DF. The former transform the features to independent components and this approach seems to work well to maintain the results of the weighting function regardless the vector size. This stability could be explained because by finding the independent components, the fundamental information is preserved through the reduction process. The latter seems to remove important features when reducing to 100 features, but moving to 500 features outperforms the rest of the combinations of term weighting functions and dimension reduction techniques for WebKB. As we commented before, we believe that DF fits better the problem of removing the least representative terms for each category. In addition to this fact, DF does not depend on the term weighting function values assigned to vocabulary terms, while LSI is applied over the 5, 000 terms having highest weights. Then LSI can only find independent components among these 5, 000 terms, so if the weighting function left some representative terms out of them, LSI can not do anything to solve it (it can be said that LSI is limited by the previous reduction to make its application computationally feasible). Nevertheless, DF is applied over the whole initial vocabulary without using the term weighting function. We believe that DF in WebKB helps TF-IDF to select better the most representative terms for differentiating the categories that the weighting function. Probably the use of IDF hinders the function by penalizing terms in the bigger categories and boosting those belonging to the smaller ones, since in this case high frequencies in the biggest categories are also high frequencies in the collection.

On the other hand, when applied on FCC, LSI achieve very bad results in WebKB, even worse than FCC RP. As LSI search form independent components, we believe these bad results are due to the bad performance of the weighting function in this collection, where FCC is not able to find the most important terms for each document. Later, in Section 4.5 we will analyze this bad performance in more detail.

Summarizing, from this initial experiments we discard RP reduction because of its bad results in almost all cases. Among the rest, LSI is the best option in Banksearch regardless the weighting function. Nevertheless, in WebKB, DF reduction lead to results even a 6% better than LSI (about 0.03 in terms of F-measure). Both of them will be used as baselines in this Chapter for WebKB, given the very good result of TF-IDF LSI with 100 features. However, DF achieves higher F-measure values in the rest of the cases, showing also higher maximum values. Globally, we recommend the use of DF together with TF-IDF for collections with heterogeneous documents within categories, that have also different sizes, and gathered from a small number of web domains. In these cases, DF can help removing too common terms and web domain related terminology spread throughout the collection (we talked about it in Section 3.3).

Lastly, there is no clear candidate combination of weighting function and reduction method to be used regardless of the collection, being TF-IDF LSI the most stable option, but at the same time, being improved by others in both collections: FCC LSI in Banksearch, TF-IDF DF in WebKB.

4.4.2 MFT: a Proposal for Dimension Reduction

On the one hand, as we saw in Section 4.2.2, PF reduction method does not allow to reduce the initial vocabulary to concrete sizes. In particular, the smallest reduction it could achieve depends on the number of unique terms in the first position of each document ranking. This usually leads to reduced vocabulary sizes larger than $1,000$ features, depending on the collection. This is a problem for our analysis because of the good results that can be achieved with only 100 features, shown with some combinations of term weighting functions and dimension reduction techniques in the previous Section 4.4.1. On the other hand, a reduction like PF is especially appropriate for functions like FCC that aims to establish term importance from document contents. It also allows to directly analyze the correlation between the estimated importance given to terms and clustering results, since by selecting the most important terms we should achieve better clustering results. Then, this kind of reductions would allow a direct analysis of the weighting function, which is quite interesting for the goals of this thesis.

For these reasons, we presented a reduction method called Most Frequent Terms (MFT) (Pérez García-Plaza et al., 2008) that, as in the case of PF, is based on term importance estimated by means of a term weighting function. It can be seen as a method to select a subset of features composed of the n features with highest estimated importance. The algorithm also grants to get always the same output when reducing to a concrete vocabulary size, in contrast with other methods like DF.

MFT method works as follows:

1. The terms in each document are ranked based on the term weighting function values.

2. Then, terms on the first position in the document rankings are put in order according to how many times they have appeared in the rankings. If two or more terms appear the same number of times in different rankings, we put them in order based on the maximum weight found for each of them.

3. Next we take the terms appearing in the second position in the rankings, and so forth.

4. The process stops when the desired number of terms is reached. Notice that by following this algorithm the resulting list may be larger than the required size, because there are as many rankings as documents in the dataset (this is exactly the same issue we have commented about PF). Nevertheless, as we put the list in order, we can get the exact number of terms just taking the first n terms of the list.

This method was applied to FCC and TF-IDF. Table 4.4 shows the results of MFT reduction compared with the experiments of the previous section. The results corresponding to RP were removed for clarity, since it achieved the worst results.

Looking at the results of TF-IDF MFT, it is clear that this reduction does not work well for TF-IDF when reducing to smaller vocabulary sizes:

- Compared with LSI and DF in Banksearch, MFT outperformed the bad results of DF, while LSI improved MFT with small vector sizes (100 and 500 features). However, with $1,000$ and $2,000$ features MFT obtained higher results, and with $5,000$ the difference is less than a $0,01$, that corresponds to an improvement of only 0.66% of LSI over MFT. It seems that LSI helps TF-IDF to find independent components that were not discovered by the

Table 4.4: F-measure results for dimension reduction experiments

Rep.\Dim.	100	500	1,000	2,000	5,000	Avg.	S.D.
Banksearch							
TF-IDF DF	0.398	0.674	0.723	0.748	0.749	0.659	0.149
TF-IDF LSI	0.750	0.755	0.756	0.757	0.763	0.756	**0.005**
TF-IDF MFT	0.703	0.737	0.768	**0.772**	0.758	0.748	0.028
FCC LSI	**0.775**	**0.763**	**0.785**	0.763	0.758	**0.769**	0.011
FCC MFT	0.723	0.757	0.768	0.765	**0.768**	0.756	0.019
WebKB							
TF-IDF DF	0.469	**0.521**	**0.530**	**0.534**	**0.532**	**0.517**	0.027
TF-IDF LSI	**0.516**	0.507	0.505	0.506	0.501	0.507	**0.006**
TF-IDF MFT	0.385	0.438	0.466	0.498	0.513	0.460	0.051
FCC LSI	0.449	0.460	0.473	0.474	0.475	0.466	0.011
FCC MFT	0.453	0.472	0.475	0.468	0.475	0.469	0.009

weighting function itself (as MFT reduction process can be seen like obtaining the *n* terms with highest estimated importance from the initial vocabulary).

- In WebKB occurs something similar, but MFT only improved LSI with the largest vocabulary size. As MFT strongly depends on the term weighting function to select the most important terms, an improvement of LSI over MFT implies that the weighting function is not working as well as it could. In this sense, the use of IDF could be strongly penalizing results in WebKB, in a similar way as we commented for LSI after our initial tests in the previous section. Then, terms appearing in most of the documents in the biggest category (which has a 36% of WebKB documents, see Section 3.3.1) will be removed when reducing by means of MFT to smallest vector sizes, because their weights were penalized by their high document frequency. The same could occur with other two categories that are also bigger than a 20% of collection size. For this reason, frequency based methods as DF work better for TF-IDF in this collection, since they do not remove these terms unless they have higher DF than the upper threshold.

Focus our attention in FCC MFT, its results compared to FCC LSI show a similar relation:

- In Banksearch, FCC MFT only outperforms FCC LSI in the case with larger vocabulary size. So again, it seems like LSI is able to find independent components that are hidden for FCC weighting function, i.e., the weighting function is not finding the most representative terms, as LSI is able to find a better reduction than MFT.

- In WebKB, FCC MFT results are almost as bad as when using LSI. Sometimes MFT is slightly better, sometimes it is slightly worse. Then it seems that the problem is not in the reduction side, but in the term weighting function side.

Comparing both weighting functions again, the conclusions coincide with Section 4.4.1: while the combination of FCC and LSI outperforms TF-IDF in Banksearch, in WebKB the best results correspond to TF-IDF helped by DF and LSI.

After reviewing all these results we find three open issues that would be interesting to analyze:

- The first one has to do with FCC. After showing very good results in Banksearch, it achieved the worst ones in WebKB. Even after searching for independent components by means of LSI, its results were bad. Therefore, we believe that the problem is the weighting function itself. A deeper analysis on FCC could help find the reason of this bad performance.

- The second one, also commented in Section 4.4.1, has to do with the different performance of the same combinations of weighting function and reduction method in each collection. Therefore, there is no clear candidate combination that can be used regardless of the collection. As we said before and it is also seen in Table 4.4, TF-IDF LSI is the most stable option, but at the same time, it is improved by others in both collections: FCC LSI in Banksearch, TF-IDF DF in WebKB.

- Finally, apart from discovering which combinations of term weighting functions and reduction techniques lead to better clustering results, a second issue emerges: the difference between using MFT or LSI with the same term weighting function. The bad performance of MFT with small vector sizes points out that the weighting functions are not estimating well the importance of the words. Our hypothesis is: if a weighting function is able to assign the highest term weights to the most representative features of a document, MFT should work similar to, or even better than LSI.

4.5 Analysis of the Combination of Criteria

Section 4.4.2 left three open issues:

(1) To study the bad performance of FCC in WebKB dataset.

(2) To find a combination of weighting function and reduction technique with a more stable performance regardless the dataset than the previously tested combinations.

(3) To test whether we are able to find a term weighting function that used with MFT achieves a similar performance than used with LSI.

All of them are clearly related through a common need: to analyze the weighting function. For this reason, in this section we perform a study about how to improve the fuzzy combination of criteria performed by FCC for web page clustering.

We will use the framework offered by FCC to further investigate about how information extracted from HTML documents can be used to enrich document representation and to improve the results obtained in clustering tasks. We divide the section in two main parts:

- We perform an analysis on the combination of criteria, aiming to discover the influence of each individual criteria in the combination.
- After analyzing the results of those experiments, we present and compare two new combination proposals aiming to alleviate the problems we detected.

4.5.1 Study of Individual Criteria

The first step is to analyze the contribution of each criterion in order to find some clues about why the combination does not perform in WebKB as well as in Banksearch. To do this, we repeat the clustering process modifying the combination of criteria proposed by FCC. We did four variations of this function, one for each criterion, in such a way that the output of the system will correspond only to one criterion at a time. For example, Table 4.5 shows the rule base for the system that utilizes only `emphasis` criterion to determine the output. It is worth mentioning that the data base of the system remains untouched, as described in Section 4.2.1.

Table 4.5: Rule base for the system based on `emphasis` criterion. Inputs are related to normalized term frequencies.

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
					High			⇒	Very High
					Medium			⇒	Medium
					Low			⇒	No

We used MFT reduction because it does not transform features and selects those with higher weight, allowing us to study the effectiveness of each alternative to give more importance to the most representative terms.

Table 4.6 shows the results of each individual criterion compared to the combination, i.e., FCC. Focusing on Banksearch results, values corresponding to FCC are always higher than individual ones. This means that the combination contributes to improve the results over individual criteria in all cases. Besides, `frequency` obtains the best values, while position obtains the worst ones. We believe that `frequency` helps the good results of the combination more than the rest of criteria. The homogeneity within category documents affects positively to this criterion and allow to achieve good results in the combination.

Table 4.6: F-measure results for criteria analysis experiments

Rep.\Dim.	**100**	**500**	**1,000**	**2,000**	**5,000**	**Avg.**	**S.D.**
Banksearch							
FCC MFT	**0.723**	**0.757**	**0.768**	**0.765**	**0.768**	**0.756**	0.019
title MFT	0.626	0.646	0.632	0.634	0.639	0.635	**0.007**
emphasis MFT	0.586	0.671	0.674	0.685	0.693	0.662	0.043
frequency MFT	0.689	0.715	0.720	0.724	0.731	0.716	0.016
position MFT	0.310	0.525	0.538	0.599	0.608	0.516	0.121
WebKB							
FCC MFT	**0.453**	**0.472**	**0.475**	0.468	0.475	**0.469**	**0.009**
title MFT	0.432	0.433	0.404	**0.488**	0.479	0.447	0.035
emphasis MFT	0.415	0.431	0.433	0.465	**0.489**	0.447	0.030
frequency MFT	0.441	0.460	0.460	0.468	0.446	0.455	0.011
position MFT	0.301	0.283	0.317	0.281	0.286	0.294	0.015

WebKB results are quite different. On one hand, `frequency` is not always the best among individual criteria and, on the other hand, FCC does not always outperform individual criteria, concretely `title`, `emphasis` or `frequency` have equal or higher F-measure values in some cases when vector dimension is reduced to $2,000$ and $5,000$ features. In this collection, the frequency distribution of emphasized terms shows a more restricted use of emphasis, as it was explained (see Section 3.3.2 on page 88). It could be due to the limited number of web domains and the similarity among web page contents that only come from Universities. These factors could limit the number of different writing styles, fact that would be reflected in a less scattered distribution of emphasized term frequencies. Until now, we did not have any other information to confirm our belief, but the good results of `emphasis` with the largest vocabulary sizes lead us to confirm that emphasized terms in WebKB are particularly meaningful. The same consideration about the restrictions on the creation of WebKB can explain the good results achieved by `title` criterion. We can expect that authors use titles in a similar way than emphasis within the collection, as both resources are employed to highlight important words. Besides, it seems that `frequency` and `position` strongly affect FCC results, and going further, when `title` and `emphasis` could lead to a better clustering, their combination with `frequency` and `position` makes results worse. In particular, WebKB documents within categories can be much more heterogeneous than in Banksearch, factor that negatively affects `frequency` criterion. In this case, the combination should help to correct this issue, but it does not. Thus, it seems that `frequency` and `position` are hindering the combination.

Therefore, in general, while `frequency` gets higher results than the other criteria, the combination works fine, but when `title` or `emphasis` outperform `frequency`, the combination does not work as good as it could. Thus, `frequency` is very important for a good grouping, as well as `title` and `emphasis`, and all of them should be very important in the combination. However, `position` is the criterion with the worst results in all cases, so we have to take care using it to establish the importance of a term. This bad performance of `position` could come from its definition, since its heuristics were based on written texts and not in web pages, where document parts could have different importance level. Although one could expect to find an introduction at the beginning and brief summaries at the end of a web page, there are different types of pages and this could not happen so often. For this reason, `position` in a web page should be taken into account carefully to establish term importance.

4.5.2 Improving the Fuzzy Combination of Criteria

The first step to try to improve the combination is understanding the bad performance of FCC in WebKB. We know that the problem comes from the combination, as we showed in Section 4.5.1. In FCC rules (Table 4.2), when `frequency` is *low* output can be *very high* (the maximum) depending on `position`, if `title` and `emphasis` are *high*. As we saw before, `frequency` contributes to a good clustering much more than `position`, so the output should reflect that fact. But, in this case, `frequency` is totally ignored. This occurs again when `title` is *low* and `frequency` *medium*. Both criteria are important for a good grouping, but the output is *very high* based on `position`, the same as the previous case. In these cases we are clearly underestimating the discrimination power of `frequency` and `title`. The same happens when `frequency` is *medium*, being `title` and `emphasis` *low*: `position` decides again that `importance` can be the minimum or not, but `frequency` should count more than `position`, as we saw before in Section 4.5.1. Summarizing, FCC overestimates the contribution of `position`, underestimating at the same time the discriminative power of `title`, `emphasis` and `frequency`. When high frequencies favored the clustering, as in Banksearch where `frequency` got always better values than the rest of the criteria, the combination does not suffer the effect of this problem so much. However, when `frequency` is not so determinant in the clustering, as in WebKB, it seems that the problem has a greater effect on the results.

On the other hand, the high number of rules in FCC makes the possible combinations more difficult to understand. As the fuzzy system is able to combine the conclusions of the rules, another possibility for the knowledge base is the use of a set of single-input rules for each criterion. Thus, the system would calculate the output by combining the different outputs of the fired rules. In this thesis we call this approach AddFCC, which rule base is shown in Table 4.7. This approach reduces the number of cases that is needed to specify to the minimum. The data base of the fuzzy system remains untouched as it was described in Section 4.2.1.

Table 4.7: Rule base for AddFCC. Inputs are related to normalized term frequencies.

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
	High							⇒	Very High
	Low							⇒	No
			High					⇒	Very High
			Medium					⇒	Medium
			Low					⇒	No
					High			⇒	Very High
					Medium			⇒	Medium
					Low			⇒	No
							Preferential	⇒	Very High
							Standard	⇒	No

Nevertheless, if we are looking for very specific definitions for each criterion, we may miss part of the knowledge expressed in the FCC system, especially when dealing with dependencies among criteria and not all of them contribute equally to the combination, as occurs in our case. We strongly believe that an approach like AddFCC would reduce the expressiveness of the system, fact that, in some cases, could lead to mistakes as a consequence of a bad specification of the heuristic knowledge.

In order to avoid this problem, an intermediate approach is proposed. We refer to it as Extended Fuzzy Combination of Criteria (EFCC). Its rule base combines some criteria explicitly and for others lets the combination to the fuzzy engine (see Table 4.8). The main idea is to have two sets of rules: one for `frequency` and another for the rest of the criteria, in such a way that we have always at least one rule of each set fired by the system, which will combine the outputs. Thus, we simplify the problem of underestimating frequency, because both subsets are always evaluated and combined. We have also reduced the discriminative power of position criterion, that in EFCC is considered the least important. It is important to say that these rules are based on the same expert knowledge as FCC, but taking into account the issues discovered in Section 4.5.1 to obtain a better definition—in terms of a Fuzzy Rule Based System—of that knowledge.

Table 4.8: Rule base for EFCC. Inputs are related to normalized term frequencies.

IF	Title	AND	Frequency	AND	Emphasis	AND	Position	THEN	Importance
	High				High			⇒	Very High
	High				Medium		Preferential	⇒	High
	High				Medium		Standard	⇒	Medium
	High				Low		Preferential	⇒	Medium
	High				Low		Standard	⇒	Low
	Low				High		Preferential	⇒	High
	Low				High		Standard	⇒	Medium
	Low				Medium		Preferential	⇒	Medium
	Low				Medium		Standard	⇒	Low
	Low				Low		Preferential	⇒	Low
	Low				Low		Standard	⇒	No
			High					⇒	Very High
			Medium					⇒	Medium
			Low					⇒	No

First, in Table 4.9 we show the clustering results for the two new alternatives compared to FCC. Looking at these results, EFCC improves clustering results in WebKB, while in Banksearch AddFCC outperforms the rest. Thus, EFCC improves the bad performance of FCC in WebKB in all cases, but AddFCC does not. Moreover, EFCC also achieved good results in Banksearch.

Table 4.9: Comparison of the fuzzy logic-based alternatives in terms of F-measure.

Rep.\Dim.	100	500	1,000	2,000	5,000	Avg.	S.D.
Banksearch							
FCC MFT	0.723	0.757	0.768	0.765	0.768	0.756	0.019
EFCC MFT	0.768	0.778	0.758	0.740	0.759	0.760	0.014
AddFCC MFT	**0.775**	**0.788**	**0.777**	**0.784**	**0.779**	**0.781**	**0.005**
WebKB							
FCC MFT	0.453	0.472	0.475	0.468	0.475	0.469	**0.009**
EFCC MFT	**0.516**	**0.546**	**0.545**	**0.566**	**0.484**	**0.532**	0.032
AddFCC MFT	0.459	0.493	0.494	0.491	0.471	0.482	0.016

It is worth highlighting the bad result of EFCC in WebKB with 5,000 features, which is much lower than the rest of its results. EFCC weights terms more aggressively than FCC and TF-IDF, in the sense that its rules highly benefit terms appearing in titles and emphasized. As the number of these terms is limited, when we increase the number of features to select, the probability of selecting terms that appear neither in titles nor emphasized increases. When we reduce the initial vocabulary to 5,000 features, we have started to introduce this kind of terms in the representation, and therefore the results get worse, or better said, EFCC results approximate the results of the other representations.

Besides, AddFCC obtains the best results in Banksearch in all cases. However, it leads to worse results than EFCC in WebKB. In AddFCC all the criteria contributes equally to the combination. This way, its problem in WebKB is the same we detected for FCC: `position` is overestimated in the combination. Thus, despite its good performance in Banksearch, it does not solve the problem of FCC in WebKB. This fact supports our belief in the need for a system where not all the criteria contribute the same to the combination.

At this point, we decided to select EFCC as an alternative to FCC for our next experiments, since it solves the bad results of the other fuzzy logic based approaches in WebKB and it also achieves good results in Banksearch.

In our next experiment we apply LSI in combination with EFCC to test our hypothesis about the improvement obtained by LSI over MFT (that is based on the term weighting function). We also compare EFCC results with the best results achieved by TF-IDF and FCC for each dataset in Section 4.4.2. Table 4.10 shows the comparison.

Table 4.10: Comparison of EFCC with previous alternatives in terms of F-measure.

Rep.\Dim.	**100**	**500**	**1,000**	**2,000**	**5,000**	**Avg.**	**S.D.**
Banksearch							
TF-IDF LSI	0.750	0.755	0.756	0.757	0.763	0.756	**0.005**
FCC LSI	0.775	0.763	**0.785**	**0.763**	0.758	**0.769**	0.011
EFCC MFT	0.768	**0.778**	0.758	0.740	**0.759**	0.760	0.014
EFCC LSI	**0.780**	0.756	0.744	0.755	0.757	0.758	0.013
WebKB							
TF-IDF DF	0.469	0.521	0.530	0.534	**0.532**	0.517	0.027
TF-IDF LSI	**0.516**	0.507	0.505	0.506	0.501	0.507	0.006
FCC MFT	0.453	0.472	0.475	0.468	0.475	0.469	0.009
EFCC MFT	**0.516**	**0.546**	**0.545**	**0.566**	0.484	**0.532**	0.032
EFCC LSI	0.483	0.483	0.483	0.483	0.484	0.483	**0.000**

Looking at EFCC, these experiments corroborates our hypothesis: with EFCC we have improved our weighting function and, as a result, MFT has achieved clustering results as good as, or even better than LSI in most of the cases, with a much lower computational cost. Besides, in Banksearch EFCC performs slightly worse than FCC, but better than TF-IDF for vocabulary sizes from 100 to 1000. In WebKB, EFCC MFT achieves the best results in all cases except one, corresponding to $5,000$ features.

The case of EFCC LSI in WebKB is also interesting. When reducing to 100 features by means of LSI, the result is approximately the same than for the rest of vector sizes. Then LSI is finding almost the same independent components for all vector sizes. It is as if LSI had reached an upper bound and the number of vocabulary terms does not help to find more representative independent components. In this case, MFT reduction is able to find more representative features than LSI.

Globally, EFCC MFT offers the most stable results among collections, though in Banksearch it is not always the best alternative. Thinking in applying the representation to a new collection, EFCC MFT would be the best option. Our results shows that EFCC MFT is the best combination of weighting function and reduction method among collections. TF-IDF LSI is the only alternative, but at the price of a much higher computational cost and worse results in most of the cases when reducing to $1,000$ features or less.

Another idea appears looking at the results of EFCC. It seems like dealing with collections with homogeneous documents within categories that are also well balanced, EFCC MFT needs less terms to get to its maximum. In the case of collections composed of more heterogeneous documents within categories, that are also more unbalanced, the number of terms needed to obtain the best results with EFCC MFT is greater. This makes sense, because in a corpus with heterogeneous documents within categories, to find the most representative terms to differentiate the categories should be harder.

Furthermore, the additive properties of the fuzzy system make possible to reduce the number of rules needed to specify the knowledge base of EFCC and therefore, the system is easier to understand.

Throughout this section we have tried to answer the three issues we stated at the beginning of Section 4.5. First, EFCC MFT is able to achieve good performance in both collections. This has been possible after analyzing the worst performance of FCC in WebKB, by improving the combination on the basis of our findings. Finally, we have shown that a lightweight dimension reduction technique as MFT is able to obtain similar or even better results than LSI when used with the proper term weighting function. In fact, the good behavior of MFT depends on the term weighting function applied before. Because of this, we believe that the use of light dimension reduction techniques is a good alternative, at the price of selecting a proper term weighting function for the clustering problem we want to solve.

4.6 Criteria Beyond the Document Itself

Until now, we have used web pages content to represent them, but there exists other information that could be useful. This is the case of IDF and anchor texts, two widely used resources in clustering, as we described in Chapter 2. In this section we explore both alternatives, trying to improve EFCC by adding contextual information, i.e., information from sources external to the document itself.

4.6.1 Inverse Document Frequency

By adding IDF we try to include information coming from the whole collection to our representation proposal. In order to add IDF function (Equation 2.5) to EFCC we use a linear combination of both:

$$\text{EFCC-IDF}(t, d, D) = \text{EFCC}(t, d) \times \text{IDF}(t, D) \tag{4.8}$$

where t is a term, d a document, D the whole corpus.

The experimental settings are the same as in the previous sections. In this case we use EFCC for the comparison because it was the most stable fuzzy logic based alternative among collections in Section 4.5.2.

Table 4.11: F-measure results for efcc-idf experiments

Rep.\Dim.	**100**	**500**	**1,000**	**2,000**	**5,000**	**Avg.**	**S.D.**
Banksearch							
EFCC MFT	**0.768**	**0.778**	0.758	0.740	0.759	**0.760**	**0.014**
EFCC-IDF MFT	0.522	0.773	**0.799**	**0.825**	**0.827**	0.749	0.129
WebKB							
EFCC MFT	**0.516**	**0.546**	**0.545**	**0.566**	**0.484**	**0.532**	**0.032**
EFCC-IDF MFT	0.383	0.346	0.291	0.282	0.451	0.350	0.070

Looking at the Table 4.11, EFCC-IDF works really well above 500 features in Banksearch, but really bad with 100, probably due to the penalization that IDF causes to the most common terms. In Banksearch, categories are well defined in terms of homogeneity among category documents and document sources, as we showed in our category analysis in Section 3.2.2. Then, we can assume that most important terms will be those appearing in most of the documents of a single category. Some of these terms are probably receiving high EFCC scores, but at the same time, they are penalized by IDF. Thus, by reducing to only 100 features, the possibility of losing those terms increases, which affects the representation. As we incorporate more terms to the vocabulary, the results improve, because the representative terms that were penalized by IDF begin to appear step by step. This behavior is similar to the case of TF-IDF with MFT in Section 4.4.2, but much more accentuated by the combination of EFCC and IDF. As we said before, EFCC weights terms more aggressively than FCC and TF-IDF, since its rules highly benefit terms appearing in titles and emphasized. As the number of these terms is limited, the difference between these terms and the rest should be clear in terms of `importance`. Thus, if IDF penalizes some of the terms that appear in titles or emphasis, when we apply MFT they will fall from the top of the document rankings to intermediate positions, and when we reduce the initial vocabulary to 100 features, we will lose them.

However, the use of IDF strongly penalizes results in WebKB. In previous experiments with TF-IDF and MFT (see Section 4.4.2) we saw a penalization similar to the one we have analyzed in Banksearch. But here, the representation directly does not work. Only the case of biggest vocabulary size achieves reasonable clustering results. As we saw in Section 4.4.2, IDF strongly affects the weighting function in a negative sense for the case of WebKB. The reason can be found in the size of WebKB categories (see Section 3.3.1). The biggest one, having a 36% of total collection documents could suffer the effect of IDF on its most common terms. As mentioned above in the case of Banksearch, this situation is similar than the one analyzed before for TF-IDF MFT in WebKB (see Section 4.4.2), but much more accentuated, due to the more aggressive term weighting performed by EFCC in combination with MFT. EFCC assigns well differentiated weights to terms appearing in titles and emphasis with respect to the rest. Nevertheless, at the same time, some of these terms could be penalized by IDF due to their high document frequency as a consequence of the effect of the biggest collection categories. This effect is more accentuated in WebKB because most of its categories are bigger than Banksearch ones. Then, these terms would fall from the first positions in the rankings and we would need to introduce more features in the vocabulary to recover them, like in the case of $5,000$ features, where clustering results get better.

Summarizing, the combination of EFCC and IDF does not help to find a good clustering solution in all the cases. Although it can lead to very good results, it also can drive us to very bad ones. Furthermore, in unsupervised environments where we do not have any category information, a bad selection of the vocabulary size could have a terrible effect. Particularly bad is the case of WebKB, where most of the results are far away from the rest of representations. From a general point of view, IDF introduces a penalization for common terms. However, in clustering problems we search for common terms to group documents. The compromise of both things is not easy to find in unsupervised environments. As we have seen, there are cases where IDF could not help the clustering, but the opposite.

4.6.2 Anchor Texts

For this experiment we needed to employ a recently crawled collection, in such a way that it was easy to find other web pages with hyperlinks to the collection documents. We decided to use the SODP dataset—described in Section 3.4—composed of $12,616$ documents retrieved from social bookmarking sites and classified by extracting the category for each URL from the first classification level of Open Directory Project. Thus, the entire collection is divided in 17 unbalanced categories, having from 39 to 3,289 documents each. In addition to the documents themselves, we collected the anchor texts corresponding to a maximum of 300 unique inlinks per each document in the collection (2,704 web pages have less than 50 inlinks, 4,717 have less than 100, so the rest, approximately 60%, have more than 100 inlinks).

There are a number of ways of adding anchor texts to document representation methods. We are interested in elucidating whether anchor texts could help improve web page representation in clustering or not, but at the same time, we want to investigate different alternatives for the combination. Therefore, we decided to combine anchor texts with EFCC in two different ways:

(a) In addition to each document textual content, as other document terms, i.e., they are added to `frequency` criterion. The idea is to analyze whether anchor texts terms are similar to document content terms and, therefore, they contribute in the same way to the combination in order to estimate term importance. Other works like Wang and Kitsuregawa (2002) and Huang et al. (2006) followed a similar approximation.

(b) In addition to each document title, i.e., giving them the same importance than title terms in the combination. We aim to discover whether anchor texts are better suited than document content terms for clustering tasks. The format of anchor texts is also similar to document titles, since they usually are short texts.

Besides, we did three experiments for each case:

(1) Just adding anchor texts.

(2) Adding anchor texts and removing text corresponding to outlinks (links from a dataset page to other pages).

(3) Removing a set of stop words based on a study over collection anchor text terms. We created a list of terms appearing in anchor texts and looked at their document frequencies in the collection to find non-informative but frequent terms. This set contains words like click, link or homepage. The whole list can be found in Appendix D.

This way, our experiments are oriented to find out not only whether anchor texts are useful for web page representation in clustering tasks, but also to compare different manners to include this information in our combination proposal. We perform the experimentation using the same settings as before, except the dataset, since with Banksearch and WebKB was not possible to recover a sufficient number of inlinks pointing at collection pages. As we introduce here a new collection, we decided to add FCC as baseline to validate our previous EFCC results. We also include AddFCC in these experiments due to its good performance in Banksearch (Section 4.5.2). The main aim of this decision was to ensure that by adding anchor texts to EFCC, we were adding them to the best of our fuzzy combinations for this collection. Hence, we employ SODP not only to test the usefulness of anchor texts in our representation proposal, but also to validate the results of EFCC against the other fuzzy logic-based representations in a new dataset.

Table 4.12: F-measure results for anchor text experiments

Rep.\Dim.	**100**	**500**	**1,000**	**2,000**	**5,000**	**Avg.**	**S.D.**
SODP							
FCC MFT	0.195	0.237	0.254	0.256	0.266	0.242	0.028
AddFCC MFT	0.208	0.267	0.276	0.279	0.282	0.262	0.031
EFCC MFT	0.233	0.273	**0.287**	0.283	**0.296**	0.275	0.025
EFCC a-1 MFT	0.225	0.262	0.279	0.286	0.290	0.268	0.027
EFCC a-2 MFT	0.245	0.246	0.285	0.289	0.269	0.267	0.024
EFCC a-3 MFT	0.248	0.260	0.285	**0.294**	0.293	0.276	0.022
EFCC b-1 MFT	**0.254**	**0.287**	0.275	0.282	0.285	**0.277**	0.015
EFCC b-2 MFT	**0.254**	0.249	0.276	0.279	0.291	0.270	0.016
EFCC b-3 MFT	0.249	0.261	0.263	0.278	0.285	0.267	**0.012**

Table 4.12 shows the results of the different alternatives. Each one is followed by a letter and a number, corresponding to the way in which the anchor texts were added to the representation. The letters and numbers were defined above.

The first three table rows show that EFCC based approaches outperform FCC and AddFCC in almost all cases. Particularly, EFCC always outperforms FCC and AddFCC in SODP. This corroborates our findings about the drawbacks of FCC, stated in Section 4.5, and confirms our belief in the need of a system where not all the criteria contribute the same to the combination, in contrast to AddFCC.

Regarding the contribution of anchor texts, there is no clear alternative to improve EFCC in all cases. Looking at the averages, there are only slight differences among the different EFCC alternatives. Focusing our attention on concrete vocabulary sizes, anchor texts help improve clustering results with small vector sizes, particularly when anchor texts terms are considered as page titles (b alternative).

However, when we increase vector size, they seem to introduce noise, because clustering results get worse. About using anchor texts as titles, the best option is just adding anchor texts as title terms (named b-1). It is interesting to have found an improvement for smaller vector sizes. In the best case, this improvement is about 0.02 in terms of F-measure, which corresponds to about a 9% over the result of EFCC with the same vocabulary size. However, in some scenarios this improvement does not compensate for all the process needed to collect anchor texts.

A possible explanation for these results might be a poor link density or bad anchor text quality, or just the nature of clustering problems, where the aim is to capture the aboutness of documents. This conclusion coincide with other works like Eiron and McCurley (2003); Noll and Meinel (2008) (see Chapter 2, Section 2.2.1.3 on page 45), where authors conclude that anchor text terms are more similar to terms used in search queries. Also they experimentally found that these terms are not often in web page contents, and therefore this information is not so good for capturing the aboutness of web documents..

Finally, it is important to highlight that, in general, all the results are very bad in this collection. SODP is composed of 17 unbalanced categories and there is also a clear bias towards *Computers* category that contains a 26% of documents. Besides, the other 74% of documents is divided in 16 categories with different number of documents each. Terminology from bigger categories will favor the division of documents belonging those categories instead of finding smaller ones. Thus, finding a clustering for SODP collection that corresponds with the categories in the gold standard is very difficult.

4.7 Empirical Evaluation of the Proposed Modifications

In this section we perform a robust evaluation of EFCC to be sure about whether or not exists a real improvement over FCC. As we are using a deterministic algorithm, we want to avoid the possible bias introduced by feeding the algorithm with a single set of vectors for each dataset. The solution presented here involves dividing each dataset in 100 different sub-datasets 50% smaller than the original, where the categories are in proportion to the original ones. We performed 100 experiments per each vector size and each sub-dataset, resulting a total of 3,000 different clustering experiments. Due to computational reasons, we chose MFT reduction for all the experiments. This decision was also made to compare both term weighting functions in the exactly same conditions. Besides, MFT does not transform features, but selects those with higher weight, allowing us to study the effectiveness of each alternative to give more importance to the most representative terms.

Basically, we want to ensure that EFCC and FCC lead to different results. Nevertheless, the average value of a set of experiments it is not enough to ensure that one of them is better than the other. To make such a statement we decided to employ statistical significance analysis over the samples of both methods. In our case we use statistical hypothesis testing for paired samples, because our data come from applying both methods over the same datasets. In particular we employ a paired two-tailed t-test over the results obtained by both representations for each concrete vector size in the 100 sub-datasets. This test assumes that the error follows the normal distribution, but it often performs well even when this assumption is violated. The t-test is relatively robust to many violations of normality and only heavy skewness[6] or large outliers[7] will seriously compromise its validity (Hull, 1993).

The t-test is a statistical hypothesis test where the null hypothesis is that the values we want to compare are drawn from the same population. The alternative hypothesis depends on the type of t-test we apply. For two-tailed tests, the alternative hypothesis is that the values we want to compare come from different populations, that is, their means are different from each other.

More formally, for two populations X and Y that we want to compare and paired samples $(x_1, y_1), ..., (x_k, y_k)$ the hypothesis for the t-test would be:

- Null hypothesis: $\mu_X - \mu_Y = \Delta_\mu$
- Alternative hypothesis: $\mu_X - \mu_Y \neq \Delta_\mu$

where μ_X is the mean of X population, μ_Y the mean of Y population and Δ_μ is the difference between both means. Then, the t statistic is calculated as:

$$T = \frac{\overline{D} - \Delta_\mu}{S_D/\sqrt{n}} \tag{4.9}$$

where $\overline{D}$ is the sample mean of the differences, i.e., the differences between all pairs (Equation 4.10), Δ_μ is the difference between the populations means (Equation 4.11) and S_D is the standard deviation of the above mentioned differences (Equation 4.12):

$$\overline{D} = \overline{X} - \overline{Y} = \frac{1}{k}\sum_{i=1}^{k} x_i - y_i \tag{4.10}$$

$$\Delta_\mu = \mu_X - \mu_Y \tag{4.11}$$

$$S_D = \sqrt{\frac{\sum_{i=1}^{k} ((x_i - y_i) - \overline{D})^2}{k-1}} \tag{4.12}$$

[6] Lack of symmetry in the distribution.

[7] Observations that are numerically distant from the rest of the data.

By using the Student's T distribution with $k-1$ degrees of freedom, the value of T is converted in a probability value $P(T)$ (also noted as p-value): the probability of obtaining a difference like the observed $\overline{D}$, if the real difference between the means of the populations is Δ_{μ}.

If confidence in the hypothesis (reported as p-value) is lower than a significance level α, then it is common to assume that values are from different populations with likelihood greater than $1-\alpha$ to reject the null hypothesis (Sanderson and Zobel, 2005). It is usual to set $alpha$ to 0.05 or 0.01 as upper bound. Other works like Zhao and Karypis (2004); Higa and Tozzi (2008); Farahat and Kamel (2010) took a similar approach.

Table 4.13: F-measure results for EFCC/FCC t-test experiments

Rep.\Dim.	**100**	**500**	**1,000**	**2,000**	**5,000**
Banksearch					
EFCC MFT	**0.764**	**0.774**	**0.770**	0.760	0.753
FCC MFT	0.718	0.760	0.765	**0.768**	**0.768**
Difference	0.047	0.014	0.006	-0.008	-0.015
p-value	**0.000**	**0.000**	**0.002**	**0.000**	**0.000**
WebKB					
EFCC MFT	**0.487**	**0.514**	**0.528**	**0.534**	0.483
FCC MFT	0.446	0.462	0.470	0.485	**0.490**
Difference	0.041	0.051	0.059	0.049	-0.007
p-value	**0.000**	**0.000**	**0.000**	**0.000**	0.016
SODP					
EFCC MFT	**0.230**	**0.271**	**0.279**	**0.282**	**0.289**
FCC MFT	0.200	0.233	0.246	0.251	0.266
Difference	0.030	0.037	0.033	0.031	0.023
p-value	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

In Table 4.13, for each vector size and representation we show the average F-measure values corresponding to the 100 clustering experiments (one per each sub-dataset), the difference between the corresponding averages, and the p-value resulting of applying the statistical t-test between the samples corresponding to both representations. Attending to p-values, in all cases except one, we can say that values are from different populations with likelihood greater than 99%. In those cases the p-values are highlighted in bold font.

Besides, looking at the averages, in most of the cases EFCC outperforms FCC. Regarding differences between representations, just in three cases FCC performs better than EFCC, being the difference lower than 0.01 in two cases and lower than 0.02, that corresponds to an improvement of a 1.9% of FCC over EFCC, in the other. In the rest of the experiments EFCC gets an improvement over FCC, higher than 0.03 in SODP (corresponding to more than a 12%), and greater than 0.04 in WebKB (corresponding to about a 10% in all cases), and also with the smallest vector size in Banksearch (corresponding to about a 6%).

4.8 Conclusion

Throughout this chapter we have studied how to combine different criteria extracted from the information contained in HTML web pages to represent them for document clustering. We have explored the possibilities of fuzzy systems to help apply expert knowledge and combine these criteria, and tested our findings with three different datasets.

Parts of the research in this chapter have been published in Pérez García-Plaza et al. (2008) and Pérez García-Plaza et al. (2012a).

We believe that fuzzy combinations of criteria, as FCC, fits better the problem of establishing term importance, because there are dependencies among criteria that should be taken into account in order to deal with authors' writing style, automatism in web page creation (as titles automatically generated by HTML editors), etc. In general, to deal with heuristic knowledge about how to establish term importance by means of combining different criteria. These facts encourages us to explore the way on which criteria can be combined by means of a fuzzy system. Our work is oriented to propose a web page representation method that works reasonably well regardless the collection of documents. In this sense, achieving the best results is not the most important condition to fulfill. In our research, we consider more important to obtain stability among collections. That is to say, the same representation model (weighting function and dimension reduction technique) should achieve good results in different collections.

We studied different dimension reduction techniques in order to apply some methods that had not been used in the context of FCC. We have shown that with a good weighting function it is feasible to employ lightweight dimension reduction techniques, as the proposed MFT, instead of using other more complex techniques like LSI, which implies an important reduction in the computational cost.

Regarding LSI, MFT and DF, as the first one is computationally more expensive and it transforms the features—losing the reference between the original features and the final vector values—, we find MFT and DF more appropriate in order to study the weighting functions, as they keep the features as they are, allowing the direct analysis of the functions.

Another interesting issue is that DF showed to be useful in concrete circumstances, particularly when dealing with collections composed of heterogeneous web pages coming from a small number of web domains. In these cases, DF can help removing too common terms and web domain related terminology spread throughout the collection. This is the case of collections like WebKB, described in Section 3.3, where the probability of sharing terminology among categories is high, given the heterogeneity of documents within collection categories.

It is worth mentioning that RP method was also tested as a lightweight alternative to LSI, but our experimental results showed that it is not a good alternative.

Moreover, the experimental results of our initial tests on dimension reduction techniques showed the bad performance of FCC in WebKB collection. For that reason, we analyzed FCC, finding some issues that could cause its bad performance in WebKB dataset. We detected that some individual criteria performed better isolated than the FCC combination.

After identifying the aspects that, in our opinion, hinder FCC—the overestimation of `position`, underestimating at the same time the rest of the criteria as detailed in Section 4.5.2—, we proposed two alternative ways of combining criteria, AddFCC and EFCC, within the same fuzzy logic framework. Our experiments showed that EFCC worked better than FCC by means of a different way of combining criteria, where term frequency is considered as discriminant as title and emphasis, and position is taken into account as the least important criterion. This approach makes also possible to reduce the number of rules needed to specify the knowledge base taking advantage of the additive properties of the fuzzy system. Thus, it makes the system easier to understand. In contrast, AddFCC obtained very good results in Banksearch, but its clustering results in WebKB were bad. The problem of AddFCC comes from the way of combining criteria, where all the criteria contribute the same to the combination. This fact supports our belief in the need for a system where not all the criteria contribute the same to the combination.

In order to continue exploring new criteria for the combination, we considered to employ collection information and anchor texts to represent documents. First, we evaluated the possibility of a linear combination between EFCC and IDF, but we rejected this alternative based on the bad experimental results we found in some cases, particularly in WebKB. Second, we also used anchor texts to enrich document representation with contextual information. Although results were not bad, they were not clearly better than the ones obtained by EFCC. Besides, the cost of preprocessing anchor texts and their dependence on link density limit the applicability of this alternative. For these reasons, we believe that it could be an interesting option when a collection fulfills these requirements and time complexity is not a problem, but in most of the cases this will not happen and we will have to carry out document representation only with document contents.

Finally we performed statistical significance tests to ensure that the application of our findings in our representation proposal has a real effect compared to FCC, as both representations relies on the same fuzzy engine. We conclude that our proposed approach, EFCC, improves the results of FCC in most cases. It is particularly interesting the good results obtained with vocabulary sizes below $1,000$ features, since using smaller vocabularies allows to reduce the computational cost of the clustering algorithm.

5

Fitting Document Representation to Specific Datasets by Adjusting Membership Functions

Fuzzy ruled-based systems have been successfully used to represent web documents by means of heuristic combinations of criteria. In these systems, rules were established based on the way humans have a quick look at documents in order to extract the essential information, as we have analyzed in previous chapters. However, membership functions parameters were fixed by default, assuming that any document would follow similar patterns regardless of the rest of documents in the collection. In the current chapter we analyze to what extent collection information could be used to adjust the membership functions in order to improve document representation and, therefore, clustering results. We compare our proposal to the previous systems we have been working with in this thesis, particularly with the fuzzy ones in which it is based. Results show that adjusting document representation parameters to concrete collections leads to better clustering results when collections present particular characteristics.

The chapter is organized as follows. First, in Section 5.1 we briefly introduced the main ideas of the chapter. Next, in Section 5.2 we review works related with fuzzy rule-based system tuning. Then, in Section 5.3 on page 147 we describe the problem we study in this chapter, analyzing every aspect in detail and presenting our approach to deal with it. Next, we evaluate our proposal in Section 5.4 on page 156 and finally we present our findings to conclude the chapter in Section 5.5 on page 162.

5.1 Introduction

In previous chapters we have shown that, although using term frequencies is a common approach to represent documents, Fuzzy Rule-Based Systems (FRBSs) have become an effective alternative in order to exploit other additional information that HTML tags provide. Moreover, FRBSs can be tuned to fit a concrete problem, aiming to improve their results. This fact is especially interesting when we consider the problem of document representation, where each document collection may differ from the rest. For example, Banksearch (see Section 3.2.2 on page 80) and WebKB (see Section 3.3.2 on page 88) show clear differences between their term distributions, particularly for emphasized terms. Therefore, our initial hypothesis in this chapter is that some dataset characteristics could have an influence on the way of defining the FRBS for representing the documents belonging to that concrete collection. As we will see later in this chapter, these characteristics has to do with term frequencies, since they could allow to better capture the information related to each criterion.

Fuzzy modeling (FM) is problem modeling by means of FRBSs. One of the most important fields within FM is linguistic FM. As stated in Casillas et al. (2005): “the two main requirements in FM are *interpretability*, capability to express the behavior of the real system in a comprehensible way, and *accuracy*, capability to faithfully represent the real system”.

Throughout this chapter we analyze the possibility of adjusting a FRBS developed for web page representation to different datasets. We evaluate our results to elucidate whether this adjustment can improve clustering results or not. In addition we perform a statistical significance test to verify our results compared to those obtained by other FRBSs whose parameters are fixed by default.

5.2 Tuning of Fuzzy Rule-Based Systems

In this work we are interested in adapting a FRBS to improve web page representation for clustering tasks. Before going into further details it is worth to remember that the knowledge base of a fuzzy system is composed of the rule base, that is the set of rules, and the data base, which contains the membership functions associated to the linguistic variables.

Regarding FRBSs tuning aiming to extract a suitable set of fuzzy rules from numerical data, in Casillas et al. (2005) the authors presented a genetic tuning process for knowledge base refinement. They focus their work in maintaining a good trade-off between accuracy and interpretability by reducing the rule set in a first step and tuning the resulting system next. They conclude that this order is crucial to obtain good results in terms of efficiency and accuracy. They applied their method to two real world problems as benchmarks: the rice (Nozaki et al., 1997) and electrical (Cordón et al., 1999) problems, concluding that tuning and reduction processes can significantly improve the accuracy of a fuzzy model.

A similar approach focused in obtaining more compact models was presented in Gacto et al. (2008). To do so, they used Multi-Objective Evolutionary Algorithms (MOEAs) as a tool to get an improved solution with respect to a classic single objective approach. They also evaluated their method using two datasets about real world problems: the electrical problem, the same as the previously commented work, and the Abalone dataset (Waugh, 1995). Their results showed that an appropriate use of MOEAs can help obtain more accurate and simpler linguistic models than those obtained by only considering performance measures.

Again, in Li et al. (2010) one of the main tasks is to learn fuzzy rules from examples of the problems, but the aim of the authors was to avoid local convergence as a result of the increasing complexity and dimensionality in classification problems. They used a fitness sharing method based on the similarity level of each rule and its neighbors rules. Their method was studied for sonar signal classification (Ishibuchi et al., 2005) and hand movement recognition (Dias et al., 2009) problems. Their experimental results show an improvement over two classical genetic machine learning approaches, that are widely used in the construction of fuzzy rule based classification systems: Pittsburgh approach (Venturini, 1993), where the whole rule set is handled as a genetic individual, and Michigan approach (Booker et al., 1989), where each rule is considered an individual.

Another recent work exploiting MOEAs to generate FRBSs proposed to adopt partition integrity (interpretability of fuzzy partitions) as an objective of the evolutionary process (Antonelli et al., 2011). The authors introduced a three-objective evolutionary algorithm which generates a set of FRBSs with different trade-offs between complexity, accuracy and partition integrity by concurrently learning the rule base and the membership functions parameters. They experimented on six real-world regression problems, where their approach improved results of applying the same MOEA, but with accuracy and complexity as objectives only. They also compare their work with a similar approach proposed by Gacto et al. (2010), showing that their solutions were characterized by a better trade-off between complexity and accuracy.

A more complete review on the most representative genetic fuzzy systems relying on Mamdani-type FRBSs to obtain interpretable linguistic fuzzy models can be found in Cordón (2011).

There are two important differences between the above mentioned works and the problem we deal with in this chapter. First, there are no works oriented to web page representation among those oriented to FRBSs tuning. Second, all the above mentioned works employ real data samples in the learning process, that is, they employ an approach corresponding to the supervised classification field, since they use training information to adjust the system. In our case, we deal with datasets composed of web pages and we represent them from an unsupervised point of view, by utilizing only their own contents.

Thus, although the previous approaches are not oriented to document representation for clustering, but to classification tasks, it is interesting for this dissertation to briefly summarize how the fuzzy systems are modified to fit concrete problems.

Different from the above mentioned works, in document clustering we do not have category information. However, it is possible to analyze the documents we want to cluster to find common patterns or particular features that can help improve the way of capturing criteria information. This work focuses on adapting membership functions to dataset concrete features, which could enhance document representation, leading to improve clustering results.

Tuning of Membership Functions

In general, there are two main ways of tuning membership functions:

- Changing basic parameters (Casillas et al., 2005; Gacto et al., 2008; Li et al., 2010) (see the left side of figure 5.1). It involves varying the shape of the fuzzy set associated to the membership function by modifying these parameters (a, b, c and d in the figure 5.1).

- Using non-linear scaling factors Liu et al. (2001); Casillas et al. (2005) (see the right side of figure 5.1). This approach is based on changing the membership function in a nonlinear fashion, but does not modify the basic parameters and, when dealing with symmetrical fuzzy sets, their centers of mass do not change. Some works like Li et al. (2010) used the most similar triangular shapes to replace the curve shapes (this work was reviewed in the previous section 5.2).

It is worth mentioning that both tuning alternatives are not exclusive but complementary.

In this chapter our goal is to automatically adjust a FRBS to better capture the information of the documents we want to represent and cluster. The main

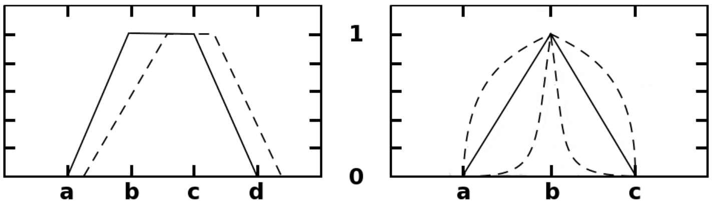


Figure 5.1: Left: Example of changing basic membership function parameters (a, b, c and d in this case); Right: Example of tuning by using non-scaling factors. Dashed lines show possible results of each type of tuning.

difference with previous works is that this adjustment must be done before the clustering process, so the information available is that contained in dataset documents themselves. In this scenario, our hypothesis is that fuzzy sets can be adjusted to better fit input data, in order to improve the process of information capture. The effect of this improvement should be reflected in document representation and, therefore, in clustering results. As a mean to do that, we explore the effect of modifying fuzzy sets basic parameters taking into account the particular characteristics of each dataset. We analyze statistical data of each criterion within every dataset in order to find these particular characteristics and adjust the system. The use of non-linear scaling factors is out of the scope of this thesis, since it requires an objective function to maximize while performing the adjustment, and these functions are usually based on category information employed for training.

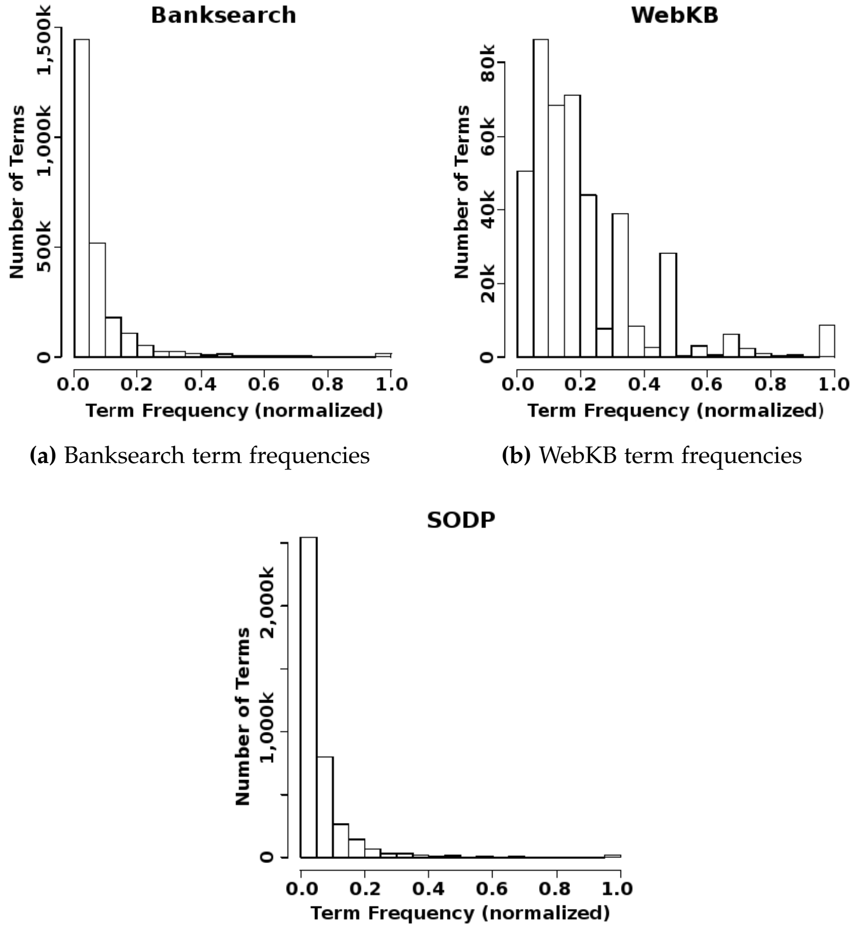
5.3 Problem Analysis

As we see in Chapter 4, both systems, FCC and EFCC, use the same membership functions (see figure 4.3), where the input frequencies for each criterion are normalized using the maximum number of appearances of a term in the corresponding criterion within the document (since we want to grant independence to the rules regarding the document size). In order to study the problem of adapting document representation to concrete datasets, we first analyze three different document collections to find differences and common patterns and then we study the original membership functions to discover if they could fit better each dataset, proposing an automatic way of adjusting them.

5.3.1 Datasets

In the previous chapter we performed several experiments with our fuzzy systems on three different datasets: Banksearch, WebKB and SODP. Our first step here is analyzing them to check whether they follow similar patterns or not. In

Chapter 3 we analyzed the datasets one by one, showing their main features and giving a general idea of their differences and similarities. Nevertheless, we did not do a direct comparison among them. In this section we come back to the term distribution analysis in order to fill this gap. As we tackle the problem of web page representation from an unsupervised point of view, we are interested in term distribution analysis. Category analysis is based on supervised information, so it remains out of the scope for adjusting the system. In this thesis, this kind of information—supervised information—is used only with evaluation or result analysis purposes.



(a) Banksearch term frequencies

(b) WebKB term frequencies

(c) SODP term frequencies

Figure 5.2: Comparison of normalized term frequencies for `frequency` criterion in Banksearch, WebKB and SODP.

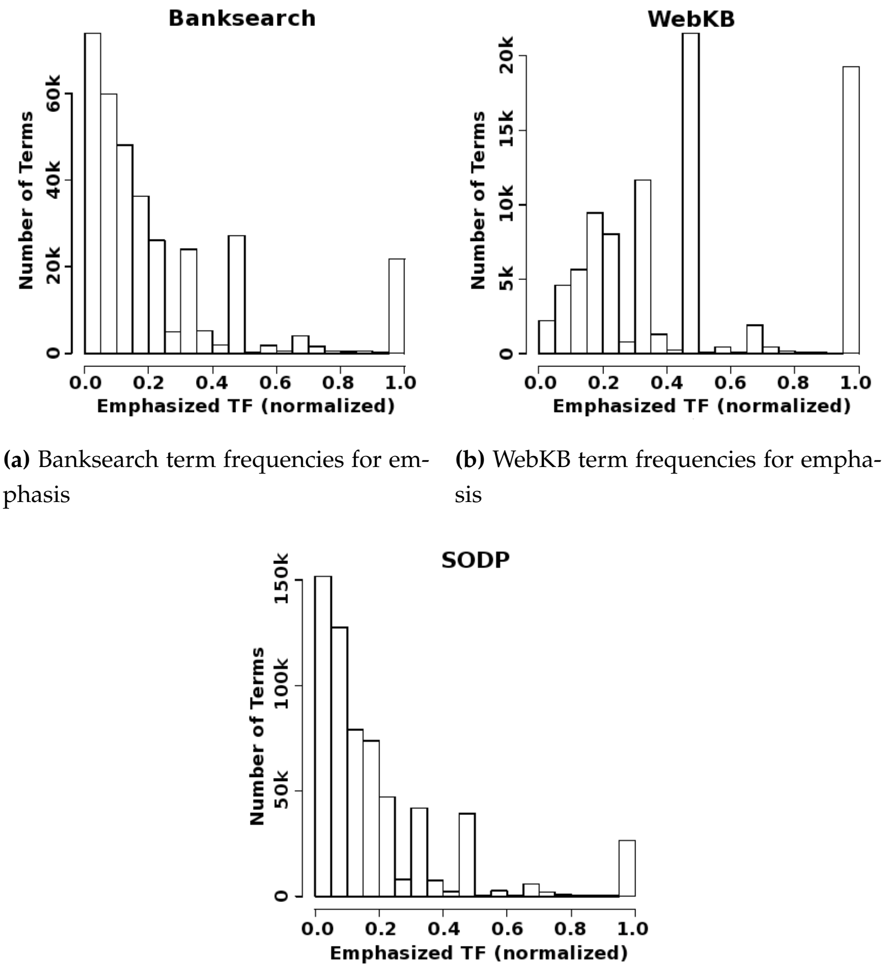
Figure 5.2 shows normalized term frequencies—corresponding to `frequency` criterion—in the three collections. Each bar represents the amount of terms having a concrete normalized frequency. All of them show a tail of terms with high frequency, but WebKB seems to have a different distribution under 0.5 value. In WebKB there are more terms with intermediate frequencies, that is, not as far from document maximum values than in Banksearch or SODP. Thus, the long tail we found in Banksearch and SODP from frequencies above 0.2 is not clearly visible in WebKB, since the bars in the distribution do not always decrease with respect to the immediately previous one. However, if we divide the space between 0.2 and 1 in quartiles, we found almost the exact same values for Banksearch, SODP and WebKB. Then, the number of terms between 0.2 and 1 follow a similar proportion among quartiles in all the collections. In the case of WebKB the tail could be not seen as clearly as in the other cases due to smaller maximum values per document, that lead to a smaller set of possible values after normalization. Having less possible values, we can expect to find the term frequencies more concentrated in concrete points, like in Figure 5.2b. This way, tails look different in the figures, yet splitting them in quartiles shows that they are not so different at certain degree of detail.

Due to the scale imposed by the normalization process, terms having low frequency values should be considered as noise because all of them are far away from document maximum value and there is no way to difference among their representativeness due to the high term density in that area. At the same time, in Banksearch and SODP there are few terms with high frequencies, probably because document maximum values are far away from the rest.

Figure 5.3 shows normalized frequencies for emphasized terms—`emphasis` criterion—in the three collections. In this case the Banksearch and SODP datasets show very similar distributions, but they are totally different compared with WebKB distribution. In our opinion, this fact implies that emphasis have been used by web page authors in a different manner. Emphasized terms have higher frequencies than in other collections. Besides, the bar corresponding to a frequency of 0.5 is clearly the larger one. This indicates that there are a small set of possible values within each document, that makes the distribution tend towards the maximum. This reflects a more restricted use of emphasis in WebKB, where less terms are emphasized and that emphasis is probably more meaningful.

Finally, figure 5.4 shows normalized frequencies for title terms, inputs for `title` criterion, in the three datasets. In this case, as titles are usually short text strings, there are not much different possible values. In most cases, the whole set of title terms of a document will appear just once, so the whole set will have the maximum frequency. The second most probable case will be when the term with maximum frequency appears twice, and then there will be two possible values: 1 and 1/2. Even with these extreme conditions, Banksearch and SODP look more similar between them than compared with WebKB.

Summarizing, these datasets allow us to verify that different document col-



(a) Banksearch term frequencies for emphasis

(b) WebKB term frequencies for emphasis

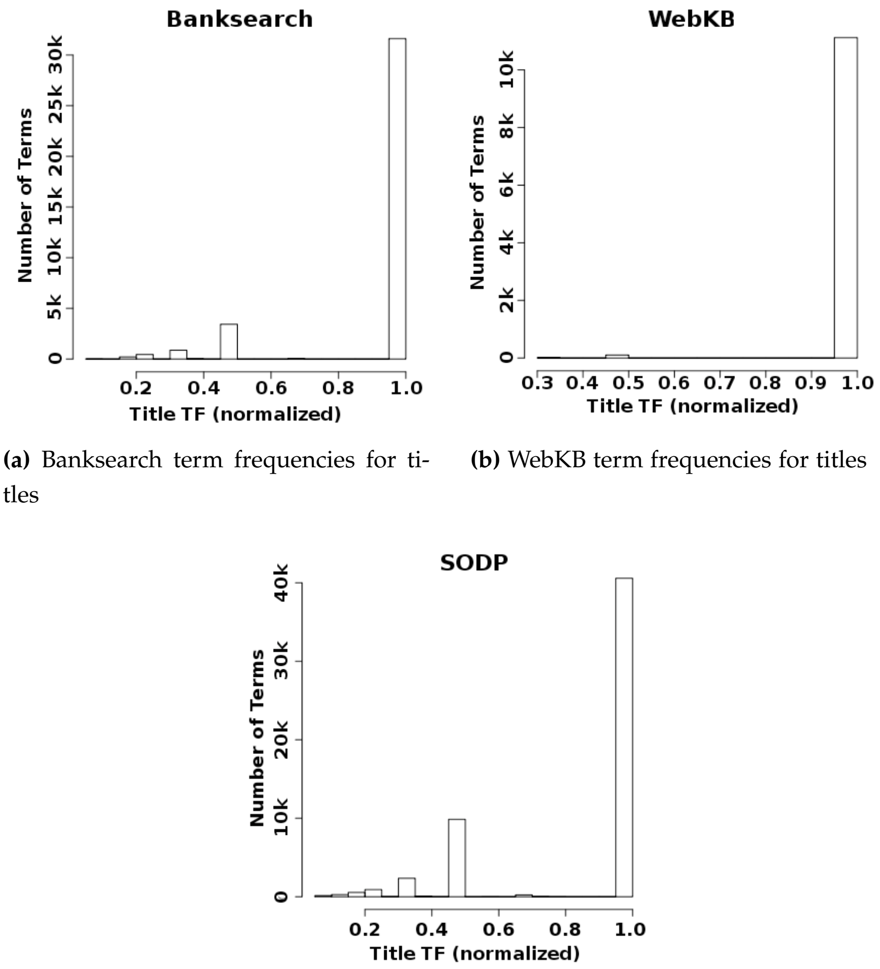
(c) SODP term frequencies for emphasis

Figure 5.3: Comparison of normalized term frequencies for `emphasis` criterion in Banksearch, WebKB and SODP.

lections can have different features that should be taken into account when representing their documents to establish the importance of their terms in each criterion. Banksearch and SODP presents different features than WebKB, which is the most different one. Banksearch and SODP tend more clearly to exponential distributions for `frequency` and `emphasis`. But WebKB, in particular for `emphasis`, tends more to a uniform distribution.

How to interpret these distributions is a key factor for adjusting the membership functions:

- Exponential distributions tell us that most of the terms are indistinguishable in terms of frequency, since they are too far from document maximum.



(a) Banksearch term frequencies for titles

(b) WebKB term frequencies for titles

(c) SODP term frequencies for titles

Figure 5.4: Comparison of normalized term frequencies for `title` criterion in Banksearch, WebKB and SODP.

In this case the first goal is to separate these terms from the rest, that will be the really representative ones. Then, we could establish different importance levels for the rest by dividing them in a relative manner with respect to the maximum.

- A more uniform distribution, like the case of `emphasis` in WebKB, points out in the opposite direction, i.e, there are more representative terms, since their term frequencies are higher compared to the maximum. As we commented above, this kind of distribution corresponds to a small set of possible values within each document (probably due to smaller maximums). This small set of possible values in turn reflects a more restricted use of emphasis in WebKB, where less terms are emphasized and that emphasis is probably more meaningful.

In contrast, for titles there is a small set of possible values and then frequency distributions look very similar in all cases.

5.3.2 Analysis of the Membership Functions

One question that needs to be answered is whether considering the same partitions in membership functions regardless the dataset is the best option when dealing with term frequencies in a document. Moreover, these sets were defined taking into account the possible input values, i.e., frequencies from 0 to 1, and not the concrete input values corresponding to the term frequencies for each criteria in a given collection.

We saw in section 5.3.1 that different datasets could have different frequency distributions for each criterion. Terms are commonly distributed in such a way that most of them have very low frequency and, as the frequency increases, the number of words having those frequencies in the document decreases. Besides, the effect of normalizing frequencies—to the maximum frequency of a term in the document for each concrete criterion—is that low values are compressed, making impossible in practice to make any distinction among their representativeness, further than all of them are far from the maximum on each corresponding document. Notice that this compression effect would be even worse if the normalization process had been performed by using the total maximum in the collection or the sum of all the term frequencies for a criterion in a document.

Looking at the original fuzzy sets defined for FCC and EFCC (Figure 4.3 on page 110) and comparing them with the frequency distributions extracted from Banksearch, WebKB and SODP (Figures 5.2, 5.3 and 5.4), it seems that those sets does not fit the tails as much as they could, in the sense that they do not take into account the number of terms in the collection belonging to those sets. For instance, the long tail in Banksearch and SODP term frequency distributions could lead to consider in the high fuzzy set only the terms with maximum frequency value in each document.

The fuzzy sets for FCC and EFCC were defined as symmetrical, except for emphasis. In fact, symmetrical sets are also defined as the initial state of most of the FRBSs tuning processes. Thus, some of the fuzzy sets defined for FCC and EFCC coincide with this initial state. As we saw in Section 4.2.1, `emphasis` is considered separately because when the maximum frequency value for emphasized words in a document is small, the normalization could mislead us about the importance of other emphasized terms. For this reason, the sets for `emphasis` were asymmetrically defined (see Figure 4.3c), with bigger *medium* and *high* sets and a smaller *low* set, i.e., the *low* set for `emphasis` will only include terms with small emphasized frequency that also correspond to documents with high maximums.

Moreover, we believe that what we call *high* or *low* are not absolute values, but relative values. This is the main point to capture criteria information in a better way. A term is not very important because its normalized frequency is 1, it is important because its normalized frequency is higher than most of the rest of term frequencies. This way, what we consider *high*, *medium* or *low* should depend on the concrete frequency distribution of the dataset.

In an ideal case, all term frequencies would be uniformly distributed between 0 and 1, and then we could configure the fuzzy set basic parameters using the original heuristic information, e.g., with equal size sets for `frequency` (see Figure 4.3a on page 110), because of the relative difference among frequencies, which would be uniformly distributed.

Nevertheless, we are working with texts, so in most of the real cases their term frequency distributions will tend to follow Zipf's law, that establishes an inversely proportional relationship between the frequency of a term and its ranking in the frequency table. Thus, the frequency for the term with maximum number of occurrences in a document will be approximately the double of the frequency corresponding to the second most frequent term, and so forth (Manning and Schütze, 1999).

Actually, we could expect to find different term distributions depending on different aspects like thematic divergence among categories, web page author's style, web domain of pages in the dataset, etc. Some of these aspects were analyzed in detail Chapter 3 and reviewed in the previous section. As we saw, though tending to Zipf's law in most cases, we could also find distributions in some point between the ideal case, exposed above, and a power law distribution like stated in Zipf's law.

At this point, it is clear that the question that opened this section needs to be clarified. Each particular dataset will have its own features and membership function tuning could be a useful way to improve the information capture process. As the input data patterns are not always the same, we strongly believe that the way of capturing criteria information can not be always the same. More than that, even if the original fuzzy sets defined for FCC and EFCC were valid for all the cases, it would be interesting to be able to obtain them in a more automated way.

5.3.3 Tuning of the membership functions

In previous sections we talked about the input criteria and the shape of their frequency distributions. Now we focus our attention in how to adapt our fuzzy system to these distributions in order to capture the input information in a better way.

In order to automatically adjust the membership functions basic parameters, we assume two base cases in our hypothesis:

1. Text in web pages follows the Zipf's law. Figure 5.2 shows that a long queue appears in the three collections we are working with, with most of their terms falling between 0 and 0.2 term frequency values.

2. Web page authors sometimes want to stress concrete terms they consider important to understand the document. This is the case of emphasized terms, whose frequency distribution can be totally different depending on how the author use them. Looking at figure 5.3 we can see the case of Banksearch and SODP, with emphasized term frequencies following a distribution more similar to a power law, and the case of WebKB, with a totally different distribution, due to, in our opinion, a more restricted and meaningful use of emphasis.

The first base case corresponds to `frequency` criterion. We consider we have a distribution tending to a power law when the majority of terms, i.e., more than a half of them (55%) have normalized frequencies below 0.2. Depending on whether this condition is fulfilled or not, we set the membership functions as follows:

- When the precondition is fulfilled, we assume a distribution tending to a power law, with low frequencies for the most of the terms. As we need 5 intervals to build three sets (*low*, *medium* and *high* and two intersection areas between them, see Figure 4.3a), our worst case would be to have only one possible value for each interval, that is, a maximum frequency of 5. With a maximum of 5, there would be just one possible value for each interval, as we would have only 5 possible values. Thus, to guarantee at least one possible value for the low set in that case, we chose the first interval from 0 to 1/5. This aims to separate noise from important terms. Finally, the rest of the intervals are selected using equidistant percentiles for term frequencies from 1/5 to 1, because in this way we consider high frequencies relative to intermediate ones, excluding terms with low representativeness, which we call here noise and would correspond to term frequencies below 1/5. Of course, it would be possible to have a maximum value smaller than 5, but taking our precondition into account, we are sure than more than 55% of terms in the collection belong to documents with maximum above 4, because their normalized frequency is smaller than 0.2. Moreover, small maximum values will correspond to short documents. Our system will give more importance to terms belonging to short documents, since the frequency of these terms will be closer to the maximum than in longer documents. Therefore, we can expect the behavior of our system will be reasonable even in those cases.

- When our precondition is not fulfilled, then we assume the distribution tends to be more uniform, so we can establish the fuzzy sets with the original heuristic, that is, all of them will have equal size. We use the corresponding percentiles to fit the distribution slightly better than using exact values (0.2, 0.4, 0.6, 0.8, see Figure 4.3a on page 110). Notice that in case of a uniform distribution, the adjustment corresponding to the first case—distribution tending to Zipf's law—would lead to these exact values too, because as the distribution moves towards a uniform distribution, the percentile 0.2 will approximate to $1/5$ and the rest of the parameters correspond to equidistant percentiles relative to this initial value in both cases. In those cases, the fuzzy sets would be symmetrical, that it is not only the case of the original sets in FCC and EFCC, but also the initial case used by most of the FRBS tuning methods we reviewed previously.

About the `emphasis` criterion, we follow the same precondition as with `frequency` to determine whether or not the distribution tends to a power law, but modifying the fitting rules due to the different meaning of emphasis. It is worth mentioning that the original intervals used for the emphasis sets were not of equal size. This decision was made because of the meaning of emphasis, which is used by authors to stress terms, so its use will be more restricted. Again, we have two alternatives for `emphasis`:

- When the distribution tends to a power law, we set the first interval as in the `frequency` case, and the rest with decreasing percentiles, each one a half of the previous. The reason is that in the original heuristic-based fuzzy sets, the medium set is the biggest one, and we want to preserve the original heuristic knowledge, but always taking into account the relative difference between the number of elements in each set instead of absolute exact values, as we explained in Section 5.3.2.

- If the collection do not fulfills our precondition, we assume the distribution tends to be more uniform, so we can establish the membership functions basic parameters by using the original heuristic rules but, as in the case of `frequency`, we use the percentiles instead of the exact values to fit slightly better the distribution (in this case the values were 0.05, 0.15, 0.55, 0.75, see Figure 4.3c on page 110).

The special case of titles was also mentioned in section 5.3.1, and actually there are not much possibilities to adjust their membership functions. We just try to do it taking into account the worst case in which we have a possible value for each interval. As this is an extreme case, we use the lowest value of the distribution to set the first interval, dividing the rest of the space in equidistant percentiles. The idea here is to guarantee that the low set will have at least one value despite of the small number of possible values.

Finally, it must be noted that we do not adjust the auxiliary system because word positions on a page do not depend on anything else than the number of words in the document.

Summarizing the ideas behind our proposal, we believe that membership functions should change depending on the inputs in order to keep their correct meaning, because the distribution of term frequencies could be different on each dataset. As the meaning of each criterion is different, we use different approaches for each of them. We have seen that this kind of distributions raises two problems, specially when they tend to a power law. First, words with low frequencies are usually not representative enough for the document theme, so establishing the low fuzzy set is crucial to remove noise in the final representation. Second, and supposing we are able to remove noise with the low set, we have to configure the rest of the intervals. Our method sets the basic parameters of membership functions to fit distributions that tends to follow the Zipf's law. At the same time, as these distributions tend to have more uniformly distributed frequency values, our system progressively adapts its membership functions towards the case where all the frequencies are uniformly distributed.

5.4 Empirical Analysis

The experimental settings we use in order to evaluate our proposal are the same as in the previous chapter. As we present a modification over the fuzzy logic based approaches, it is natural to use FCC and EFCC as baselines. In addition, TF-IDF will be also used. Thus, we start our research using these three weighting functions in our experiments in addition to our new proposal, called here Abstract Fuzzy Combination of Criteria (AFCC). We apply MFT dimension reduction technique in all cases in order to compare the weighting functions in the same conditions. We also take into account the conclusions of the previous chapter about the fact that MFT reduction is suitable in order to study the weighting functions, as it keeps the features as they are, selecting those with higher weights. This allows the direct analysis of the weighting functions.

Table 5.1 shows F-measure results for these representations, with the best ones for each vector size in bold font. Averages and standard deviations for each method are also shown in the table.

On the one hand, looking at the results, among the fuzzy logic based representations, AFCC outperforms the rest in WebKB in all cases, while in Banksearch got better results than the others with 2 out of 5 vector sizes, having also a higher average F-measure. This different performance between collections could be due

Table 5.1: F-measure results for membership functions experiments

Rep.\Dim.	**100**	**500**	**1000**	**2000**	**5000**	**Avg.**	**S.D.**
Banksearch							
TF-IDF MFT	0.703	0.737	0.768	**0.772**	0.758	0.748	0.028
FCC MFT	0.723	0.757	0.768	0.765	**0.768**	0.756	0.019
EFCC MFT	**0.768**	0.778	0.758	0.740	0.759	0.760	**0.014**
AFCC MFT	0.767	**0.785**	**0.787**	0.757	0.753	**0.770**	0.016
WebKB							
TF-IDF MFT	0.385	0.438	0.466	0.498	0.513	0.460	0.051
FCC MFT	0.453	0.472	0.475	0.468	0.475	0.469	**0.009**
EFCC MFT	0.516	0.546	0.545	0.566	0.484	0.532	0.032
AFCC MFT	**0.528**	**0.580**	**0.579**	**0.589**	**0.549**	**0.565**	0.025
SODP							
TF-IDF MFT	**0.244**	**0.300**	**0.293**	**0.307**	**0.323**	**0.293**	0.030
FCC MFT	0.195	0.237	0.254	0.256	0.266	0.242	0.028
EFCC MFT	0.233	0.273	0.287	0.283	0.296	0.275	0.024
AFCC MFT	0.233	0.269	0.292	0.284	0.282	0.272	**0.023**

to the fact that Banksearch frequency distributions tends to power law, and in those cases, once the less important terms have been assigned to the low fuzzy set, there are not so much terms to assign to medium and high sets, so the difference with EFCC and FCC fixed sets is small. Comparing AFCC with EFCC, it gets better results in Banksearch in 3 out of 5 cases, while with 100 features both performs similarly, and with 500 and 5,000 features the difference between them is smaller than 0.01, that would correspond to an improvement smaller than a 1% in both cases. The same occurs in SODP, where EFCC and AFCC get similar results, though FCC performs worse, probably due to its underestimation of `frequency` (see Section 4.5.2 on page 127). However, when term frequency distribution is more uniform, as in the case of WebKB, adjusting the fuzzy sets has a much bigger effect in clustering results, because there are much more terms to be distributed between medium and high fuzzy sets, and small variations of the membership functions basic parameters will have a much bigger effect. More than that, it is important to adjust this kind of distributions, because they reflect a different use of terms and other resources to highlight them, so the way of capturing this information must be changed too.

On the other hand, TF-IDF obtained surprisingly good results in SODP compared to the results of the same function with Banksearch and WebKB datasets. In general, all the representations get bad results in SODP, due to the special difficulties of this collection. The differences among category sizes are a key factor for clustering this dataset (see Figure 3.11 on page 91). As we saw in the previous chapter, there is a clear bias towards *Computers* category, that contains a 26% of collection documents, while from the other 16 categories, 15 contain less than a 10% of collection documents each, and 11 contain less than a 5% of documents each. We believe that the use of IDF could help improving results of TF-IDF because it would alleviate the effect of the bigger categories, whose terms would be penalized giving more representativeness to those belonging to smaller categories. This fact would slightly reduce the bias introduced by the bigger categories, allowing to cluster the smaller ones slightly better. This improvement in the clustering of smaller categories could cause the consequent improvement in the overall clustering results of TF-IDF.

Looking at AFCC and EFCC in Banksearch with 2,000 features and in WebKB with 5,000, AFCC improves the results of EFCC. In these cases EFCC shows a fall on its performance. A reason for this behavior could be that EFCC is more sensible to noise when the vocabulary size increases. The different information capture process of AFCC seems to overcome this fall.

In general, adjusting the membership functions to the dataset seems to be useful not only to add more automatism to the document representation process, but also because this automation allows to the system a better adaptation to datasets with specific characteristics. The proposed method is able to achieve similar results to EFCC when dealing with exponential distributions. Moreover, when the distribution shape changes, the adjustment helps improve clustering results, as is the case of WebKB.

5.4.1 Statistical significance

To sum up the previous section, AFCC performed well in two collections, Banksearch and WebKB—obtaining particularly good results with WebKB dataset—and in the third one, SODP, it obtained similar results than EFCC. As our proposal is a modification of a previous representation method (EFCC, described in Chapter 4), we decided to perform a robust evaluation of AFCC to be sure about whether or not exists a real improvement over EFCC. In other words, we analyze in depth the difference between using membership function tuning as described in section 5.3.3 and the original representation with fixed fuzzy sets. This analysis aims to check the conclusions stated at the end of previous section about when our tuning method contributes to improve the web page representation.

Besides, we also include FCC in the comparison. It was utilized in the previous chapter (Section 4.7) for a similar analysis together with EFCC. At this point, it is interesting to see the global benefits or drawbacks of the new proposal, AFCC, with respect to the original baseline.

To this end, we employ a similar approach than in the previous chapter. We performed 100 experiments per each vector size corresponding to each sub-dataset, resulting a total of 4,500 different clustering experiments. We calculated the statistical significance between F-measure results of each pair of representations (AFCC-FCC, EFCC-FCC, AFCC-EFCC), utilizing the same procedure described in Section 4.7 on page 136, that is, basically, a paired two-tailed t-test for each concrete vector size.

Table 5.2: Results for AFCC/EFCC/FCC t-test experiments

		100	**500**	**1000**	**2000**	**5000**
Banksearch						
F-measure	AFCC MFT	0.759	**0.776**	**0.776**	0.765	0.760
	EFCC MFT	**0.764**	0.774	0.770	0.760	0.753
	FCC MFT	0.718	0.760	0.765	**0.768**	**0.768**
Difference	AFCC-FCC	0.041	0.016	0.011	-0.003	-0.007
	EFCC-FCC	0.047	0.014	0.006	-0.008	-0.015
	AFCC-EFCC	-0.005	0.003	0.005	0.005	0.008
p-value	AFCC-FCC	**0.000**	**0.000**	**0.000**	0.039	**0.000**
	EFCC-FCC	**0.000**	**0.000**	**0.002**	**0.000**	**0.000**
	AFCC-EFCC	**0.000**	0.092	**0.003**	**0.001**	**0.001**
WebKB						
F-measure	AFCC MFT	**0.489**	**0.538**	**0.540**	**0.572**	0.485
	EFCC MFT	0.487	0.514	0.528	0.534	0.483
	FCC MFT	0.446	0.462	0.470	0.485	**0.490**
Difference	AFCC-FCC	0.043	0.076	0.070	0.087	-0.004
	EFCC-FCC	0.041	0.051	0.059	0.049	-0.007
	AFCC-EFCC	0.002	0.025	0.011	0.038	0.002
p-value	AFCC-FCC	**0.000**	**0.000**	**0.000**	**0.000**	0.122
	EFCC-FCC	**0.000**	**0.000**	**0.000**	**0.000**	0.016
	AFCC-EFCC	0.512	**0.000**	**0.001**	**0.000**	**0.002**
SODP						
F-measure	AFCC MFT	**0.235**	**0.274**	**0.280**	**0.284**	**0.289**
	EFCC MFT	0.230	0.271	0.279	0.282	**0.289**
	FCC MFT	0.200	0.233	0.246	0.251	0.266
Difference	AFCC-FCC	0.035	0.040	0.033	0.033	0.023
	EFCC-FCC	0.030	0.037	0.033	0.031	0.023
	AFCC-EFCC	0.005	0.003	0.000	0.001	0.000
p-value	AFCC-FCC	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	EFCC-FCC	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	AFCC-EFCC	**0.001**	**0.002**	0.668	0.059	0.586

In Table 5.2, for each vector size and representation we show the average F-measure values corresponding to the 100 clustering experiments (one per each sub-dataset), the difference between the corresponding averages, and the *p*-value resulting of applying the statistical t-test between each pair of representations.

First, attending to p-values, in all cases except 5 (out of 15), we can say that F-measure results of AFCC and EFCC are from different populations with likelihood greater than 99% (we use bold font to highlight p-values confirming this fact). Besides, looking at the averages of the F-measure, in most of the cases AFCC outperforms EFCC. Regarding differences between representations, just in one case EFCC performs better than AFCC, being the difference lower than 0.01, which would correspond to an improvement smaller than a 1% over AFCC.

In particular, AFCC gets an improvement over EFCC in the case of WebKB. We commented the same fact in section 5.4, so detecting the same behavior in this more exhaustive evaluation confirms our previous conclusions. Therefore, the difference between term frequency distributions of the datasets, in combination with all of these results allow us to conclude that membership function tuning helps capture criteria information in a better way, which improves clustering results.

About the comparison between EFCC and FCC, it was previously carried out in Section 4.7. With respect to the comparison between AFCC and FCC, the conclusions exposed above are also applicable, because of the improvements of AFCC over EFCC are also translated as improvements over FCC, as it results logical due to the improvement that EFCC obtained over FCC. It is interesting to highlight that this improvement of AFCC over FCC in WebKB means, in 3 out of 5 cases, a difference above 0.07 in terms of F-measure, that leads to improvements higher than a 15% of AFCC over FCC. In fact, the case with the smallest difference between AFCC and FCC in WebKB, AFCC gets an improvement about a 10%.

There is a strange case in WebKB for AFCC with $5,000$ features. The result for this case is very bad compared to the rest of AFCC results in WebKB. Nevertheless, the sub-datasets are a 50% smaller than the original one. Then, their initial vocabularies should be also considerably smaller. As the size of the initial vocabulary in WebKB is $40,258$ terms, the initial vocabulary of its sub-datasets could be about $20,000$ and $30,000$ terms. By reducing from those sizes to $5,000$ is probably not enough to remove noise. For this reason, AFCC (and also EFCC) could suffer a performance drop.

So far, we know that AFCC is able to obtain statistically significant better results than EFCC in most cases, concretely in 9 of 15, while in 5 cases both alternatives perform the same, and just in 1 case EFCC outperforms AFCC. However, averages are often confusing, because they do not show the distribution of values that they come from. For this reason, in Table 5.3, we compare all the individual cases ($1,500$ per each pair of representations, $4,500$ comparisons in total) we employed to calculate the statistical significance among FCC, EFCC and AFCC. The first column indicates the difference between F-measure results of each pair of representations for the data on a given row. Then we show three blocks, one for each pair of representations in the comparison: *EFCC vs FCC*, *AFCC vs FCC* and *AFCC vs EFCC*. Each block is divided in two different columns. The first one, entitled "*# cases*" shows the number of cases in which we find the difference corresponding to a given row. The second column, "*% cases*" shows the percentage that these cases represents over the total number of cases. For example, if we look at the first row, labeled "< -0.09", the column "*# cases*" indicates that there were only 3 cases in which the difference between EFCC and FCC was lower than -0.09, that is, FCC outperformed EFCC with a difference greater than 0.09 in 3 cases out of $1,500$, corresponding to a 0.2% of the total number of cases, as shown in the next column of the same row, that is "*% cases*" within the block *EFCC vs FCC*. Finally, the table has a horizontal line to separate the negative and the positive cases. When the difference is below 0, it means that the second representations of the corresponding pair outperforms the first. When the difference is greater than 0, then the first one outperforms the second.

Table 5.3: Results summary for sub-datasets experiments.

Difference	**EFCC vs FCC**		**AFCC vs FCC**		**AFCC vs EFCC**	
	# cases	**% cases**	**# cases**	**% cases**	**# cases**	**% cases**
< -0.09	3	0.2	1	0.1	1	0.1
< -0.08	6	0.4	2	0.1	1	0.1
< -0.07	14	0.9	5	0.3	3	0.2
< -0.06	19	1.3	8	0.5	8	0.5
< -0.05	25	1.7	10	0.7	17	1.1
< -0.04	29	1.9	13	0.9	23	1.5
< -0.03	38	2.5	19	1.3	40	2.7
< -0.02	67	4.5	38	2.5	83	5.5
< -0.01	155	10.3	126	8.4	202	13.5
< 0 and > -0.01	155	10.3	120	8.0	438	29.2
< 0	310	20.7	246	16.4	640	42.7
> 0	1190	79.3	1254	83.6	860	57.3
> 0 and < 0.01	156	10.4	112	7.5	432	28.8
$>= 0.01$	1034	68.9	1142	76.1	428	28.5
$>= 0.02$	882	58.8	978	65.2	282	18.8
$>= 0.03$	690	46.0	756	50.4	182	12.1
$>= 0.04$	437	29.1	517	34.5	126	8.4
$>= 0.05$	266	17.7	337	22.5	96	6.4
$>= 0.06$	170	11.3	249	16.6	75	5.0
$>= 0.07$	117	7.8	206	13.7	54	3.6
$>= 0.08$	92	6.1	172	11.5	34	2.3
$>= 0.09$	70	4.7	134	8.9	17	1.1
$>= 0.10$	44	2.9	88	5.9	6	0.4
$>= 0.12$	11	0.7	23	1.5	1	0.1

We assume that results with differences lower than 0.01 in terms of F-measure could be considered as equal results, because they imply a very small difference between the corresponding representations. Therefore, following this assumption and looking at the Table 5.3 from left to right, in most of the cases EFCC works as well as (20.7%) or better (68.9%) than FCC. Between AFCC and FCC, the first one works as well as the second in a 15.5% of cases, and better in a 76.1% of cases. In the last block, AFCC works as well as EFCC in a 58% of cases, and AFCC outperforms EFCC in a 28.5% of cases.

Focusing our attention on the comparison of AFCC and EFCC, the difference is favorable to AFCC and greater than 0.02 in a 18.8% of total cases, and greater than 0.03 in a 12.1% of cases. On the other side, EFCC improves AFCC in more than 0.02 only in a 5.5% of the cases and in more than 0.03 in a 2.7%.

Summarizing, adjusting the membership functions to a dataset leads to results as good or better than FCC in a 91.6% of cases, and as good or better than EFCC in a 86.5% of the cases. EFCC and AFCC outperform FCC in most of the cases, and between them, AFCC allows to improve the results of EFCC in a 28.5% of cases.

5.5 Conclusion

In this chapter we aimed to study the possibility of automatically fitting a document representation, based on fuzzy logic, to different datasets. Our main concern was not to use any other information than that included in the datasets themselves. Keeping this in mind, in this chapter we have analyzed whether or not clustering results obtained by the fuzzy approaches analyzed in Chapter 4 can be improved using information from the datasets to perform the adjustment.

Parts of the research in this chapter have been published in Pérez García-Plaza et al. (2012b).

First, our analysis of the datasets showed clear differences among them. We compared them on the basis of their term frequency distributions. We showed the frequency distributions for terms within each criterion used in FCC and EFCC. Banksearch and SODP distributions showed a clear tendency towards exponential distributions. However, in WebKB we found a slightly different distribution for `frequency` criterion and a clearly different distribution for `emphasis` criterion.

Based on this analysis, we proposed a way of automatically establishing the basic membership function parameters, taking into account term frequency distributions and the original heuristics. This way, our proposal, called AFCC, is able to automatically adjust its membership functions aiming to better capture the information corresponding to each criterion. The proposed method does not adjust `position` criterion, because word positions on a page do not depend on anything else than the number of words in the document.

Then, we compared the resulting web page representation, called AFCC, with the previous ones in which it is based (FCC and EFCC), and with TF-IDF as standard term weighting function. Our evaluation showed that adjusting the representation to concrete collections can help improve clustering results.

Besides, we also saw the bad performance of all the representations with SODP collection. This dataset is characterized by a strong imbalance of the number of documents within each category. This could introduce a bias towards the biggest categories, causing that the clustering algorithm favors the division of the documents belonging to those categories. It is also worth noting the good performance of TD-IDF compared to our proposals in the case of SODP. We believe that the use of IDF could be a key factor to explain this good behavior, because it could alleviate the effect of the biggest categories by giving more representativeness to terms coming from smaller categories. For this reason, we believe that TF-IDF can be useful to deal with collections where most of the documents belong to a small number of categories, while most of the categories contain a much smaller number of documents. In these cases, TF-IDF could allow to improve the overall clustering results by improving the clustering related to the small categories.

We found that representations not tuned to fit concrete collections could perform reasonably well in most cases, but with our automated proposal it is possible to maintain their results in common cases, i.e., those following Zipf's law, at the same time that the representation is able to deal with not so common cases. Moreover, identifying those cases is not a computationally expensive task. Thus, we conclude that tuning the system to fit the dataset is a feasible way of improving web page representation for clustering tasks.

6 Test Scenario: Hierarchical Clustering Applied to Learn a Taxonomy from a Set of Web Pages in Two Languages

After analyzing different ways of combining criteria we presented two proposals to improve web page representation for clustering tasks (EFCC and AFCC). Nevertheless, these representations have been only tested in a plain clustering environment, with documents written in English. To validate our previous conclusions in a different environment, in this chapter we propose to apply that representations in a totally different framework. We introduce here the problem of taxonomy learning and a possible solution by means of web page hierarchical clustering. We also perform the experiments for two different languages: English and Spanish.

The chapter is organized as follows. First, in Section 6.1 on the following page we present the motivation of our research in this chapter. Next, in Section 6.2 on the next page we review previous work on the field of taxonomy learning to situate the scenario of our research. Then, in Section 6.3 on page 170 we describe the clustering method utilized in the experiments of this chapter. The evaluation methodology for taxonomy learning that we apply in this thesis is detailed in Section 6.4 on page 171, followed by a description of the experimental settings in Section 6.5 on page 174. Then, we show our experimental results in Section 6.6 on page 176 and finally we present our findings to conclude the chapter in Section 6.7 on page 180:

6.1 Introduction

In previous chapters we have proposed different ways of representing web pages for clustering tasks by means of fuzzy combinations of criteria. However, the whole set of experiments was performed in an environment of plain clustering. Besides, these experiments were performed for web pages written in English. In this chapter we extend this experimental framework to test the appropriateness of the proposed representations for hierarchical clustering tasks. We use a clustering process based on the SOM algorithm, with the aim of validating our results in a totally different environment. Finally, this experimentation is performed with a comparable corpus (WAD dataset, see Section 3.5), composed of web pages written in English and Spanish. This corpus contains Wikipedia documents that describe concepts. Each concept is described in a separate web page, and then the hierarchical clustering over that documents aims to create a taxonomy.

In this work we consider a taxonomy as a simplification of an ontology. Ontologies are a useful tool to represent relations between concepts. In fact, they are usually defined as formal representations of knowledge by means of a set of concepts and their relations. Of course, they are focused on concrete domains that are, by extension, described by the ontologies. We refer to a taxonomy as a simplified ontology, because of the relations between concepts are limited to parent-child relations only. In this way, we can see a taxonomy as a set of concepts hierarchically arranged.

Taxonomies and ontologies have a broad field of application that includes several natural language processing tasks like recommender systems (Yang and Hsu, 2010), IR (Pruski et al., 2011), text clustering and classification (Bloehdorn et al., 2006; Sridevi and Nagaveni, 2011), among others.

Due to the cost of the manual creation of ontologies, more automated methods based on machine learning have emerged. Along these lines, we presented an unsupervised method for creating a taxonomy of concepts from a set of web pages (Paukkeri et al., 2010, 2012) that is used in this thesis to complete the validation and evaluation of our representation proposals. The experimental framework of this method will be explained later in this chapter, but the main idea is organizing a set of web pages in a hierarchical structure like shown in the Figure 6.1.

6.2 Works on Taxonomy Learning

This chapter presents a different scenario to test our web page representation proposals. For this reason, we briefly describe here some previous works related with taxonomy learning, the field of application of our proposals in this chapter.

First of all, as this chapter deals with taxonomy learning from a set of web

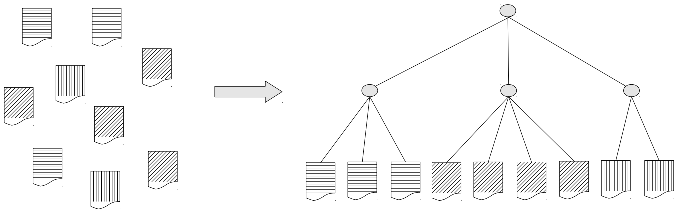


Figure 6.1: Web pages are hierarchically clustered to build a taxonomy; from Paukkeri et al. (2012).

pages, and a taxonomy can be considered a simplified ontology, this application of our representation proposals could be seen as a first step towards a most general problem of ontology learning. On the one hand, inducing a taxonomy of concepts, that is, a concept hierarchy, implies that each deeper hierarchy level has associated an increase in the specificity of the concepts it contains. Then, there is only one type of relation between concepts, that can be seen as hypernyms and hyponyms through the hierarchy levels. On the other hand, when dealing with non-taxonomic structures, the possible types of relations increase, appearing others like meronymy, synonymy or antonymy.

Without the intention of writing an exhaustive review on the state-of-the-art about taxonomy learning, we present here a brief review on the four main paradigms for inducing taxonomies:

- The first paradigm comprises methods that apply lexico-syntactic patterns to detect hyponymy relations. It is based on the possibility of finding explicit knowledge in some kind of texts. For instance, handbooks, textbooks, dictionaries, or even other popular resources as Wikipedia or social tagging systems can contain definitions such as "a red-tailed hawk is a bird" or "mammals such as moose, bat or mouse". One of the most relevant approaches was introduced by Hearst (1992). Her method identify the patterns that indicate hyponymy relations. For example, the following lexico-syntactic pattern corresponds to the hyponymy relation:

  If (NP_0 such as NP_1 , $NP_2 \ldots$, (and | or) NP_n)
  For all NP_i , $1 \leq$ i $\leq$ n, hyponym(NP_i , NP_0)

  where NP means Noun Phrase. Nevertheless, this approach suffers some drawbacks. First, it is not easy to find sufficient lexico-syntactic patterns. Second, it is also difficult to find examples enough including the terms of interest (Brewster and Wilks, 2004). Thus, this kind of approaches achieve reasonable precision values, but in exchange, their recall is very low. Brewster showed the extremely low probability of finding sufficient examples of any given lexico-syntactic pattern to be able to get reliable results. Another approach exploits the internal structure of noun phrases in order to derive taxonomic relations between classes. These classes are represented by the head of the noun phrase and its subclasses, that can be derived from a combination of the head and its modifiers (Buitelaar, 2000). A similar approach was presented in Pennacchiotti and Pantel (2006), where a bootstrapping method was applied to learn lexical patterns from raw text, given a small set of seed instances for a particular relation. Again, their results showed good precision values, but low recall. Natural language tools have also been used to extract semantic relations from text. In particular, Specia and Motta (2006) uses lemmatizer, syntactic parser, part-of-speech tagger, pattern-based classification and word sense disambiguation models together with resources such as domain ontology and lexical databases. The work of Ciaramita et al. (2005) presents an unsupervised model for learning arbitrary relations between concepts. This approach utilizes, in addition to syntactic patterns, a corpus of manually tagged named entities that correspond to ontology concepts.

- The second paradigm is based on the distributional hypothesis stated by Harris (Harris, 1954): words that occur in the same contexts tend to have similar meanings. Most of the approaches following this paradigm exploit hierarchical clustering algorithms to automatically derive term hierarchies from text, e.g., Grefenstette (1994), Faure and Nédellec (1998) and Cimiano et al. (2005). Hierarchical and non-hierarchical clustering, similarity measures and different linking schemes, among other statistical methods for extraction of taxonomic relations, were explored in Maedche et al. (2002). Besides, Staab (2005) introduced a guided agglomerative hierarchical clustering algorithm to create concept hierarchies from text collections by employing a hypernym oracle. This oracle uses hypernyms from WordNet and the above mentioned lexico-syntactic patterns proposed by Hearst, matched in a corpus or the Internet. Another approach based on clustering was proposed by Snow et al. (2006). The authors incorporates evidence from multiple classifiers to optimize the entire structure of the taxonomy. Our work has to do with this paradigm, as it is common to find hierarchical clustering approaches relying on the VSM. Our proposals are based on the VSM, so they can be directly applied. Besides, our work presented in Paukkeri et al. (2012) proposes an unsupervised method to hierarchically cluster a set of web pages to build a taxonomy. We rely on this method to perform our experiments in this chapter, since this method only needs a set of web pages represented in the VSM as input.
- The third paradigm comes from the IR field. It basically relies on the use of documents retrieved in response to queries. With this approach, Sander-

son and Croft (1999) derived concept hierarchies from text documents by using a type of co-occurrence known as subsumption. Given a query and a document collection, they use Local Context Analysis (Xu and Croft, 1996) to expand the query with additional terms and retrieve the top 500 ranked documents to apply two means of term selection over them. One of them is based on the co-occurrences of words and phrases in the best matching passages of the top ranked documents and the other on a comparison of a term's frequency of occurrence in the retrieved documents with its occurrence in the collection. Finally, every selected term was compared to every other term to test for subsumption relationships. In Sanchez and Moreno (2005), another automatic and unsupervised methodology to obtain taxonomies of terms from the Web was presented. It is based on an intensive use of web search engines to retrieve domain suitable resources to extract knowledge and to obtain web scale statistics from which infer knowledge relevancy. They employ a domain defined by an initial keyword and linguistic patterns involving that keyword to find candidate concepts for the domain. The result is a hierarchical organization of the available knowledge and web resources for the given domain. The main drawback comes from the use of a keyword, because as the authors stated: "if several ways of expressing the same concept exist (e.g. synonyms, lexicalizations or even different morphological forms), a considerable amount of potentially interesting web resources will be omitted". In Riloff and Kozareva (2009) a supervised bootstrapping algorithm is presented for reading web texts and learning taxonomy terms. The algorithms starts with two seed words and progressively learns hyponym and hypernym terms using Google search engine to obtain documents to apply the patterns.

- The fourth paradigm is based on social media. Concretely, social tagging systems have a flat and uncontrolled underlying resource organization paradigm. This fact has encouraged the development of new approaches for taxonomy learning based on folksonomies, like the work of Benz and Hotho (2007). Moreover, taxonomies have been learned also from Wikipedia by using its categories as concepts in a semantic network, and finding relations between concepts on the basis of the connectivity of the network. In a recent work, Wong (2009) introduces the Tree-Traversing Ants (TTA) clustering technique for learning taxonomic relations. TTA is based on dynamic tree structures and it adopts a two-pass approach for term clustering. During the first pass, nodes are recursively broken into sub-nodes using Normalized Google Distance. The second pass is a refinement phase where terms are relocated according to the *n-degree of Wikipedia* measure, that uses information from Wikipedia categories.

Our decision on using the method presented in Paukkeri et al. (2012) was made because of our interest in a method that relies only in the dataset documents themselves, as the web page representations proposed in this thesis are also oriented to this kind of unsupervised environment. We chose this test scenario as an interesting application of hierarchical clustering to test our representation proposals. We do not aim to improve taxonomy learning techniques, but check whether or not our representation can be successfully applied to that problem.

6.3 Clustering Method

In this case, we want to cluster concepts. These concepts correspond to the documents in the dataset, that is, each document is considered a concept definition. In this way, it is assumed that each concept has a hypernym, that corresponds with a parent node in the hierarchical clustering tree. Thus, the algorithm is feed by a set of vectors that represents the documents in the VSM. In this chapter, as each document corresponds to a concept definition, by extension we will refer to document vectors also as concept vectors.

Intuitively, the approach used to create the taxonomy is top-down, starting from the zero-level, that corresponds with the root node. This node will contain the whole collection. In the next step, the collection will be divided in separated clusters. Thus, the first level will consist of different clusters, each one containing a different part of the collection. Every cluster on the first level will be split again, resulting in a new level of the hierarchy. The process will continue until finding the desired hierarchy.

In particular, the clustering process applied on each level is based on the SOM algorithm. A more detailed description of this algorithm can be found in Section 2.3 on page 66. The SOM algorithm does not produce proper clusters, but a set of neurons arranged in such a way that they represent a topological ordering of the input data. So, proximity in the map will imply similarity between the input vectors.

To explain the process, Figure 6.2 show an example of how the hierarchy is generated by splitting the SOMs in clusters. We start the process in the zero level, with the whole set of input vectors, that corresponds to collection documents. The SOM is trained with these vectors and the vectors are then mapped onto the SOM. The next step is splitting the map in proper clusters, because as we said above, the SOM does not generate proper clusters. However, several adjacent neurons—and the document vectors mapped onto them—could be considered a cluster, as proximity in the map implies similarity. To define the clusters and their limits on the map, i.e., to split the SOM in separate clusters, an agglomerative hierarchical binary algorithm is applied. These clusters will constitute the first level of the taxonomy, that is, the level just below the root. It is worth noting that each first level cluster will contain part of the initial set of document vectors.

Then, on the first level we have a set of clusters coming from the zero level (resulting of splitting the SOM that contained the whole set of document vectors). For each cluster on the first level, a new SOM is trained with the vectors belonging to that concrete cluster, that are mapped onto the SOM after the training stage. Then the agglomerative algorithm mentioned above is applied over each SOM to split them into clusters again. The same clustering procedure is repeated for each level in the desired taxonomy, or until finding a cluster containing only one concept.

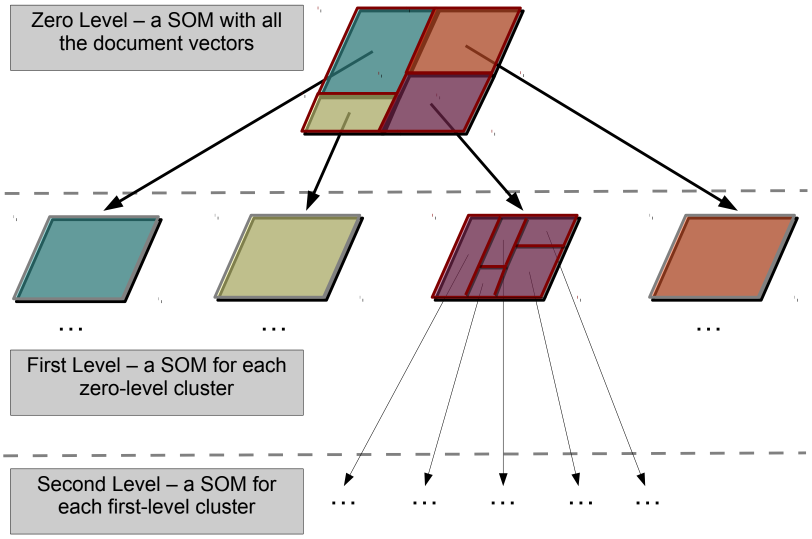

Zero Level – a SOM with all the document vectors
...
...
...
First Level – a SOM for each zero-level cluster
Second Level – a SOM for each first-level cluster
... ...

Figure 6.2: Example of the SOM based hierarchy.

6.4 Evaluation Methodology

Ontologies have been evaluated in previous works using four different approaches (Brank et al., 2005):

- By comparison to a gold standard, that could be directly a reference ontology.
- Evaluating the ontology as a part of a concrete application.
- By comparison to source data about the domain.
- Evaluation made by humans.

In this thesis we employ the first method, that is, we compare the taxonomies created by means of the hierarchical clustering method against a manually constructed reference ontology.

To do the comparison, we rely in the method proposed by Dellschaft and Staab (2006). The authors propose a set of taxonomic measures oriented to perform a gold standard based evaluation of ontology learning. Concretely, three measures were introduced: taxonomic precision, taxonomic recall and taxonomic F-measure. Other works like Brewster et al. (2003) and Brewster and Wilks (2004) applied these metrics to evaluate their results by using a gold standard. However, the experimentation of those works was oriented to find single concepts and their semantic relations, that is a different approach than ours, where the goal is to create a taxonomy.

For calculating the global taxonomic precision (TP_{csc}) of two ontologies, they rely on the common semantic cotopy (csc). The semantic cotopy of a concept is defined as the set of all its super- and subconcepts. The common semantic cotopy excludes all concepts which are not also available in the set of concepts of the other ontology:

$$csc(c, \mathcal{O}_1, \mathcal{O}_2) = \{c_i | c_i \in \mathcal{C}_1 \cap \mathcal{C}_2 \wedge (c_i <_1 c \vee c <_1 c_i)\}. \tag{6.1}$$

where $\mathcal{O}_i$ are the ontologies to be compared, c is a concept, $\mathcal{C}_i$ are the sets of concept identifiers and $c <_1 c_i$ means that c is a subconcept of c_i. Then, the local taxonomic precision (tp_{csc}) compares common semantic cotopies of concepts of $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$:

$$tp_{csc}(c_1, c_2, \mathcal{O}_1, \mathcal{O}_2) = \frac{|csc(c_1, \mathcal{O}_1, \mathcal{O}_2) \cap csc(c_2, \mathcal{O}_1, \mathcal{O}_2)|}{|csc(c_1, \mathcal{O}_1, \mathcal{O}_2)|} \tag{6.2}$$

The global taxonomic precision TP_{csc} is based on the number of semantic relations that can be found in both, the learned ontology and the reference ontology:

$$TP_{csc}(\mathcal{O}_C, \mathcal{O}_R) = \frac{1}{|\mathcal{C}_C \cap \mathcal{C}_R|} \sum_{c \in \mathcal{C}_C \cap \mathcal{C}_R} tp_{csc}(c, c, \mathcal{O}_C, \mathcal{O}_R). \tag{6.3}$$

In the same way as we calculated the recall in plain clustering, the global taxonomic recall (TR_{csc}) is computed using the global taxonomic precision:

$$TR_{csc}(\mathcal{O}_C, \mathcal{O}_R) = TP_{csc}(\mathcal{O}_R, \mathcal{O}_C). \tag{6.4}$$

Finally, the taxonomic F-measure follows the same approach than traditional F-measure does. It is calculated as the harmonic mean of the global taxonomic precision and recall:

$$TF_{csc}(\mathcal{O}_C, \mathcal{O}_R) = \frac{2 \cdot TP_{csc}(\mathcal{O}_C, \mathcal{O}_R) \cdot TR_{csc}(\mathcal{O}_C, \mathcal{O}_R)}{TP_{csc}(\mathcal{O}_C, \mathcal{O}_R) + TR_{csc}(\mathcal{O}_C, \mathcal{O}_R)}. \tag{6.5}$$

This evaluation measures are oriented to the comparison of two ontologies. As the output of the clustering algorithm is a hierarchy of concepts, we need to transform it to an ontology. To do that, the concepts of the learned taxonomy are labeled using concept labels that can be found as meta information in each document. It is important to highlight that this meta information is only explicitly used for evaluation purposes. After this labeling process, it is possible to compare the obtained ontology with the reference one.

The procedure we follow for labeling the learned taxonomy begins using titles of documents as concept labels in the lowest level of the hierarchy. In that case, labels are used to know the categories represented by each cluster in the system answer. Then the parent concepts of each concept are collected from the reference ontology. These parent concepts are used to label the clusters based on the majority of parent concepts for each and every cluster. If more than one parent concept are candidates for labeling a cluster, i.e., they appear the same number of times, the label is randomly selected among them. This procedure is followed for each cluster in the taxonomy.

As the evaluation measure only allows that each label appears once in the whole hierarchy, clusters that having the same label are siblings (they share the same parent node in the hierarchy) are merged in a single cluster for labeling purposes. If the same situation occurs with two clusters that sharing the same label are not siblings, i.e., they have different parent nodes, the smaller one remains as *unclassified*, penalizing the solution, since unlabeled clusters will be counted as errors in the learned taxonomy. An example of this procedure is shown in Figure 6.3.

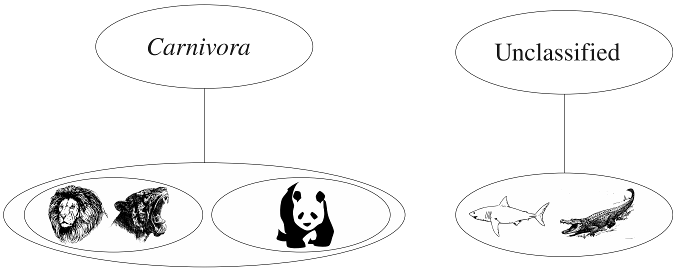


Figure 6.3: Example of labeling. On the left two clusters sharing parent are merged, while on the right a third cluster that does not share the same parent remains unclassified.

It is worth mentioning that to calculate precision, recall or F-measure in plain clustering, we would search for the best cluster representing each category, so all categories would be represented in the evaluation of our solution. The problem with the present approach comes from the use of taxonomic measures. These measures take into account the structure of a solution and the concepts included in that structure. In this way, the results would be better if we had all the concepts in the solution, even without introducing any modification in the hierarchy, than in the case of missing concepts. With the present approach we search for the best cluster for each category, but we do not need to find clusters for all the categories in the reference solution, which will penalize the solution.

About modifying the structure in evaluation, this step do not lead to obtain better results. First, the evaluation step is the same for all the representations. Last, but not less important, when merging clusters the probability of not to find one cluster per each category in the reference ontology will increase, so merging will increase the probability of finding unlabeled clusters, which implies an important penalization in evaluation, as stated above.

6.5 Experimental Settings

The experiments described in this chapter were carried out for the WAD dataset, analyzed in Section 3.5 on page 94. It consists of 166 documents for each language, English and Spanish. Thus, there are 166 concepts to create a taxonomy in each language. Both taxonomies are equivalent, as the documents correspond to the same concepts in English and Spanish. This allows to evaluate the learned taxonomy by using the same reference ontology. The taxonomy we want to build from the concepts corresponds to the scientific classification of the animals that are the concepts defined in the documents. As explained in Section 3.5 on page 94, the reference hierarchy was slightly simplified to three levels of the scientific classification.

To be consistent with the previous experiments performed in this thesis, the same preprocessing steps, described in Section 4.3 on page 114, were performed. After that, the documents were represented by means of AFCC, EFCC, FCC and TF-IDF term weighting functions, and reduced using MFT dimension reduction technique in the case of the fuzzy logic based representations and by means of DF in the case of TF-IDF. The latter decision was made on the basis of the good results obtained by this combination in WebKB dataset (see Table 4.3 on page 119). WebKB consists of documents coming from a restricted domain and most of its web pages were collected from only four Universities. This could introduce web domain related terminology as we saw before. Also, WebKB categories are composed of heterogeneous pages within each category, as we showed in Section 3.3.1 on page 84. This facts should have a bigger influence on WAD dataset, as it contains documents coming from only one web domain, so the terminology associated to that domain will be present in the whole collection. Moreover, all the documents deal with animals, that is, the degree of homogeneity among collection documents will be high. Thus, DF method fits very well the problem of filtering too common or too rare terms, because there is a greater probability of finding terms non-representative enough for distinguishing single categories globally distributed through the whole collection. The vocabulary sizes selected for applying the reduction techniques were 100, 500, $1,000$ and $2,000$ features. The case of $5,000$ features was not used in these experiments because of the size of the collections. To give a concrete example, the total number of terms in Banksearch vocabulary is $210,785$, while in WAD dataset these numbers are $27,004$ terms for English and $14,404$ terms for Spanish. Having this difference in mind, and taking into account the difference in the dataset size, that is approximately 27 times smaller than the smaller one used previously in this thesis (that is, WebKB), we believe that $5,000$ features are too much to represent WAD documents without introducing noise in the representation.

Different from the previous experiments, the hierarchical clustering algorithm does not receive the exact number of clusters on each hierarchy level. Instead, a maximum number of clusters was fixed on each level. On the first one, a maximum of 4 clusters was set, but the algorithm could decide to split the documents in a smaller number of partitions. This decision was made taking into account that 4 is the number of categories in the first level of the reference taxonomy (see Figure 3.16 on page 96 and Table 3.4 on page 97). For the second hierarchy level, a maximum of 20 clusters was configured, that allows 5 clusters for each first level cluster in the case of finding the 4 first level clusters. Thus, if the algorithm is not able to find the correct number of clusters to fit the reference taxonomy, other options are allowed. It is worth noting that the degree of each node in the reference taxonomy is different, that is, not all the nodes of the reference hierarchy have the same number of child nodes.

For each clustering step, a SOM is employed to organize the corresponding documents. As the level of the hierarchy increases, the number of document clusters we need to find increases too (see the Figure 3.16 on page 96), so the size of the SOMs is configured accordingly. On the first level, we trained a SOM of $5x5$ cells. On the second level we chose a slightly larger size of $7x7$ cells, as we want to find a larger number of clusters. To establish these sizes some preliminary tests were carried out (Paukkeri et al., 2012). These tests revealed that larger maps—up to $20x20$ cells—performed worst than smaller sizes. However, the exact number of cells did not reflect a strong effect on map results.

With respect to the rest of SOM algorithm parameters, in this work we used random initialization, normalization of the variance and batch training, all of them configured via SOMToolbox[1] implementation. As a consequence of the random implementation and different from Cluto rbr algorithm utilized in the previous experiments in this thesis, the SOM algorithm is non-deterministic. Knowing that, we repeat the clustering process on each level 10 times. So, for the first level it corresponds to 10 different clustering solutions, and then, for the second level,

[1] `http://www.cis.hut.fi/somtoolbox/`

another 10 clustering solutions are computed for each cluster coming from the first level. This leads to a total number of 100 taxonomies generated for each document representation. The total number of experiments was $3,200$ (which implies to learn and evaluate $3,200$ different taxonomies), because we tested 4 different representations, each one with 4 different vocabulary sizes, and all of this twice, for English and Spanish web pages.

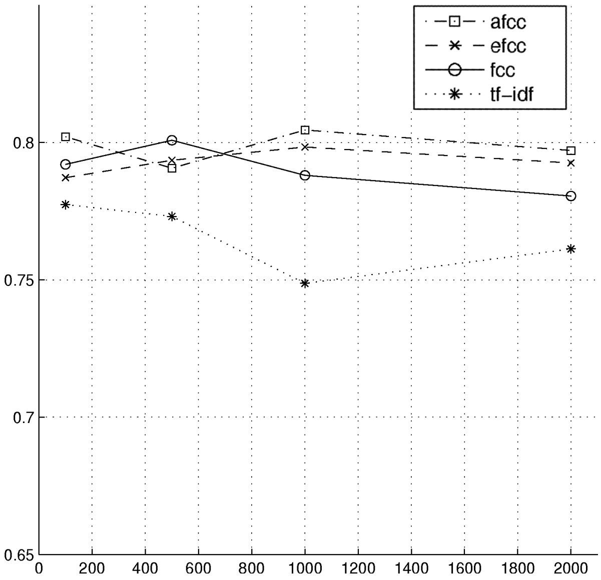
Regarding the agglomerative hierarchical binary clustering performed to split each SOM in proper clusters, the correlation distance using the complete linking scheme was employed. Again, some preliminary experiments were performed using other measures as cosine or Spearman, and linking schemes as single or average. The three measures obtained very similar results, in particular for cosine and correlation, while Spearman distance performed slightly worse. Among the linking schemes, it was detected that by using single and average linking, the standard Matlab implementation of the algorithm used in our experiments was not able to find a clustering solution in some cases. Because of this, the complete linking scheme was used instead.

6.6 Results

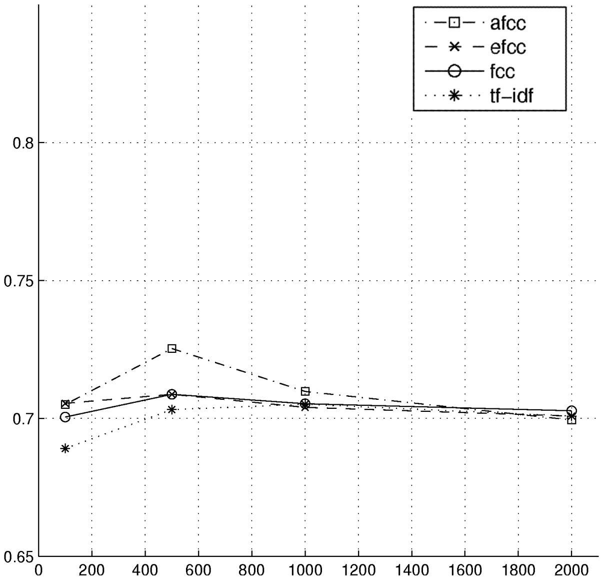
After the evaluation of the learned taxonomies against the reference ontology, the taxonomic F-measure results obtained for both languages, English and Spanish, are shown in Figure 6.4. These results are the average value of whole set of experiments performed for each representation and vector size (see Section 6.5 for more details).

In all cases the fuzzy logic based approaches got similar or better results than TF-IDF. In English, these approaches clearly outperforms TF-IDF, while in Spanish all the representations perform in a more similar manner, with the exception of the case of AFCC that performs slightly better, in particular when using 500 features. In the same manner, TF-IDF performs slightly worse for Spanish than the rest when reduced to the smaller vector sizes.

One obvious issue emerges when comparing results between languages: Spanish results are considerably worse than English ones. Documents written in Spanish are shorter than the English ones, which could explain, in part, that difference. Besides, TF-IDF gets more similar results to the other methods than in English, which can be seen as a lack of effectiveness on the side of the fuzzy logic based representations when dealing with Spanish texts. Due to the combinations of criteria in which AFCC, EFCC and FCC are based, the stemming stage is very important, because after this stage, a given stem should have the same form in the title, the body and emphasis, which may not be the case for some Spanish words.



(a) English WAD results



(b) Spanish WAD results

Figure 6.4: F-measure results for WAD experiments in English and Spanish.

Using a stemmer[2] is not as appropriate as in English, and it could directly affect the results. These two factors constitute a possible explanation for the difference

[2] We used the Spanish stemming algorithm from Snowball web page: `http://snowball.tartarus.org/algorithms/spanish/stemmer.html`

we found comparing the taxonomic F-measure values between languages.

Among representations, AFCC performs slightly better than the rest of the fuzzy logic alternatives and, at the same time, it achieves a difference a bit larger with TF-IDF. On the other hand, EFCC and FCC obtain very close results.

In addition to Figure 6.4 and in a similar manner that in previous chapters, Table 6.1 shows a summary of our experiments. Following a similar approximation than in the rest of this dissertation, we also performed a statistical two-sample t-test for each representation and vocabulary size. The table contains the taxonomic F-measure results that are also shown in Figure 6.4 for both languages. Moreover, for each representation, the average and the standard deviation of the taxonomic F-measure are shown. Next, the differences between each pair of representations for each vocabulary size are detailed. Finally, the table also contains the *p*-values corresponding to the t-test experiments.

Looking at the averages in the table, the difference among AFCC and EFCC is greater than the difference between EFCC and FCC. In this case, adjusting the membership functions has a bigger effect on the results than the modification of the rules.

Within the *p*-value blocks in the table, bold font is used to highlight the cases in which the results of the corresponding pair of representations come from different populations with likelihood greater than 99%.

In English, the fuzzy logic based representations obtained significantly better results than TF-IDF in all cases. Among the fuzzy logic based approaches, there are no significant differences between AFCC and EFCC in most cases, except in the case with smallest number of features. Between EFCC and FCC the difference are significant only with $1,000$ and $2,000$ features. However, AFCC got significantly better results than FCC with 100 features only, while FCC improves AFCC with 500 features, where FCC achieves its best result. In general, FCC got a good maximum value with 500 features, but it is less stable than AFCC, whose results are significantly better than FCC ones in the rest of the cases. These results are also more stable regardless the number of features. EFCC results are slightly worse than AFCC ones, though they show also great stability among the different vocabulary sizes.

In Spanish, above 500 features all the representations tend to the same values, without statistical differences among them. We mentioned above that Spanish documents are shorter than English ones. Thus, the initial vocabulary of Spanish documents is smaller and, by reducing to $1,000$ features or more, we are probably exceeding the limit above which the reduction process starts introducing noise. Nevertheless, with the smallest vector sizes— 100 and 500—the fuzzy logic based representations outperform TF-IDF. Among them, AFCC is the best one, achieving in both cases statistically significant differences with TF-IDF.

Finally, the overall taxonomic F-measure is relatively high, and the fuzzy logic

Table 6.1: Results for AFCC/EFCC/FCC/TF-IDF t-test experiments

English							
		100	**500**	**1000**	**2000**	**Avg.**	**S.D.**
F-measure	AFCC MFT	**0.802**	0.791	**0.805**	**0.797**	**0.799**	0.006
	EFCC MFT	0.787	0.794	0.798	0.793	0.793	**0.005**
	FCC MFT	0.792	**0.801**	0.788	0.781	0.790	0.008
	TF-IDF MFT	0.777	0.773	0.749	0.761	0.765	0.013
Difference	AFCC - EFCC	0.015	-0.003	0.006	0.005		
	AFCC - FCC	0.010	-0.010	0.017	0.017		
	AFCC – TF-IDF	0.025	0.018	0.056	0.036		
	EFCC - FCC	-0.005	-0.007	0.010	0.012		
	EFCC - TF-IDF	0.010	0.020	0.050	0.031		
	FCC – TF-IDF	0.015	0.028	0.039	0.019		
p-value	AFCC - EFCC	**0.000**	0.186	0.024	0.099		
	AFCC - FCC	**0.000**	**0.000**	**0.000**	**0.000**		
	AFCC – TF-IDF	**0.000**	**0.000**	**0.000**	**0.000**		
	EFCC - FCC	0.053	0.015	**0.001**	**0.000**		
	EFCC - TF-IDF	**0.003**	**0.000**	**0.000**	**0.000**		
	FCC - TF-IDF	**0.000**	**0.000**	**0.000**	**0.000**		
Spanish							
		100	**500**	**1000**	**2000**	**Avg.**	**S.D.**
F-measure	AFCC MFT	**0.705**	**0.725**	**0.710**	0.699	**0.710**	0.011
	EFCC MFT	**0.705**	0.709	0.704	0.701	0.705	**0.003**
	FCC MFT	0.700	0.709	0.705	**0.703**	0.704	0.004
	TF-IDF MFT	0.689	0.703	0.705	0.701	0.700	0.007
Difference	ABS - EFCC	0.000	0.017	0.006	-0.001		
	ABS - FCC	0.005	0.017	0.004	-0.003		
	ABS – TF-IDF	0.016	0.022	0.005	-0.001		
	EFCC - FCC	0.005	0.000	-0.001	-0.002		
	EFCC – TF-IDF	0.016	0.006	-0.001	0.000		
	FCC – TF-IDF	0.011	0.006	0.000	0.002		
p-value	ABS - EFCC	0.838	**0.000**	0.044	0.457		
	ABS - FCC	0.034	**0.000**	0.096	0.125		
	ABS – TF-IDF	**0.000**	**0.000**	0.126	0.492		
	EFCC - FCC	**0.009**	0.998	0.599	0.344		
	EFCC – TF-IDF	**0.000**	0.055	0.695	0.998		
	FCC – TF-IDF	**0.000**	0.061	0.944	0.377		

based representations reveal their usefulness in hierarchical clustering scenarios, where they obtained good results compared to a standard representation as TF-IDF. More than that, in Paukkeri et al. (2012) another state-of-the-art statistical technique for keyphrase extraction was tested in the same clustering environment, being its results worse than the ones showed here for the fuzzy logic approaches.

6.7 Conclusion

In this chapter we have described a test scenario to validate our proposals in a hierarchical clustering environment. Besides, we extend the comparison beyond English language and decided to test the fuzzy based representations also in Spanish.

Parts of the research in this chapter have been published in Paukkeri et al. (2010) and Paukkeri et al. (2012).

We apply hierarchical clustering to a taxonomy learning problem from a set of text documents that contain concept definitions. We considered that this kind of clustering environment is considerably different from the others presented in previous chapters and therefore it constitutes an appropriate field for validating our proposals.

First we described the problem we faced, the clustering method we used, the evaluation methodology we followed and the experiments we performed.

After analyzing the results, our proposals showed a good behavior when applied to this taxonomy learning problem. We used a comparable bilingual corpus, one side composed of documents written in English, that is a Germanic language, and the other composed of the corresponding web pages written in Spanish, that belongs to the Romance language family. Documents in English and Spanish describe the same concepts, but are not exact translations from each other. Even in Spanish and with some issues related with the preprocessing stage and the length of the documents, the results of our representation proposals, AFCC and EFCC are still comparable or even improve TF-IDF. Among languages, AFCC shows the best behavior, achieving particularly good results in English.

Taking everything into account, we can say that AFCC is suitable for representing web pages in order to deal with hierarchical clustering problems in two different languages: English and Spanish. Then, we can state that the experimentation carried out in this chapter serves to validate the suitability of our representation proposals in a different test scenario.

7

Conclusions and Future Research

We conclude the thesis in this chapter. We expose our main conclusions in Section 7.1. Next we summarize the main contributions of this work in Section 7.2 on page 189. Finally, in Section 7.3 on page 190 we present an outlook on future directions of the research work in this thesis.

7.1 Conclusions

This section is structured as follows. First, we summarize the different topics we have studied throughout this dissertation. Second, we present a detailed review of our conclusions organized by chapters.

7.1.1 Brief Summary of the Research Included in this Thesis

The main goal of this thesis is to perform a deep study with the aim of making the most of a fuzzy model to represent HTML documents for clustering tasks. Web pages are commonly written in HTML language, that offers explicit information (tags, in this case) about their visual representation, the typography of the text or its structure, among others.

Our proposals are directed towards finding a web page representation method allowing to easily express expert knowledge about how a human being has a quick look at a document to find out its main theme. By using fuzzy logic we separate the knowledge declaration from the calculation procedure. It also allows to specify the knowledge by means of a set of rules close to natural language.

All things considered, we analyzed three different aspects of web page representation for clustering: the feature selection sources to extract essential information for web page representation, the term weighting functions to estimate the weight of each feature, and the dimension reduction techniques to select the most representative features and to reduce the computational cost of the clustering, that otherwise could be unapproachable in terms on computational cost. For feature selection, we explored some new criteria to improve the representation with collection information or anchor texts. For term weighting we explored the fuzzy combination of criteria performed by FCC (Fresno, 2006) aiming to get the most of the fuzzy system and the heuristics in which it is based. We use TF-IDF as baseline, since it is a standard weighting method employed to represent documents. We presented an improved representation called EFCC, and another alternative called AddFCC, that worked worse than EFCC and was discarded. Both alternatives propose to exploit the fuzzy system in a different manner than FCC, taking advantage of its additive properties. For dimension reduction we presented MFT, a lightweight dimension reduction technique, based on the term weighting function, able to improve the results of more complex techniques as LSI when used together with EFCC in our test collections. Thus, we proposed the combination of EFCC and MFT as a general method to represent web pages for clustering, since it showed a good performance among three different collections compared to other similar approaches as FCC or AddFCC.

Besides, we wanted to study whether EFCC could be tuned to fit concrete characteristics of different collections. The aim of this adjustment is not only to improve clustering results in those collections, but also to adapt the representation to different datasets that could have different features. We showed that most of the term distributions we analyzed for our test collections follow Zipf's law. However, we found the case of WebKB, which shows different features, particularly for emphasis. This fact encourages us to study fuzzy system tuning from an unsupervised point of view. We decided to propose a new web page representation method called AFCC, where membership function basic parameters are adjusted on the basis of the term distributions of the collections. We showed that AFCC maintained or improve the good results of EFCC and FCC in common cases, i.e., those following Zipf's law, being also able to deal with not so common cases, where improved their results.

Finally we tested EFCC and AFCC in a hierarchical clustering environment. In particular, we presented the problem of learning a taxonomy from a comparable corpus of web pages written in English and Spanish. This scenario was used to validate our proposals and explore their possibilities in a language different from English.

7.1.2 Conclusions Detailed by Chapters

In order to clearly expose our conclusions, this section is structured by chapters.

Conclusions on the Review of Previous Web Page Representation Proposals

In Chapter 2 we reviewed different approaches to web page representation from the point of view of clustering tasks. We showed that different representation models mainly differ in the information sources they use, the weighting functions they apply over such information, and the dimension reduction techniques they employ.

Among the term weighting functions, TF-IDF or sometimes just TF, are usually applied. These functions are based in plain text only. In order to improve the results TF-IDF as document representation method, several works have proposed to employ different information extracted from the web page contents. Most of these works rely on criteria like document title, emphasized text segments, headers or information related with hyperlinks to enrich the representation.

We saw that a common approximation to include this information in the representation is weighting each term with a term weighting function like TF or TF-IDF on the basis of term occurrences within each criterion (title, headers, etc.). Thus, for each document, we obtain different weights for the same term in each criterion. To combine these weights, most of the works rely on linear combinations, where the importance of a term in a single criterion is calculated regardless the rest of the components. We consider that this kind of combinations is not the best option to combine criteria, as we can not express dependencies among them. Besides, the approaches based on linear combinations usually rely on coefficients to establish the influence of each criterion in the combination. These coefficients are manually or empirically selected. In fact, in some cases we have seen that these coefficients need to be empirically adjusted to each collection in order to achieve better results. This points out the fact that each different collection could need different adjustments for the combination. Other works fix their values beforehand, but most of them do not explain the reasons for that selection. Although these coefficients strongly affects the results, to the best of our knowledge, there are no proposals to automatically determine their values when category information is not available, neither advices to establish them when dealing with a particular collection.

In other cases, the combination of criteria is carried out by the algorithm, losing the independence between the representation and clustering processes. Then, to introduce some modification on either side, representation or algorithm, the whole system has to be changed. Besides, this approach does not allow to perform a direct comparison of different document representations. This way, this kind of approximations make difficult to analyze whether the benefits or drawbacks of an approach come from the representation or from the algorithm. We consider the independence between representation and algorithm very important, because we aim to propose a web page representation that can be applied in different clustering scenarios, and different problems may require different clustering algorithms.

In order to allow the definition of related conditions for establishing term importance—e.g., a term having high frequency in the document should appear in the title or emphasized to be considered important—, we showed our interest in fuzzy rule based systems to combine criteria. We believe that these systems are more suitable to express heuristic knowledge about the combination. They allow to focus our attention on defining the rules of the combination without specifying the calculation procedure. Fuzzy logic is based on the combination of a set of linguistic expressions based on words instead of numeric values and it was previously applied to document representation tasks. We believe that fuzzy combinations of criteria fits better the problem of establishing term importance, because there are dependencies among criteria that should be taken into account in order to deal with authors' writing style, automatism in web page creation (as titles automatically generated by HTML editors), criteria that not always contribute to the combination (like the case of `position`), etc. The expressiveness power of the rules and the non-linearity in the combination, together with the possibility of creating vectors within the VSM, encouraged us to explore the way on which the criteria can be combined by means of a fuzzy system.

At this point, the dimension reduction process can also affect the effectiveness of the representation. It should remove useless features by selecting those more representative to find relations among documents belonging to the same dataset categories. There are works comparing different approaches as DF, RP, ICA or LSI. Besides, LSI, RP and DF are widely used to reduce dimensionality when dealing with clustering problems in the literature. Nevertheless, these works do not analyze how each of them behave with different weighting functions.

On the other hand, in addition to the content of the documents, link structure has also been employed to represent web pages. In most cases, both information sources has been combined. These combinations usually employ standard weighting functions within the VSM for content side. In this sense, we could substitute this function by other alternative based on the VSM. Thus, by improving either the link structure side or the content side, the improvement should be reflected in the final combination. There are also works using anchor texts linearly combined with document contents. In this linear combinations, anchor texts are treated as other elements like page titles or terms that comes from document contents. In other cases, anchor texts terms are used in a similar manner as terms from the contents of the page, i.e., adding them to page contents. However, these combinations usually include other elements as page titles or link structure, and we did not find any study on the usefulness of adding anchor texts in particular to page contents for clustering tasks.

Recently, Wikipedia has also been used as external source of information to enrich document representation. These approaches rely on a linear combination, where coefficients are based on preliminary results, to integrate Wikipedia-based information with document contents. As a previous step for document representation, these approximations need to perform the parsing of a Wikipedia corpus to extract concept information. In this thesis we do not use Wikipedia information. However, it could be an interesting alternative for future works, since it showed encouraging results.

Regarding the datasets employed in evaluation, they differ from one work to another. There is no a standard set of web page collections for evaluating clustering tasks. In this sense, even when the same dataset is used, like is the case of WebKB, each work utilizes it with a different preprocessing. For example, it is common to use only some of its categories, or even just a part of these categories. However, the filtering process is not always well described.

Taking all the above mentioned issues into account, we perceive the lack of a standard methodology to compare web page representations. Each work establishes its own framework and, though some aspects are shared through different works, they do not follow a common process that allow to obtain results comparable with previous works. Because of this, in this thesis we try to make the representation process independent from the rest, centering our research and modifications mainly on this stage, at the same time we try to keep the rest of the framework as standard as possible, by means of employing techniques, algorithms, datasets and measures widely used in the literature.

Conclusions on the Selection and Analysis of Web Page Datasets

In this dissertation we employed four different web page collections to evaluate our proposals, that are described in detail in Chapter 3. We saw that Banksearch is easier to cluster than WebKB and SODP, because of its good balance of documents per category and its well differentiated themes. WebKB is more difficult, because most of its documents come from mainly four Universities and they are organized in unbalanced categories. The main difficulty of WebKB is the heterogeneity among documents within categories, that appears due to the design of its categories. SODP is the hardest one, with a larger number of unbalanced categories, and greater differences among the number of documents within each category. We extended this collection by retrieving anchor texts corresponding to inlinks to collection pages, in order to include those anchor texts in the combination. Finally, WAD was created for hierarchical clustering tasks and all its web pages come from Wikipedia. For each collection we also showed term distributions in the different criteria considered in this thesis. Basically, Chapter 3 contains an analysis of the characteristics of each dataset that are referred from successive chapters.

Conclusions on the Study of a Fuzzy System for Web Page Representation

In Chapter 4 we aim to make the most of a fuzzy rule based representation model applied to web page clustering tasks.

We began our research testing the effect of different dimension reduction techniques and comparing a previous fuzzy logic based representation (FCC) with TF-IDF, in order to establish a starting point to analyze such fuzzy system. Besides, we presented a dimension reduction technique, called MFT, that tries to select the most important terms for representing the documents in a collection based on the results of the weighting function applied over the terms. We showed that with a good weighting function it is feasible to employ lightweight dimension reduction techniques, as the proposed MFT, instead of using other more complex techniques like LSI, which implies an important reduction in the computational cost. We also found that DF reduction is particularly useful compared to MFT for TF-IDF when dealing with WebKB. We believe that there are three particular factors that favor this behavior: first, it is a collection composed of web pages coming from a small number of web domains, second, the heterogeneity among documents within each category, and third, the imbalance in the number of documents among categories. Regarding RP method, that is used in the literature as a lightweight alternative to LSI, our experimental results showed that it is not a good alternative. Its clustering results are much worse than those obtained by LSI in most of cases and other lightweight alternatives like DF or MFT also achieved better results than RP.

Our initial experiments with FCC and different dimension reduction techniques showed the bad performance of FCC in WebKB collection. An analysis of FCC was performed by isolating each criterion and comparing their results to the combination. The aspects that, in our opinion, hinder FCC—the overestimation of `position`, underestimating at the same time the rest of the criteria as detailed in Section 4.5.2 on page 127—were identified. Thus, we proposed two fuzzy logic based alternatives of combining criteria, AddFCC and EFCC. Our experiments showed that EFCC worked better than FCC by means of a different way of combining criteria, where term frequency is considered as discriminant as title and emphasis, and position is taken into account as the least important criterion. This approach made also possible to reduce the number of rules needed to specify the knowledge base taking advantage of the additive properties of the fuzzy system. On the other hand, despite of the good results of AddFCC in Banksearch, its clustering results in WebKB were bad. The problem of AddFCC comes from the way of combining criteria, where all the criteria contribute the same to the combination. This fact supports our belief in the need for a system where not all the criteria contribute the same to the combination.

All the criteria considered in EFCC come from document contents, since it is based on the same criteria as FCC. FCC was defined as self-content, and external information from inlinks or collection information was not considered. For EFCC, we tested the usefulness of IDF and anchor texts. The former lead to bad experimental results, particularly in WebKB and it was rejected. The latter, anchor texts, were added to the combination in several ways, but the results were not clearly better than those obtained by EFCC by itself. Besides, the cost of preprocessing anchor texts and their dependence on link density limit the applicability of this alternative. For these reasons, we believe that it could be an interesting option when a collection fulfills these requirements and time complexity is not a problem, but in most of the cases this will not happen and we will have to carry out document representation only with document contents.

To ensure that the application of our findings had a real effect compared to FCC, we performed statistical significance tests. EFCC showed better results in most cases. The good results obtained with vocabulary sizes below $1,000$ features are particularly interesting , since using smaller vocabularies allows to reduce the computational cost of the clustering.

Conclusions on Fitting Document Representation to Specific Datasets

Another issue we deal with in this thesis is whether different datasets should be represented in different ways. In other words, whether the representation could improve its clustering results by adapting to particular dataset characteristics. In Chapter 5 we aimed to study the possibility of automatically fitting EFCC to different datasets by adjusting its membership functions. Our main concern was not to use any other information than that included in the datasets themselves.

Our analysis of the datasets showed clear differences among them. While most of their term distributions in the whole document tend to power law with slight differences, we find that in WebKB the tendency is different, and this change is still more stressed for emphasized terms. Based on this analysis, we proposed a way of automatically establishing the basic membership function parameters, taking into account term frequency distributions and the original heuristics. We proposed a method to automatically adjust the membership functions basic parameters for all the criteria except position, because word positions on a page do not depend on anything else than the number of words in the document.

Then, we compared the resulting web page representation, called AFCC, with the previous ones in which it is based—FCC and EFCC—, and with TF-IDF as standard term weighting function. Our evaluation showed that adjusting the representation to concrete collections can help improve clustering results. We found that representations not tuned to fit concrete collections could perform reasonably well in most cases. Nevertheless, with our automated proposal it was possible to maintain or improve their results in common cases, i.e., those following Zipf's law, at the same time that the representation was able to deal with not so common cases, achieving very good results compared to the rest of alternatives tested in this thesis. Moreover, identifying those cases is not a computationally expensive task. We concluded that automatically tuning the system to fit the dataset is a feasible way of improving web page representation and clustering performance.

It is worth mentioning the case of SODP collection. The performance of all the representations with this dataset was really bad. This dataset is characterized by a strong imbalance of the number of documents within each category. This could introduce a bias towards the biggest categories, causing that the clustering algorithm favors the division of the documents belonging to those categories instead of finding the smaller ones. Compared with the fuzzy logic based approaches, TF-IDF achieved reasonably good performance. The use of IDF could be a key factor to explain this good behavior, because it could alleviate the effect of the biggest categories by giving more representativeness to terms coming from smaller categories. For this reason, we believe that TF-IDF can be useful to deal with collections where most of the documents belong to a small number of categories, while most of the categories contain a much smaller number of documents. In these cases, TF-IDF could allow to improve the overall clustering results by improving the clustering related to the small categories.

Conclusions on the Validation of our Proposals in a new Test Scenario

Lastly in this dissertation, we proposed a new test scenario for our proposals in Chapter 6. We applied hierarchical clustering to a taxonomy learning problem from a set of text documents that contain concept definitions. We considered that this kind of clustering environment is considerably different from the others presented in previous chapters and therefore it constitutes an appropriate field for validating our proposals. We performed our experiments over comparable corpora written in English, that is a Germanic language, and in Spanish, that belongs to the Romance language family. The whole dataset is composed of Wikipedia articles about animals (it was described in Section 3.5 on page 94).

Among languages, AFCC showed the best behavior, achieving particularly good results in English. Even in Spanish and with some issues related with the preprocessing step—stemming words is more appropriate in English than in Spanish—, and the length of the documents—Spanish documents are shorter than English ones—, the results of our representation proposals, AFCC and EFCC are, in the worst cases, at least comparable to TF-IDF. In particular, AFCC outperforms TF-IDF in most cases (there is only one case where TF-IDF achieved higher taxonomic F-measure results, but the difference was 0.001). Thus, AFCC showed its suitability for representing web pages in order to deal with a hierarchical clustering problem in at least two different languages.

Concluding Remarks

All things considered, we believe that in a real scenario AFCC would be the best option to represent documents among the alternatives tested in this thesis. In a real case we do not know what kind of dataset we could deal with. AFCC showed good results for the most common case, where term distributions tend to follow Zipf's law, and also for different collections, like WebKB, where the automatic tuning of our proposal produced a considerable improvement in clustering results. Even when we moved to hierarchical clustering in two languages, AFCC still was able to achieve good results.

7.2 Summary of Contributions

Summarizing, the main contributions of this research work are:

1. To review the literature about web page representation methods for clustering tasks.
    (a) To identify the most commonly used methods for representing web pages.
    (b) To summarize some relevant works on FRBS tuning and analyze whether or not they can be applied to web page representation for clustering.
    (c) To summarize some relevant works on taxonomy learning, particularly from the point of view of hierarchical clustering approaches.
2. To select and analyze four dataset to be used in the experimentation of this thesis.
    (a) From the point of view of categories and web domains, to determine the difficulties of applying clustering algorithms on them.
    (b) From the point of view of term distributions, to discover concrete characteristics that can be used to improve the representation of their web pages for clustering tasks.
3. To analyze the fuzzy combination of criteria performed by the selected fuzzy system (FCC) aiming to get the most of the fuzzy system and heuristics in which it is based.

4. To present and evaluate an improved representation called EFCC, and another alternative called AddFCC.
5. To present and evaluate a dimension reduction method called MFT, a lightweight dimension reduction technique, based on the term weighting function, able to improve the results of more complex techniques as LSI when used together with EFCC.
6. To apply the combination of EFCC and MFT as a general method to represent web pages for clustering in four different collections
7. To extend SODP collection by adding anchor texts corresponding to inlinks to collection web pages.
8. To evaluate the inclusion of some new criteria to the representation: IDF and anchor texts.
9. To propose a method to adapt EFCC to fit concrete characteristics of different collections. The result was the AFCC representation method.
10. To validate EFCC and AFCC in a hierarchical clustering environment, with a comparable corpus written in English and Spanish. This corpus is available for research purposes[1].
11. To identify the contexts where our proposals can achieve good results.

Part of the work presented in this thesis has also been published in several international conferences and journals. A detailed list of these publications is available in Appendix A.

7.3 Future Directions

This thesis points out several conclusions about the use of fuzzy logic for web page representation, but there are also some open issues:

- To consider the study of the effect of non-linear scaling factors (see Figure 5.1 on page 147) as a complementary tool to our proposal to adjust the representation to concrete datasets. We modified the membership function basic parameters, but exploring a way of using non-linear scaling factors from an unsupervised point of view it would be also interesting.
- To explore whether partial clustering solutions could be used for system tuning. Other approaches used category information that is not available in clustering. Thus, it would be interesting to explore whether or not partial clustering solutions could be used instead.

[1] WAD dataset can be downloaded at `http://nlp.uned.es/~alpgarcia/wad.php`

- To propose a standard benchmark to compare web page representations for clustering tasks. We tried to keep our comparison as standard as possible, but actually there is no standard way to perform this kind of comparison. This is an important point to grant compatibility among different works and it would make possible to easily compare results of different proposals. This benchmark should be composed of different web page collections with different features and a specific evaluation methodology to test different approaches.

- To study new ways of considering the `position` criterion. It is now the least important in the combination, but maybe its definition could be modified. The use of the DOM tree or even visual analysis could be alternative ways to find the positions of the page where words could be considered more important. There are works in the literature following a similar approach, but, to the best of our knowledge, all of them use category information to establish the importance of concrete page parts.

- To study new criteria to include in the combination. It would be interesting to compare the effect of new criteria on the combination, adding them one by one. This is not a trivial task, as adding new criteria would imply to modify the rule set and therefore adding new heuristic knowledge to the system.

- To learn the rule set from a set of examples. It would be very interesting trying to find a set of rules to represent a collection of documents from a set of preclassified document samples. The idea would be to analyze whether the resulting rules are coherent and could correspond to some heuristic knowledge related with the documents. The initial idea could be something similar to the proposals we can find in the literature about FRBS tuning, but taking into account that, different from other works, our objective function is not directly the output of the fuzzy system, but the result of a grouping process applied over a set documents that are represented by the outputs of the system.

Bibliography

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, August 2009. ISSN 1386-4564. 116

Antonelli, M., Ducange, P., Lazzerini, B., and Marcelloni, F. Learning knowledge bases of multi-objective evolutionary fuzzy systems by simultaneously optimizing accuracy, complexity and partition integrity. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 15:2335–2354, 2011. ISSN 1432-7643. 145

Benz, D. and Hotho, A. Position paper: Ontology learning from folksonomies. In Hinneburg, A., editor, *LWA 2007: Lernen - Wissen - Adaption, Workshop Proceedings (LWA)*, pages 109–112. Martin-Luther-University Halle-Wittenberg, 2007. ISBN 978-3-86010-907-6. 169

Bezdek, J. C., Ehrlich, R., and Full, W. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191 – 203, 1984. ISSN 0098-3004. 69

Bingham, E. and Mannila, H. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 245–250, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. 62, 117

Bloehdorn, S., Cimiano, P., and Hotho, A. Learning ontologies to improve text clustering and classification. In Spiliopoulou, M., Kruse, R., Nürnberger, A., Borgelt, C., and Gaul, W., editors, *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the German Classification Society (GfKl 2005), March 9-11, 2005, Magdeburg, Germany*, volume 30 of

Studies in Classification, Data Analysis, and Knowledge Organization, pages 334–341. Springer, Berlin–Heidelberg, Germany, 2006. 166

Boley, D. Principal direction divisive partitioning. *Data Min. Knowl. Discov.*, 2(4): 325–344, dec 1998. ISSN 1384-5810. 44

Booker, L. B., Goldberg, D. E., and Holland, J. H. Classifier systems and genetic algorithms. *Artif. Intell.*, 40(1-3):235–282, September 1989. ISSN 0004-3702. 145

Borgs, C., Chayes, J., Kalai, A. T., Malekian, A., and Tennenholtz, M. A novel approach to propagating distrust. In *Proceedings of the 6th international conference on Internet and network economics*, WINE'10, pages 87–105, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-17571-6, 978-3-642-17571-8. 43

Brank, J., Grobelnik, M., and Mladenic, D. A survey of ontology evaluation techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, 2005. 171

Brewster, C. and Wilks, Y. Ontologies, taxonomies, thesauri: Learning from texts. In Deegan, M., editor, *Proceedings of the Workshop on the Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content*, 2004. 167, 172

Brewster, C., Ciravegna, F., and Wilks, Y. Background and foreground knowledge in dynamic ontology construction. In *In Proceedings of the SIGIR Semantic Web Workshop*, 2003. 172

Buitelaar, P. *Semantic lexicons: Between terminology and ontology*, pages 16–24. MIT Press, 2000. 168

Cai, D., Yu, S., Wen, J. R., and Ma, W. Y. Extracting content structure for web pages based on visual representation. In *Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications*, APWeb'03, pages 406–417, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-02354-2. 47

Carullo, M., Binaghi, E., and Gallo, I. An online document clustering technique for short web contents. *Pattern Recognition Letters*, 30(10):870 – 876, 2009. ISSN 0167-8655. 67

Casillas, J., Cordón, O., del Jesús, M. J., and Herrera, F. Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE T. Fuzzy Systems*, 13(1):13–29, 2005. 144, 145, 146

Chen, J., Shankar, S., Kelly, A., Gningue, S., and Rajaravivarma, R. An adaptive bottom up clustering approach for web news extraction. In *Proceedings of the 18th international conference on Wireless and Optical Communications Conference*, WOCC'09, pages 64–68, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-5217-0. 42

Chen, T. S., Tsai, T. H., Chen, Y. T., Lin, C. C., Chen, R. C., Li, S. Y., and Chen, H. Y. A combined k-means and hierarchical clustering method for improving the clustering efficiency of microarray. In *Intelligent Signal Processing and Communication Systems, 2005. ISPACS 2005. Proceedings of 2005 International Symposium on*, pages 405 – 408, dec. 2005. 49

Choi, S. Independent component analysis. In *Encyclopedia of Biometrics*, pages 735–741. 2009. 62

Ciaramita, M., Gangemi, A., Ratsch, E., Šaric, J., and Rojas, I. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, pages 659–664, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc. 168

Cimiano, P., Hotho, A., and Staab, S. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence research*, 24:305–339, 2005. 168

Cordón, O. A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning*, 52(6):894 – 913, 2011. ISSN 0888-613X. 146

Cordón, O., Herrera, F., and Sánchez, L. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence*, 10: 5–24, January 1999. ISSN 0924-669X. 145

Correa, R. and Ludermir, T. A hybrid som-based document organization system. In *SBRN '06*, pages 16–16, Oct. 2006. 64

Crabtree, D., Andreae, P., and Gao, X. Qc4 - a clustering evaluation method. In Zhou, Z.-H., Li, H., and Yang, Q., editors, *Advances in Knowledge Discovery and Data Mining*, volume 4426 of *Lecture Notes in Computer Science*, pages 59–70. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-71700-3. 115

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. Learning to construct knowledge bases from the world wide web. *Artif. Intell.*, 118:69–113, April 2000. ISSN 0004-3702. 76, 83

Dellschaft, K. and Staab, S. On how to perform a gold standard based evaluation of ontology learning. In *In Proceedings of the 5th International Semantic Web Conference (ISWC'06*, pages 228–241. Springer, 2006. 172

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977. 69

Dhillon, I. S., Fan, J., and Guan, Y. *Efficient Clustering of Very Large Document Collections*, page 357–381. Massive Computing. Kluwer Academic Publishers, 2001. ISBN 9781402001147. 37

D'hondt, J., Vertommen, J., Verhaegen, P. A., Cattrysse, D., and Duflou, J. R. Pairwise-adaptive dissimilarity measure for document clustering. *Inf. Sci.*, 180: 2341–2358, June 2010. ISSN 0020-0255. 77

Dias, D. B., Madeo, R. C. B., Rocha, T., Bíscaro, H. H., and Peres, S. M. Hand movement recognition for brazilian sign language: a study using distance-based neural networks. In *Proceedings of the 2009 international joint conference on Neural Networks*, IJCNN'09, pages 2355–2362, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3549-4. 145

Dredze, M., Jansen, A., Coppersmith, G., and Church, K. Nlp on spoken documents without asr. EMNLP, pages 460–470, 2010. 70

Eiron, N. and McCurley, K. S. Analysis of anchor text for web search. In *Proceedings of the 26th SIGIR*, pages 459–460, 2003. 48, 136

Farahat, A. and Kamel, M. Enhancing document clustering using hybrid models for semantic similarity. In *Proceedings of the Eighth Workshop on Text Mining at the SDM*, 2010. 138

Faure, D. and Nédellec, C. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *In LREC workshop on*, pages 5–12, 1998. 168

Fernández, S., SanJuan, E., and Torres-Moreno, J. M. Textual energy of associative memories: performant applications of enertex algorithm in text summarization and topic segmentation. In *Proceedings of the artificial intelligence 6th Mexican international conference on Advances in artificial intelligence*, MICAI'07, pages 861–871, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76630-8, 978-3-540-76630-8. 55

Fersini, E., Messina, E., and Archetti, F. A probabilistic relational approach for web document clustering. *Information Processing & Management*, 46(2):117 – 130, 2010. ISSN 0306-4573. 47, 58

Fisher, M. and Everson, R. When are links useful? experiments in text classification. In *Advances in Information Retrieval*, volume 2633, pages 547–547. 2003. 45

Forsati, R., Mahdavi, M., Kangavari, M., and Safarkhani, B. Web page clustering using harmony search optimization. In *CCECE*, pages 1601 –1604, 2008. 67

Fresno, V. *Representacion autocontenida de documentos HTML: una propuesta basada en combinaciones heuristicas de criterios*. PhD thesis, 2006. 28, 29, 30, 40, 58, 69, 71, 75, 77, 84, 106, 114, 117, 182, 222

Fresno, V. and Ribeiro, A. An analytical approach to concept extraction in html environments. *J. Intell. Inf. Syst.*, 22(3):215–235, 2004. 26, 40, 56, 104, 105

Gacto, M. J., Alcalá, R., and Herrera, F. Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Comput.*, 13:419–436, December 2008. ISSN 1432-7643. 145, 146

Gacto, M. J., Alcalá, R., and Herrera, F. Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems. *IEEE T. Fuzzy Systems*, 18(3):515–531, 2010. 145

Galavotti, L., Sebastiani, F., and Simi, M. Experiments on the use of feature selection and negative evidence in automated text categorization. In *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '00, pages 59–68, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-41023-6. 215

Garfield, E. *Citation indexing: its theory and application in science*. John Wiley & Sons Inc., New York (Reprinted by ISI Press, 1983), 1979. 43

Garrod, S. and Daneman, M. *Reading, Psychology of*. John Wiley & Sons, Ltd, 2006. ISBN 9780470018866. 106

Garza Villarreal, S. E., Martínez Elizalde, L., and Canseco Viveros, A. Clustering hyperlinks for topic extraction: An exploratory analysis. In *Proceedings of the 2009 Eighth Mexican International Conference on Artificial Intelligence*, MICAI '09, pages 128–133, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3933-1. 44

Getoor, L. Link mining: a new data mining challenge. *SIGKDD Explor. Newsl.*, 5: 84–89, July 2003. ISSN 1931-0145. 43

Ghani, R. and Kumar, M. Online cost-sensitive learning for efficient interactive classification. In *Budgeted Learning Workshop at the 27th ICML*, 2010. 69

Giannopoulos, G., Dalamagas, T., Eirinaki, M., and Sellis, T. Boosting the ranking function learning process using clustering. In *Proceedings of the WIDM*, pages 125–132, 2008. 69

Golub, K. and Ardö, A. Importance of html structural elements and metadata in automated subject classification. In Rauber, A., Christodoulakis, S., and Tjoa, A. M., editors, *Research and Advanced Technology for Digital Libraries, 9th European Conference, ECDL 2005, Vienna, Austria, September 18-23, 2005, Proceedings*, volume 3652 of *Lecture Notes in Computer Science*, pages 368–378. Springer, 2005. ISBN 3-540-28767-1. 40

Grefenstette, G. *Explorations in Authomatic Thesaurus Construction*. Kluwer, 1994. 168

Hamming, R. W. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160, 1950. 55

Hammouda, K. and Kamel, M. Distributed collaborative web document clustering using cluster keyphrase summaries. *Information Fusion*, 9(4):465 – 480, 2008. 26, 56, 104

Hansen, B. A fuzzy logic–based analog forecasting system for ceiling and visibility. *Weather and Forecasting*, 22(6):1319–1330, 2007. 26

Harris, Z. Distributional structure. *Word*, 10(23):146–162, 1954. 168

Hearst, M. A. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics. 167

Higa, R. and Tozzi, C. Prediction of protein-protein binding hot spots: A combination of classifiers approach. In *Advances in Bioinformatics and Computational Biology*, volume 5167, pages 165–168. 2008. 138

Hill, T. and Lewicki, P. *STATISTICS Methods and Applications.* StatSoft, Tulsa, OK., 2007. 68

Honkela, T. *Self-Organizing Maps In Natural Language Processing*. PhD thesis, 1998. 69

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997. 69

Hopfield, J. J. Neurocomputing: foundations of research. chapter Neural networks and physical systems with emergent collective computational abilities, pages 457–464. MIT Press, Cambridge, MA, USA, 1988. ISBN 0-262-01097-6. 55

Hopgood, A. A. *Intelligent Systems for Engineers and Scientists*. Taylor & Francis, 2011. ISBN 9781439821206. 112

Hu, X., Zhang, X., Lu, C., Park, E. K., and Zhou, X. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD*, pages 389–396, 2009. 51, 52, 53, 57

Huang, A. Similarity measures for text document clustering. In *Proceedings of the Sixth NZCSRSC*, pages 49–56, 2008. 35, 70

Huang, A., Milne, D., Frank, E., and Witten, I. Clustering documents using a wikipedia-based concept representation. In *Advances in Knowledge Discovery and Data Mining*, volume 5476, pages 628–636. 2009. 52, 57

Huang, S., Chen, Z., Yu, Y., and Ma, W. Multitype features coselection for web document clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 18 (4):448 – 459, 2006. 65, 66, 84, 134

Hubert, L. and Arabie, P. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. ISSN 0176-4268. 48, 58

Hull, D. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '93, pages 329–338, New York, NY, USA, 1993. ACM. ISBN 0-89791-605-0. 137

Hung, B. Q., Otsubo, M., Hijikata, Y., and Nishida, S. Hits algorithm improvement using semantic text portion. *Web Intelli. and Agent Sys.*, 8:149–164, April 2010. ISSN 1570-1263. 44

Hyvärinen, A. and Oja, E. Independent component analysis: algorithms and applications. *Neural Netw.*, 13(4-5):411–430, May 2000. ISSN 0893-6080. 62

Isakson, C. S. and Spyridakis, J. H. The influence of semantics and syntax on what readers remember. *Technical Communication*, 50(4):538–553, 2003. 106

Isermann, R. On fuzzy logic applications for automatic control, supervision, and fault diagnosis. *IEEE Transactions on Systems man and Cybernetics Part Asystems and Humans*, 28(2):221–235, 1998. 26

Ishibuchi, H., Yamamoto, T., and Nakashima, T. Hybridization of fuzzy gbml approaches for pattern classification problems. *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, 35:359–365, 2005. 145

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, September 1999. ISSN 0360-0300. 22

Johnson, W. and Lindenstrauss, J. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984. ISBN 9780821850305. 62

Karypis, G. CLUTO - a clustering toolkit. Technical Report #02-017, November 2003. 69, 115

Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *J. ACM*, 46: 604–632, September 1999. ISSN 0004-5411. 44

Kohonen, T. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. ISSN 0018-9219. 68

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., and Saarela, A. Self organization of a massive document collection. *Neural Networks, IEEE Trans. on*, 2000. ISSN 1045-9227. 64

Kohonen, T., Schroeder, M. R., and Huang, T. S., editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001. ISBN 3540679219. 68

Kosko, B. Global stability of generalized additive fuzzy systems. *IEEE Transactions on Systems, Man, and Cybernetics - C*, 28:441–452, 1998. 114

Kovacevic, M., Diligenti, M., Gori, M., and Milutinovic, V. Visual adjacency multigraphs - a novel approach for a web page classification. In *Proceedings of the Workshop on Statistical Approaches to Web Mining*, pages 38–49, 2004. 41

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. Trawling the web for emerging cyber-communities. In *Proceedings of the eighth international conference on World Wide Web*, WWW '99, pages 1481–1493, New York, NY, USA, 1999. Elsevier North-Holland, Inc. 44

Kwon, O. W. and Lee, J. H. Text categorization based on k-nearest neighbor approach for web site classification. *Inf. Process. Manage.*, 39:25–44, January 2003. ISSN 0306-4573. 39

Landauer, T. K., Foltz, P. W., and Laham, D. An Introduction to Latent Semantic Analysis. *Discourse Processes*, (25):259–284, 1998. 61, 117

Larson, R. Bibliometrics of the world wide web: an exploratory analysis of the intellectual structure of cyberspace. *Journal of The American Society for Information Science and Technology*, 1996. 43

Li, J. D., Zhang, X. J., and Gao, Y. Learning fuzzy rules for modeling complex classification systems using genetic algorithms. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 10, pages V10–576 –V10–580, oct. 2010. 145, 146

Li, Y., Luo, C., and Chung, S. M. Text clustering with feature selection by using statistical data. *Knowledge and Data Engineering, IEEE Transactions on*, 20(5): 641–652, 2008. 65

Liu, B. D., Chen, C. Y., and Tsao, J. Y. Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 31(1):32–53, 2001. 146

Liu, T., Liu, S., and Chen, Z. An evaluation on feature selection for text clustering. In *ICML*, pages 488–495, 2003. 65

Liu, X. and Lu, H. Fragile watermarking schemes for tamperproof web pages. In *Proceedings of the 5th international symposium on Neural Networks: Advances in Neural Networks, Part II*, ISNN '08, pages 552–559, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87733-2. 77

Liu, Y. and Liu, Z. An improved hierarchical k-means algorithm for web document clustering. In *ICCSIT*, pages 606–610, 29 2008-sept. 2 2008. 26, 49, 57, 67, 69, 104

Liu, Y. C., Wu, C., and Liu, M. Research of fast som clustering for text information. *Expert Syst. Appl.*, 38:9325–9333, August 2011. ISSN 0957-4174. 77

Lowe, W. *Towards a theory of semantic space*, pages 576–581. 2001. 37

Maedche, A., Pekar, V., and Staab, S. Ontology learning part one - on discovering taxonomic relations from the web. In Zhong, N., Liu, J., and Yao, Y., editors, *Web Intelligence*, pages 301–322. Springer Verlag, 2002. 168

Mahdavi, M., Chehreghani, M. H., Abolhassani, H., and Forsati, R. Novel metaheuristic algorithms for clustering web documents. *Applied Mathematics and Computation*, 201(1-2):441–451, 2008. 67

Manning, C. D. and Schütze, H. *Foundations of Statistical Natural Language Processing*. Mit Press, 1999. ISBN 9780262133609. 153

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to information retrieval*. Cambridge University Press, 1 edition, July 2008. ISBN 0521865719. 115

Massa, P. and Hayes, C. Page-rerank: using trusted links to re-rank authority. In *Proceedings of the WI-IAT*, pages 614 – 617, 2005. 43

Matharage, S., Alahakoon, D., Rajapakse, J., and Huang, P. Fast growing self organizing map for text clustering. In Lu, B.-L., Zhang, L., and Kwok, J., editors, *Neural Information Processing*, volume 7063 of *Lecture Notes in Computer Science*, pages 406–415. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-24957-0. 77

Mohamed, E. A., El-Beltagy, S. R., and El-Gamal, S. A feature reduction technique for improved web page clustering. In *Innovations in Information Technology*, pages 1–5, 2006. 63, 84

Molina, A., Sierra, G., and Torres-Moreno, J. M. La energía textual como medida de distancia en agrupamiento de definiciones. In *Proceedings of the 10th International Conference on Statistical Analysis of Textua Data, JADT*, 2010. 55

Montalvo, S., Martínez, R., Casillas, A., and Fresno, V. Multilingual news clustering: Feature translation vs. identification of cognate named entities. *Pattern Recognition Letters*, 28(16):2305–2311, 2007. 69

Morkes, J. and Nielsen, J. Concise, scannable, and objective: How to write for the web, 1997. 106

Noll, M. G. and Meinel, C. The metadata triumvirate: Social annotations, anchor texts and search queries. In *Proceedings of the WI-IAT*, volume 1, pages 640–647, 2008. 48, 136

Nozaki, K., Ishibuchi, H., and Tanaka, H. A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86(3): 251–270, 1997. ISSN 0165-0114. 145

Oikonomakou, N. and Vazirgiannis, M. A review of web document clustering approaches. In *Data Mining and Knowledge Discovery Handbook*, pages 921–943. 2005. 35, 67

Ozcan, R., Sengör Altingövde, I., and Ulusoy, Ö. In praise of laziness: A lazy strategy for web information extraction. In *ECIR*, pages 565–568, 2012. 84

Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120. 43

Pant, G. Deriving link-context from html tag tree. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 49–55, 2003. 48

Park, H. and Kwon, H. Filtering methods for feature selection in web-document clustering. In *ICCS*, volume 4488, pages 1218–1221. 2007. 65

Patel, D. and Zaveri, M. A review on web pages clustering techniques. In Wyld, D. C., Wozniak, M., Chaki, N., Meghanathan, N., and Nagamalai, D., editors, *Trends in Network and Communications*, volume 197 of *Communications in Computer and Information Science*, pages 700–710. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-22543-7. 35, 67

Paukkeri, M. S., García-Plaza, A. P., Pessala, S., and Honkela, T. Learning taxonomic relations from a set of text documents. In *IMCSIT*, pages 105–112, 2010. 71, 166, 180

Paukkeri, M. S., García-Plaza, A. P., Fresno, V., Unanue, R. M., and Honkela, T. Learning a taxonomy from a set of text documents. *Applied Soft Computing*, 12 (3):1138 – 1148, 2012. ISSN 1568-4946. 18, 71, 76, 94, 96, 166, 167, 168, 169, 175, 180

Pennacchiotti, M. and Pantel, P. A bootstrapping algorithm for automatically harvesting semantic relations. In *in Proceedings of Inference in Computational Semantics ICoS-06*, pages 87–96, 2006. 168

Pérez García-Plaza, A., Fresno, V., and Martínez, R. Web page clustering using a fuzzy logic based representation and self-organizing maps. In *Proceedings of the WI-IAT*, pages 851–854, 2008. 30, 69, 117, 122, 139

Pérez García-Plaza, A., Fresno, V., and Martínez, R. Una Representación Basada en Lógica Borrosa para el Clustering de páginas web con Mapas Auto-Organizativos. 42:79–86, 2009. 117

Pérez García-Plaza, A., Fresno, V., and Martínez, R. Fuzzy combinations of criteria: An application to web page representation for clustering. In *CICLing (2)*, pages 157–168, 2012a. 139

Pérez García-Plaza, A., Fresno, V., and Martínez, R. Fitting document representation to specific datasets by adjusting membership functions. In *FUZZ-IEEE*, 2012b. 162

Pérez García-Plaza, A., Zubiaga, A., Fresno, V., and Martínez, R. Reorganizing clouds: A study on tag clustering and evaluation. *Expert Systems with Applications*, 39(10):9483 – 9493, 2012. ISSN 0957-4174. 69

Peserico, E. and Pretto, L. Score and rank convergence of hits. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 770–771, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. 44

Pruski, C., Guelfi, N., and Reynaud, C. Adaptive ontology-based web information retrieval: The target framework. *IJWP*, 3(3):41–58, 2011. 166

Qi, X. and Davison, B. D. Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41:12:1–12:31, 2009. 39, 41, 42

Quan, T., Hui, S., and Fong, A. Mining multiple clustering data for knowledge discovery. In Grieser, G., Tanaka, Y., and Yamamoto, A., editors, *Discovery Science*, volume 2843 of *Lecture Notes in Computer Science*, pages 452–459. Springer Berlin / Heidelberg, 2003. 68

Ribeiro, A., Fresno, V., Garcia-Alegre, M. C., and Guinea, D. A fuzzy system for the web page representation. 2003. 114

Riloff, E. M. and Kozareva, Z.; Hovy, E. Learning and evaluating the content and structure of a term taxonomy. pages 50–57, 2009. 169

Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. *Commun. ACM*, 1975. ISSN 0001-0782. 24, 36

Sanchez, D. and Moreno, A. Web-scale taxonomy learning. In *Proceedings of Workshop on Extending and Learning Lexical Ontologies using Machine Learning, ICML05*, pages 53–60, 2005. 169

Sanderson, M. and Croft, B. Deriving concept hierarchies from text. In *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 206–213, 1999. 168

Sanderson, M. and Zobel, J. Information retrieval system evaluation: effort, sensitivity, and reliability. In *ACM SIGIR*, pages 162–169, 2005. 138

Shih, L. K. and Karger, D. R. Using urls and table layout for web classification tasks. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 193–202, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. 41

Sinka, M. P. and Corne, D. W. The banksearch web document dataset: investigating unsupervised clustering and category similarity. *J. Netw. Comput. Appl.*, 28: 129–146, April 2005. ISSN 1084-8045. 76, 77, 79

Snow, R., Jurafsky, D., and Ng, A. Y. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 801–808, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. 168

Specia, L. and Motta, E. A hybrid approach for extracting semantic relations from texts. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 57–64, Sydney, Australia, July 2006. Association for Computational Linguistics. 168

Spyridakis, J. H. Guidelines for authoring comprehensible web pages and evaluating their success. *Journal of the Society for Technical Communication*, pages 359–382, 2000. 106

Sridevi, U. K. and Nagaveni, N. An ontology based model for document clustering. *IJIIT*, 7(3):54–69, 2011. 166

Staab, S. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. In *ICML-Workshop on Ontology Learning*, 2005. 168

Strehl, A., Strehl, E., Ghosh, J., and Mooney, R. Impact of similarity measures on web-page clustering. In *In Workshop on Artificial Intelligence for Web Search (AAAI 2000*, pages 58–64. AAAI, 2000. 70

Takahashi, K., Miura, T., and Shioya, I. Clustering web documents based on correlation of hyperlinks. In *Data Engineering Workshops, 2005. 21st International Conference on*, pages 1225 – 1225, april 2005. 46, 58

Tan, Q. and Mitra, P. Clustering-based incremental web crawling. *ACM Trans. Inf. Syst.*, 28:17:1–17:27, 2010. 70

Tang, B., Shepherd, M., Milios, E., and I.Heywood, M. Comparing and combining dimension reduction techniques for efficient text clustering. In *Proceedings of the Workshop on Feature Selection for Data Mining, SDM*, 2005. 61, 63, 84, 117

Van de Cruys, T. *Mining for Meaning: The Extraction of Lexico-semantic Knowledge from Text*. Groningen dissertations in linguistics. 2010. ISBN 9789036744270. 62, 70

Van Rijsbergen, C. J. Foundations of evaluation. *Journal of Documentation*, 30: 365–373, 1974. ISSN 0022-0418. 115

Van Rijsbergen, C. J. *Information retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 9780408709293. 70

Venturini, G. Sia: A supervised inductive algorithm with genetic search for learning attributes based concepts. In *Proceedings of the European Conference on Machine Learning*, ECML '93, pages 280–296, London, UK, UK, 1993. Springer-Verlag. ISBN 3-540-56602-3. 145

Virtanen, S. Clustering the chilean web. In *Proceedings of the First Conference on Latin American Web Congress*, LA-WEB '03, pages 229–231, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2058-8. 44

Wall, M. E., Rechtsteiner, A., and Rocha, L. M. Singular Value Decomposition and Principal Component Analysis. In Berrar, D., Dubitzky, W., and

Granzow, M., editors, *A Practical Approach to Microarray Data Analysis*, chapter 5, pages 91–109. Kluwer Academic Publishers, Norwell, MA, March 2003. ISBN 9781402072604. 61

Wang, G. and Lochovsky, F. H. Feature selection with conditional mutual information maximin in text categorization. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 342–349, New York, NY, USA, 2004. ACM. ISBN 1-58113-874-1. 65

Wang, Y. and Kitsuregawa, M. Evaluating contents-link coupled web page clustering for web search results. In *CIKM*, pages 499–506, 2002. 26, 50, 56, 104, 134

Ward Jr., J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):pp. 236–244, 1963. ISSN 01621459. 68

Waugh, S. *Extending and benchmarking Cascade-Correlation: extensions to the Cascade-Correlation architecture and benchmarking of feed-forward supervised artificial neural networks*. PhD thesis, University of Tasmania, 1995. 145

Wong, W. *Learning Lightweight Ontologies from Text across Different Domains using the Web as Background Knowledge*. PhD thesis, University of Western Australia, 2009. 169

Xu, J. and Croft, W. B. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM. ISBN 0-89791-792-8. 169

Xu, Q. and Zuo, W. Extracting precise link context using nlp parsing technique. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 64–69, 2004. 48, 84

Yang, S. Y. and Hsu, C. L. A new ontology-supported and hybrid recommending information system for scholars. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, pages 379–384, sept. 2010. 166

Yang, Y. and Pedersen, J. O. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3. 65

Yang, Y., Slattery, S., and Ghani, R. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18:219–241, 2002. ISSN 0925-9902. 45

Yu, S., Cai, D., Wen, J. R., and Ma, W. Y. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 11–18, New York, NY, USA, 2003. ACM. ISBN 1-58113-680-3. 41

Zhao, Y. and Karypis, G. Criterion functions for document clustering: Experiments and analysis. In *Technical Report TR 01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN*, 2001. 67

Zhao, Y. and Karypis, G. Comparison of agglomerative and partitional document clustering algorithms. Technical Report 02-014, 2002. 70

Zhao, Y. and Karypis, G. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Mach. Learn.*, 55:311–331, 2004. 67, 70, 138

Zhao, Y., Karypis, G., and Fayyad, U. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10:141–168, 2005. ISSN 1384-5810. 70

Zhuhadar, L. and Nasraoui, O. Personalized cluster-based semantically enriched web search for e-learning. In *Proceeding of the 2nd international workshop on Ontologies and information systems for the semantic web*, pages 105–112, 2008. 69

Zubiaga, A., García-Plaza, A. P., Fresno, V., and Martínez, R. Content-based clustering for tag cloud visualization. In *ASONAM*, pages 316–319, 2009a. 69

Zubiaga, A., Martínez, R., and Fresno, V. Getting the most out of social annotations for web page classification. In *ACM DocEng*, pages 74–83, 2009b. 76, 90

A

Publications

Summary of Publications

Most of the research included in this dissertation was also published in conferences and journals that are listed below.

Peer-Reviewed Conferences

- Alberto Pérez García-Plaza, Víctor Fresno, Raquel Martínez. 2008. *Web Page Clustering Using a Fuzzy Logic Based Representation and Self-Organizing Maps.* International Conference on Web Intelligence and Intelligent Agent Technology (IEEE/WIC/ACM). Volume 1, Page(s): 851 - 854. Sydney, Australia.

  Quality Indicator: ERA[1] C

  This paper includes part of the preliminary research for Chapter 4.

- Mari-Sanna Paukkeri, Alberto Pérez García-Plaza, Sini Pessala, and Timo Honkela. 2010. *Learning taxonomic relations from a set of text documents.* In Proceedings of 5th International Symposium Advances in Artificial Intelligence and Applications (AAIA'10). Page(s): 105 - 112. Wisla, Poland.

  Quality Indicator: International Fuzzy Systems Association Award for Young Scientist.

  This paper includes part of research developed for Chapter 6.

[1] ERA refers to the conference ranking published in 2010 by the Excellence in Research for Australia initiative. This ranking is commonly used as a quality indicator for conferences.

- Alberto Pérez García-Plaza, Víctor Fresno, Raquel Martínez. 2012. *Fuzzy Combinations of Criteria: An Application to Web Page Representation for Clustering.* In Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2012). Pages(s): 157 - 168. New Delhi, India.

  Quality Indicator: ERA[1] B.

  This paper includes part of research developed for Chapter 4.

- Alberto Pérez García-Plaza, Víctor Fresno, Raquel Martínez. 2012. *Fitting Document Representation to Specific Datasets by Adjusting Membership Functions.* FUZZ-IEEE 2012, Brisbane, Australia.

  Quality Indicator: ERA[1] A.

  This paper includes part of research developed for Chapter 5.

Journals

- Alberto Pérez García-Plaza, Víctor Fresno, Raquel Martínez. 2009. *Una Representación Basada en Lógica Borrosa para el Clustering de páginas web con Mapas Auto-Organizativos.* Procesamiento del Lenguaje Natural, vol. 42, Pages 79 - 86.

  Quality Indicator: FECYT[2] Quality Seal for Scientific Spanish Journals. Spanish Foundation for Science and Technology.

  This paper includes part of the preliminary research for Chapter 4.

- Mari-Sanna Paukkeri, Alberto Pérez García-Plaza, Víctor Fresno, Raquel Martínez and Timo Honkela. 2012. *Learning a taxonomy from a set of text documents.* Applied Soft Computing. Volume 12, Issue 3, Pages 1138 - 1148, March 2012.

  Quality Indicator: 2011 JCR[3] Impact Factor = 2.612. Ranked Q1 in Computer Science, Artificial Intelligence and Computer Science, Interdisciplinary Applications.

  This paper includes part of research developed for Chapter 6.

[2] Fundación Española para la Ciencia y la Tecnología: `http://www.fecyt.es/`

[3] Journal Citation Report.

Workshops

- Agustín D. Delgado Muñoz, Raquel Martínez, Alberto Pérez García-Plaza and Víctor Fresno. 2012. *Unsupervised Real-Time Company Name Disambiguation in Twitter.* Workshop on Real-Time Analysis and Mining of Social Streams (RAMSS), 6th International AAAI Conference on Weblogs and Social Media (ICWSM-12). Page(s): 25 - 28. Dublin, Ireland.

  In this paper we applied our representation proposal (EFCC) to another different problem related with company name filtering in tweets. This research is not part of this dissertation, but an application of its results.

B

Key Terms and Definitions

Next, we list and provide the definitions for some of the most relevant terms related to this thesis:

Aboutness In Library and Information Science, it is often considered synonymous with subject (referred to the subject of a document).

Anchor text It is the clickable text of a hyperlink.

Category Each group of objects in the ideal solution of a dataset or gold standard.

Class Each label that can be assigned to an object during a classification process.

Classification In this thesis it is used to refer to the process of assigning a document to one or more classes from a set of predefined classes after a supervised or semi-supervised training process.

Cluster Each group of related objects obtained after applying a clustering algorithm.

Clustering It is the process of grouping related objects. This relation express similarity and it is calculated by means of a similarity function. Each group is called cluster, so that the objects belonging to the same cluster are more similar to each other than those in other clusters[1]. In contrast with classification, this process is unsupervised.

Feature The elements employed to characterize the pages. These features are used by clustering algorithms to find similarities among documents. In this dissertation, these elements are mainly the words that compound the documents.

Inlinks It refers to the hyperlinks that a given page receive from other pages, that is, the set of links that refers to a given page.

[1] http://en.wikipedia.org/wiki/Cluster_analysis

Outlinks The hyperlinks within a given web page that points to other web pages.

Term A term is basically a preprocessed word. This preprocessing essentially consists of removing punctuation marks, removing stop words, and stemming the words in order to reduce each word to its main part by removing affixes. However there is not a fixed preprocessing for all cases, reason why it is important to detail the particular steps carried out to transform words in terms.

Vocabulary A set of features we use to represent the documents in a particular collection.

C
List of Acronyms

This is a list of acronyms used in this thesis:

ACC Analytical Combination of Criteria.

AddFCC Additive Fuzzy Combination of Criteria.

AFCC Abstract Fuzzy Combination of Criteria.

DF Document Frequency.

DOM Document Object Model.

EFCC Extended Fuzzy Combination of Criteria.

FCC Fuzzy Combination of Criteria.

GSS (Galavotti-Sebastiani-Simi) coefficient (Galavotti et al., 2000).

FM Fuzzy Modeling.

FRBS Fuzzy Rule Based System.

GLC Graph Local Clustering.

HITS Hyperlink-Induced Topic Search.

HTML Hyper Text Markup Language.

ICA Independent Component Analysis.

IDF Inverse Document Frequency.

IR Information Retrieval.

LE Lexical Entity.

LSI Latent Semantic Indexing.

MFT Most Frequent Terms.

MOEA Multi-Objective Evolutionary Algorithm.

MSN Microsoft Network.

NP Noun Phrase.

PF Proper Function.

RP Random Projection.

SODP Social Open Directory Project dataset.

SOM Self Organizing Map.

SSM Semantic Space Model.

SVD Singular Value Decomposition.

TI Mean TF-IDF.

TF Term Frequency.

TFV Term Frequency Variance.

TREC Text REtrieval Conference.

URL Uniform Resource Locator.

VP Verb Phrase.

VSM Vector Space Model.

WAD Wikipedia Animal Dataset.

D

Anchor Stop Word List

This is a list of stop words for anchor texts used in this thesis:

blog
blogs
click
com
cool
css
forum
forums
here
homepage
hosting
http
info
link
list
network
online
org
powered
read
resource
resources
site
view
visit
watch
website

E

Resumen (Summary in Spanish)

Un Sistema Borroso Mejorado para la Representación de Páginas Web en Problemas de Clustering

Resumen

Mantener la información organizada es un factor clave para facilitar el acceso a la misma. Aunque la información que necesitamos a veces este disponible en la Web, esta información no es útil si no somos capaces de acceder a ella. Con este objetivo, es cada vez más habitual el uso de técnicas automáticas para agrupar documentos.

En esta tesis estamos interesados en el clustering de documentos, que consite básicamente en agrupar dichos documentos en base a la similitud de sus contenidos. A este respecto, la representación de los documentos juega un papel fundamental en el clustering de páginas web y constituye el foco principal de la investigación llevada a cabo en esta tesis. El lenguaje HTML es la alternativa más común para escribir páginas web. Este lenguaje contiene información explícita (etiquetas, en este caso) sobre su representación visual, la tipografía del texto o incluso su estructura, entre otras cosas. Es también un formato muy común en Internet. El objetivo principal de esta tesis es realizar un estudio en profundidad con la intención de aprovechar al máximo un modelo borroso de representación de documentos HTML para problemas de clustering.

Nuestro estudio se centra en la idea de descubrir si alguna parte del sistema puede ser explotada de una manera diferente que nos permita mejorar los resultados de clustering. Comenzamos nuestro trabajo analizando las partes del sistema que son susceptibles de mejora y estudiamos diferentes alternativas para realizar dichas mejoras. Por lo tanto, no proponemos un modelo de representación de documentos partiendo de cero, sino que lo construimos tratando de entender, en cada paso, sus diferentes aspectos.

Para la evaluación de nuestros resultados y la comparación de las diferentes propuestas de representación, utilizamos distintas colecciones de páginas web de referencia que fueron creadas previamente para ser utilizadas como *gold standards*. El clustering se realiza por medio de algoritmos del estado del arte y nuestras propuestas son validadas en entornos de clustering plano y jerárquico. Finalmente, también tratamos de comprobar la utilidad de nuestras aproximaciones para la representación de páginas web escritas en dos idiomas, Inglés y Español.

F
Conclusiones (Conclusions in Spanish)

Un Sistema Borroso Mejorado para la Representación de Páginas Web en Problemas de Clustering

Conclusiones

Esta sección está estructurada en dos bloques con la intención de proporcionar primero una visión general de la investigación llevada a cabo en esta tesis doctoral, para después profundizar en las conclusiones del trabajo de forma detallada y organizándolas por capítulos.

F.1 Breve Resumen de la Investigación Incluida en esta Tesis

El objetivo principal de esta tesis es la realización de un estudio en profundidad con la intención de explotar al máximo un modelo borroso de representación de documentos HTML orientado a problemas de clustering. Es habitual que las páginas web estén escritas en lenguaje HTML. Este lenguaje ofrece, entre otras cosas, información explícita (etiquetas en este caso) sobre su representación visual, la tipografía del texto o su estructura.

Nuestras propuestas están dirigidas hacia la creación de un método de representación de páginas web que nos permita expresar fácilmente el conocimiento experto acerca de cómo un ser humano ojea un documento para tratar de averiguar su tema principal. Mediante el uso de la lógica borrosa podemos separar la definición del conocimiento del procedimiento de cálculo. Además, permite que dicho conocimiento sea expresado por medio de un conjunto de reglas cercanas al lenguaje natural.

Teniendo todo esto en cuenta, analizamos tres aspectos diferentes de la representación de páginas web para clustering: las fuentes para la selección de rasgos de donde extraer la información esencial para la representación de páginas web, las funciones de pesado para estimar el peso de cada rasgo, y las técnicas de reducción de dimensiones para seleccionar los rasgos más representativos y reducir el coste computacional del clustering, que de otro modo podría no ser abordable en un tiempo razonable. Para la selección de rasgos exploramos algunos nuevos criterios para mejorar la representación con información de la colección o los textos de los enlaces. Respecto a las funciones de pesado, exploramos la combinación borrosa de criterios llevada a cabo por Fresno (2006) con el objetivo de obtener el máximo rendimiento (en términos de mejora del clustering) del sistema borroso y de las heurísticas en las que se basa. Usamos TF-IDF como *baseline*, ya que es un método de pesado estándar empleado en la representación de documentos. Presentamos una representación mejorada llamada EFCC y otra alternativa, llamada AddFCC, que funcionó peor que EFCC y fue descartada. Ambas alternativas proponen explotar el sistema borroso de una manera diferente que la original (FCC), aprovechando sus propiedades aditivas. En cuanto a la reducción de dimensiones presentamos MFT, una técnica de bajo coste computacional basada en la función de pesado, que se ha mostrado capaz de mejorar los resultados de clustering de otras técnicas más complejas como LSI cuando se usa junto a EFCC en nuestras colecciones de test. De este modo, propusimos la combinación de EFCC y MFT como método general de representación de páginas web para clustering, debido al buen rendimiento que mostró en tres colecciones diferentes en comparación con otras aproximaciones similares como FCC o AddFCC.

Además, queríamos estudiar si EFCC podría ser ajustada automáticamente a las características de diferentes colecciones. El objetivo de este ajuste no es sólo la mejora de los resultados de clustering, sino la adaptación de la representación a diferentes colecciones que podrían tener diferentes propiedades. Mostramos que la mayoría de las distribuciones de frecuencias de términos analizadas sobre nuestras colecciones de test siguen la ley de Zipf. Sin embargo, encontramos también el caso de WebKB, que se diferencia del resto particularmente en la distribución de términos enfatizados. Este hecho nos animó a estudiar el ajuste de sistemas borrosos desde un punto de vista no supervisado. Decidimos proponer un nuevo método de representación llamado AFCC, donde los parámetros básicos de las funciones de pertenencia son ajustados sobre la base de las distribuciones de términos de las colecciones. Vimos que AFCC mantuvo o mejoró los buenos resultados de EFCC y FCC en los casos más comunes, esto es, aquellos que siguen la ley de Zipf, siendo además capaz de tratar con éxito casos no tan comunes, donde mejoró los resultados de las otras representaciones.

Finalmente evaluamos los resultados de EFCC y AFCC en un entorno de clustering jerárquico. En particular, presentamos el problema de aprendizaje de taxonomías a partir de un corpus comparable compuesto por páginas web escritas en Inglés y en Castellano. Este escenario se utilizó para validar nuestras propuestas y explorar sus posibilidades en un idioma distinto del Inglés.

F.2 Conclusiones Detalladas por Capítulos

Para organizar y exponer claramente las conclusiones de este trabajo, esta sección se ha estructurado por capítulos.

Conclusiones sobre la Revisión de Propuestas Previas de Representación de Páginas Web

En el Capítulo 2 revisamos diferentes aproximaciones a la representación de páginas web desde el punto de vista de los problemas de clustering. Mostramos que los diferentes modelos de representación difieren principalmente en las fuentes de información que utilizan, las funciones de pesado que aplican sobre dicha información y las técnicas de reducción de dimensiones que emplean.

Entre las funciones de pesado, TF-IDF o a veces simplemente TF, son frecuentemente utilizadas. Cabe destacar que estas funciones se basan sólo en texto plano. Para mejorar los resultados de TF-IDF como método de representación de documentos, diferentes trabajos han propuesto el empleo de información adicional extraída de los contenidos de las páginas web. La mayoría de estos trabajos se apoyan en criterios como el título del documento, los fragmentos de texto enfatizados, las cabeceras, o información relacionada con los enlaces para enriquecer la representación.

Vimos que una aproximación habitual para la inclusión de este tipo de información en la representación es el pesado de cada término con una función, como por ejemplo TF o TF-IDF, basada en el número de ocurrencias de cada término dentro de cada criterio (título, cabeceras, etc.). Así, en un mismo documento obtendríamos diferentes pesos, en concreto uno para cada término en cada criterio. Para la combinación de los pesos de un mismo término dentro de un documento, la mayoría de los trabajos utilizan combinaciones lineales, donde la importancia de un término en un criterio concreto se calcula independientemente del resto de componentes de la combinación. Consideramos que este tipo combinaciones no constituyen la mejor opción para combinar criterios, porque no permiten expresar dependencias entre ellos. Además, las aproximaciones basadas en combinaciones lineales normalmente utilizan coeficientes para establecer la influencia de cada criterio en la combinación. Estos coeficientes son seleccionados de forma manual o empírica. De hecho, en algunos casos hemos visto que necesitan ser empíricamente ajustados a cada colección para obtener mejores resultados. Este hecho señala la posibilidad de que la combinación necesite diferentes ajustes para cada colección. Otros autores fijan los valores de estos coeficientes de antemano, pero la mayoría no explica las razones para la selección que realiza. Aunque estos coeficientes influencian en gran medida los resultados, hasta donde llega nuestro conocimiento, no hay propuestas para determinar sus valores de forma automática en ausencia de información previa acerca de las categorías presentes en la colección, ni siquiera recomendaciones para establecerlos cuando trabajamos con una colección con una serie de características particulares.

En otros casos, la combinación la realiza el propio algoritmo, perdiéndose así la independencia entre la representación y el proceso de clustering. En estos casos, para introducir una modificación en uno de los dos, representación o algoritmo, el sistema al completo ha de ser cambiado. Además, este tipo de aproximaciones no permite realizar comparaciones directas entre representaciones de documentos. De este modo, este tipo de aproximaciones dificulta analizar si los posibles beneficios o desventajas de una propuesta proceden de la representación o del algoritmo. Consideramos que la independencia entre ambos procesos es muy importante, porque nuestro objetivo es proponer un modelo de representación de páginas web que pueda ser aplicado en diferentes escenarios de clustering, y diferentes problemas pueden requerir el uso de distintos algoritmos de clustering.

Para permitir la definición de condiciones relacionadas a la hora de establecer la importancia de los términos—por ejemplo, un término que tenga una frecuencia alta en el documento debería aparecer también en el título o enfatizado para ser considerado importante—, en esta tesis doctoral nos interesamos en sistemas borrosos basados en reglas. Creemos que este tipo de sistemas son más adecuados para expresar el conocimiento heurístico sobre la combinación de criterios, ya que permiten centrar nuestra atención en definir las reglas de la misma, sin necesidad de especificar el procedimiento de cálculo. La lógica borrosa consiste en la combinación de un conjunto de expresiones lingüísticas basadas en palabras en lugar de valores numéricos y ha sido aplicada previamente a problemas de representación de documentos. Creemos que las combinaciones borrosas de criterios se ajustan mejor al problema de establecer la importancia de los términos, ya que existen dependencias entre los criterios que deberían ser tenidas en cuenta a la hora de enfrentarse a diferentes aspectos, como el estilo de escritura de diferentes autores, los automatismos en la creación de las páginas web (que pueden dar lugar a títulos generados de forma automática que nada tengan que ver con el contenido, por ejemplo), criterios que no siempre contribuyen a la combinación (como es el caso de la posición, que no siempre tiene por qué ser relevante), etc. La capacidad expresiva de las reglas y la no linealidad en la combinación, junto con la posibilidad de crear vectores dentro del Modelo de Espacio Vectorial (VSM) nos alentó a explorar la forma en la que los criterios pueden ser combinados por medio de un sistema borroso.

En este punto, el proceso de reducción de dimensiones puede también afectar a la efectividad de la representación. Dicho proceso debería eliminar rasgos poco representativos, ayudando a seleccionar aquellos más adecuados para encontrar relaciones entre los documentos pertenecientes a una misma categoría dentro de la colección. Algunos trabajos previos comparan diferentes aproximaciones como la reducción por Frecuencia de Documento (DF), la Proyección Aleatoria (RP), el Análisis de Componentes Principales (ICA) o el Índice de Latencia Semántica (LSI). Además, LSI, RP y DF son ampliamente utilizados en la literatura para reducir el número de dimensiones del vocabulario con el que se representan los documentos de una colección en problemas de clustering. Sin embargo, estos trabajos no analizan cómo se comportan estos métodos con diferentes funciones de pesado.

Por otro lado, además del contenido de los documentos, la estructura de enlaces ha sido también empleada para la representación de páginas web. En la mayoría de los casos, ambas fuentes de información se han usado de forma combinada. Estas combinaciones normalmente emplean funciones de pesado estándar basadas en el VSM para la parte del contenido. En este sentido, podríamos sustituir esta función por cualquier otra alternativa que siguiese también el VSM. De esta forma, una mejora en la representación en cualquiera de los dos ámbitos, la estructura de enlaces o el contenido, debería reflejarse en la combinación de ambos. También cabe destacar que existen trabajos que utilizan el texto de los enlaces (anchor text) combinado de forma lineal con el contenido. En estas combinaciones lineales, el texto de los enlaces es tratado de la misma forma que otros elementos pertenecientes a la página que ya hemos visto anteriormente, como por ejemplo los títulos o los términos procedentes del contenido de la página. En otros casos, los textos de los enlaces se usan como si directamente fueran parte del contenido de las correspondientes páginas, esto es, añadiéndolos al contenido de las páginas. Sin embargo, estas combinaciones normalmente incluyen otros elementos como los títulos de las páginas o la estructura de enlaces, y no hemos encontrado ningún estudio sobre la utilidad de añadir los textos de los enlaces en particular a los contenidos de la página para mejorar la representación en problemas de clustering.

Recientemente, Wikipedia se ha utilizado también como fuente externa de información para enriquecer la representación de documentos. Estas propuestas se apoyan en combinaciones lineales, donde los coeficientes están basados en resultados preliminares, para integrar la información procedente de Wikipedia con el contenido de los documentos que se desea representar. Como paso previo a la representación de documentos, estos métodos requieren procesar un corpus de documentos de Wikipedia para extraer información acerca de los conceptos que se utilizarán en dicha representación. En esta tesis no hemos utilizado Wikipedia como fuente de información externa. No obstante, podría ser una alternativa interesante para futuras líneas de investigación, ya que estos trabajos han mostrado resultados prometedores.

Con respecto a las colecciones empleadas en la evaluación, estas son diferentes casi en cada trabajo. No hay un conjunto o colección estándar de páginas web que se utilice para evaluar problemas de clustering de forma habitual. En este sentido, incluso cuando se utiliza la misma colección en diferentes trabajos, como es el caso de WebKB, cada uno la utiliza con un preprocesamiento distinto. Por ejemplo, es habitual usar sólo algunas de las categorías de la colección, o incluso a veces sólo parte de los documentos de esas categorías. Además de esto, el proceso de filtrado llevado a cabo no siempre está bien descrito.

Teniendo en cuenta todas las cuestiones mencionadas hasta aquí, percibimos la falta de una metodología estándar para comparar representaciones de páginas web. Cada trabajo establece su propio marco de trabajo y, aunque algunos aspectos coincidan en diferentes trabajos, estos trabajos no siguen un proceso o metodología común que permita obtener resultados comparables con trabajos similares. Por este motivo, en esta tesis intentamos independizar el proceso de representación del resto, centrando nuestra investigación y modificaciones principalmente en esta fase, a la vez que intentamos mantener el resto del marco de trabajo tan estándar como sea posible mediante el empleo de técnicas, algoritmos, colecciones y medidas ampliamente utilizadas en la literatura.

Conclusiones sobre la Selección y Análisis de las Colecciones de Páginas Web

En esta tesis doctoral empleamos cuatro colecciones de páginas web diferentes para evaluar nuestras propuestas. Todas ellas han sido descritas en el Capítulo 3. Vimos que Banksearch ofrece menos dificultades que WebKB y SODP a la hora de llevar a cabo el clustering, dado el balance equilibrado de documentos por categoría que ofrece y a una temática bien diferenciada entre categorías. WebKB es más complicada, dado que la mayoría de sus documentos provienen principalmente de cuatro Universidades y se organizan en categorías desbalanceadas en cuanto al número de documentos que contienen. La principal dificultad de WebKB es la heterogeneidad entre documentos que pertenecen a una misma categoría, debida al diseño original de las categorías que componen la colección. SODP es la más complicada de todas, con un mayor número de categorías desbalanceadas y mayores diferencias entre el número de documentos correspondiente a cada categoría. Extendimos esta colección recolectando textos correspondientes a enlaces entrantes que apuntaban a páginas de la colección, con la intención de que fuese posible incluirlos en la combinación de criterios. Finalmente, la colección denominada WAD fue creada para clustering jerárquico y todas sus páginas proceden de la Wikipedia. Además, por cada colección mostramos las distribuciones de términos en los diferentes criterios que se consideran en esta tesis. Básicamente, el Capítulo 3 contiene un análisis de las características de cada colección que son referenciadas desde capítulos posteriores.

Conclusiones sobre el Estudio de un Sistema Borroso para la Representación de Páginas Web

En el Capítulo 4 nos planteamos el problema de explotar de la mejor manera posible un modelo de representación de páginas web basado en un sistema de reglas borroso, aplicado a problemas de clustering.

Comenzamos nuestra investigación comprobando el efecto de diferentes técnicas de reducción de dimensiones y comparando una representación borrosa previa (FCC) con TF-IDF, con el objetivo de establecer un punto de partida para el análisis de dicho sistema borroso. Además, presentamos un técnica de reducción de dimensiones (MFT), que intenta seleccionar los términos más importantes para representar los documentos en una colección, en base a los resultados de la función de pesado aplicada sobre dichos términos. Mostramos que, con una función de pesado adecuada, es posible emplear técnicas de reducción ligeras como MFT, en lugar de otras alternativas más costosas como LSI, lo que implica una importante reducción en el coste computacional. También descubrimos que DF es particularmente útil en WebKB con TF-IDF, comparada con MFT. Creemos que hay tres factores fundamentales que favorecen este comportamiento: primero, el pequeño número de dominios web del que proceden las páginas web de esta colección; segundo, la heterogeneidad entre los documentos dentro de una misma categoría; y tercero, el desbalanceo en el número de documentos que contiene cada categoría. Con respecto al método RP, que es usado en la literatura como alternativa ligera a LSI, nuestros experimentos mostraron que no es una alternativa muy buena. Sus resultados de clustering fueron mucho peores que los que obtuvo LSI en la mayoría de los casos. Además, otras alternativas ligeras, como DF o MFT, también consiguieron mejores resultados que RP.

Nuestros experimentos iniciales con FCC y diferentes técnicas de reducción del número de rasgos mostraron el mal rendimiento de FCC en WebKB. Se realizó un análisis de FCC, representando los documentos con cada criterio por separado y comparando los resultados individuales con los de la combinación de criterios. A la vista de los resultados, los aspectos que en nuestra opinión perjudican a FCC son la excesiva importancia del criterio `posición` en la combinación y, a la vez, la infravaloración del resto de criterios, como se detalló en la Sección 4.5.2. En base a esto, propusimos dos alternativas para la combinación borrosa de criterios, AddFCC y EFCC. Nuestros experimentos mostraron que EFCC funcionó mejor que FCC por medio de una manera diferente de combinar los criterios, donde la frecuencia de un término en el documento es considerada tan discriminatoria como el título y el énfasis, y la posición se tiene en cuenta como el criterio menos importante. Esta propuesta también hizo posible reducir el número de reglas necesarias para especificar la base de conocimiento aprovechando las propiedades aditivas del sistema borroso. Por otro lado, a pesar de los buenos resultados de AddFCC en Banksearch, sus resultados de clustering en WebKB fueron peores. El problema de AddFCC viene de la manera en la que realiza la combinación de criterios, donde todos los criterios contribuyen lo mismo a la combinación. Este hecho apoya nuestra creencia al respecto de la necesidad de un sistema donde no todos los criterios aporten lo mismo en la combinación

Todos los criterios considerados en EFCC provienen del contenido de los documentos, dado que está basada en el mismo conjunto de criterios que FCC. Esta última fue definida como autocontenida y, por tanto, no se tuvo en cuenta ningún tipo de información externa procedente de enlaces o de la colección. En el caso de EFCC, tratamos de comprobar la utilidad de la función IDF y de los textos de los enlaces para mejorar la representación. En cuanto a IDF, nos llevó a obtener malos resultados, sobre todo en WebKB, y fue descartada. Los textos de los enlaces fueron añadidos a la combinación de diferentes maneras, pero los resultados obtenidos no fueron claramente mejores que los que obtuvo EFCC por si misma. Además, el coste de preprocesar los textos de los enlaces entrantes y su dependencia de la densidad de enlaces limitan la aplicabilidad de esta alternativa. Por estos motivos, creemos que si bien podría tratarse de una opción interesante cuando la colección en cuestión cumple los citados requisitos y el coste computacional no es un problema, en la mayoría de los casos esto no pasará y tendremos que llevar a cabo la representación de los documentos sólo por medio de su contenido.

Para asegurarnos de que la aplicación de nuestras modificaciones tuvo un efecto real en comparación con FCC, realizamos tests de significación estadística. En estos tests, EFCC mostró mejores resultados en la mayoría de los casos. Son particularmente interesantes los buenos resultados obtenidos con tamaños de vocabulario inferiores a $1,000$ rasgos, ya que el uso de un vocabulario más reducido permite reducir a su vez el coste computacional del clustering.

Conclusiones sobre el Ajuste de la Representación de Documentos a Colecciones Específicas

Otra cuestión que nos planteamos en esta tesis doctoral es si diferentes colecciones deberían ser representadas de diferente manera. En otras palabras, si se podrían mejorar los resultados de clustering adaptando el proceso de representación de documentos a las características particulares de una colección concreta. En el Capítulo 5 exploramos la posibilidad de ajustar EFCC de forma automática a diferentes colecciones de páginas web, por medio del ajuste de las funciones de pertenencia del sistema borroso. Nuestro objetivo era hacerlo utilizando únicamente la información contenida en las propias colecciones.

El análisis que realizamos sobre las colecciones mostró claras diferencias entre ellas. Mientras que la mayoría de las distribuciones de la frecuencia de los términos en el documento completo tiende a una ley de potencias (con ligeras diferencias entre ellas), encontramos que en WebKB esta tendencia es diferente, y este cambio es aún más acentuado en el caso del énfasis. En base al citado análisis, propusimos una forma de establecer los valores de los parámetros básicos de las funciones de pertenencia de forma automática, teniendo en cuenta las distribuciones de frecuencia de los términos en los diferentes criterios y las heurísticas originales. El único criterio que no se ajusta de forma automática a la colección es la posición, dado que las posiciones de un término en una página sólo dependen del número de palabras en dicha página.

La nueva propuesta, llamada AFCC, fue comparada con las anteriores en las que se basa, FCC y EFCC, y con TF-IDF como función de pesado estándar. Nuestra evaluación mostró que ajustar la representación a colecciones concretas puede ayudar a mejorar los resultados de clustering. Además, vimos que las representaciones que no se ajustaron a las colecciones también pueden proporcionar resultados razonablemente buenos en la mayoría de los casos. Sin embargo, con nuestra propuesta automática fue posible mantener o mejorar sus resultados en los casos más habituales, esto es, aquellos que siguen la ley de Zipf, al mismo tiempo que, en casos no tan habituales, se consiguió una mejora considerable respecto al resto de alternativas probadas en esta tesis doctoral. Además, identificar estos casos no es una tarea con elevado coste computacional. A la vista de todo esto, concluimos que el ajuste automático del sistema para adaptarlo a la colección que se esté representando es una alternativa viable para mejorar la representación de las páginas web y el rendimiento del clustering.

Merece la pena mencionar también el caso de la colección SODP. El rendimiento de todas las representaciones en esta colección fue realmente malo. SODP se caracteriza por un fuerte desbalanceo del número de documentos pertenecientes a cada una de sus categorías. Esto podría introducir un sesgo hacia las categorías de mayor tamaño, provocando que el algoritmo de clustering favorezca la división de los documentos pertenecientes a estas categorías, en lugar de la formación de las de menor tamaño. En comparación con las aproximaciones basadas en lógica borrosa, TF-IDF obtuvo unos resultados razonablemente buenos. El uso de la función IDF podría ser un factor clave para explicar su buen comportamiento, porque podría paliar el efecto de las categorías de mayor tamaño, dando mayor representatividad a los términos procedentes de las de menor tamaño. Por ello, creemos que TF-IDF puede resultar particularmente útil en colecciones donde la mayor parte de los documentos pertenecen a un reducido número de categorías, mientras que la mayor parte de las categorías contienen un número de documentos mucho menor. En esos casos TF-IDF podría permitir la mejora de los resultados globales de clustering mediante un mejor agrupamiento de las categorías pequeñas.

Conclusiones sobre Validación de Nuestras Propuestas en un Nuevo Escenario

Finalmente en esta tesis doctoral, en el Capítulo 6 propusimos un nuevo escenario para validar nuestras propuestas. Aplicamos clustering jerárquico en un problema de aprendizaje de taxonomías a partir de un conjunto de documentos de texto que contienen definiciones de conceptos. Consideramos que este tipo de entorno de clustering es considerablemente diferente de los presentados a lo largo de los capítulos previos en esta tesis y, por tanto, constituye un marco apropiado para validar nuestras propuestas. Realizamos la experimentación sobre un corpus comparable escrito en Inglés, que es un idioma Germánico, y en Castellano, que pertenece a la familia de lenguas Romance. La colección completa está compuesta de artículos sobre animales procedentes de Wikipedia (su descripción completa se encuentra en la Sección 3.5).

Entre idiomas, AFCC mostró el mejor comportamiento, alcanzando resultados particularmente buenos en Inglés. Incluso en Castellano y con algunas cuestiones relacionadas con el preproceso—el proceso de *stemming* de las palabras es más apropiado en Inglés que en Castellano—, los resultados de nuestras propuestas de representación AFCC y EFCC son, en los peores casos, al menos comparables a los de TF-IDF. En concreto, AFCC mejora los resultados de TF-IDF en la mayoría de los casos (sólo hay un caso donde TF-IDF obtuvo mejores resultados en cuanto a medida-F taxonómica, pero la diferencia fue 0,001). De este modo, AFCC mostró su idoneidad para la representación de páginas web en un problema de clustering jerárquico en al menos dos idiomas.

Conclusiones Finales

Considerando todos los aspectos previamente comentados, creemos que en un escenario real, AFCC sería la mejor alternativa para representar documentos de entre las evaluadas en esta tesis doctoral. En un caso real ignoraríamos el tipo de colección con la que podríamos trabajar. AFCC se comportó de forma adecuada para el caso más habitual, cuando las distribuciones de frecuencia de los términos tienden a seguir la ley de Zipf, así como en colecciones con distribuciones no tab habituales, como WebKB, donde estas distribuciones muestran en algunos casos variaciones significativas. De hecho, en este último caso, el ajuste automático produjo una mejora considerable en los resultados de clustering. Incluso cuando variamos sustancialmente el escenario de clustering, evaluando las propuestas en un entorno de clustering jerárquico en dos idiomas, AFCC fue capaz de conseguir buenos resultados.

F.3 Resumen de Contribuciones

En resumen, las principales contribuciones de esta tesis doctoral son:

1. Revisar trabajos previos relacionados con métodos de representación de páginas web orientados a problemas de clustering.
   (a) Identificar los métodos más habituales de representación de páginas web.
   (b) Revisar algunos trabajos previos relevantes sobre ajuste automático de sistemas borrosos basados en reglas y analizar si pueden aplicarse a la representación de páginas web para problemas de clustering.
   (c) Revisar algunos trabajos previos relevantes sobre aprendizaje de taxonomías, particularmente desde el punto de vista de las aproximaciones basadas en clustering jerárquico.
2. Seleccionar y analizar cuatro colecciones de páginas web para usarlas en la experimentación de esta tesis doctoral.
   (a) Desde el punto de vista de las categorías y los dominios web, para determinar las dificultadas de la aplicación de algoritmos de clustering sobre ellas.
   (b) Desde el punto de vista de las distribuciones de términos, para descubrir características concretas que puedan ser usadas para mejorar la representación de sus páginas web para problemas de clustering.
3. Analizar la combinación borrosa de criterios llevada a cabo por el sistema borroso seleccionado (FCC) con el objetivo de sacar el mejor partido posible tanto del sistema borroso como de las heurísticas en las que se basa.
4. Presentar y evaluar una representación mejorada, llamada EFCC, y otra alternativa llamada AddFCC.
5. Presentar y evaluar un método de reducción de dimensiones llamado MFT, una técnica de reducción de bajo coste computacional, basada en la función de pesado, y que ha sido capaz de mejorar los resultados de otras técnicas más complejas como LSI cuando se ha usado conjuntamente con EFCC.
6. Aplicar la combinación de EFCC y MFT como método general para representar páginas web para problemas de clustering en cuatro colecciones diferentes.
7. Extender la colección SODP añadiendo los textos correspondientes a los enlaces entrantes.
8. Evaluar la inclusión de algunos nuevos criterios a la representación borrosa: IDF y textos de los enlaces.

9. Proponer un método para adaptar EFCC a características concretas de diferentes colecciones. El resultado fue el método de representación AFCC.
10. Validar EFCC y AFCC en un entorno de clustering jerárquico, con un corpus comparable escrito en Inglés y Castellano. Este corpus está disponible para la comunidad científica y puede ser descargado para fines de investigación[1].
11. Identificar los contextos en los que nuestras propuestas pueden obtener buenos resultados.

F.4 Trabajo Futuro

A partir del trabajo realizado en esta tesis doctoral, aparecen algunas cuestiones abiertas que resultaría interesante tratar en sucesivos trabajos de investigación. Entre ellas cabe destacar las siguientes:

- Considerar el estudio del efecto de factores de escalado no lineal (ver Figura 5.1) como herramienta complementaria a nuestra propuesta de ajuste de la representación a colecciones concretas. En esta tesis doctoral modificamos los parámetros básicos de las funciones de pertenencia, pero explorar una forma de usar factores de escalado no lineal desde un punto de vista no supervisado sería también interesante.
- Explorar si soluciones parciales de clustering podrían ser usadas en el ajuste del sistema. Otras aproximaciones usaron información de las categorías que no está disponible en clustering. Por ello, sería interesante comprobar si soluciones parciales de clustering podrían usarse para sustituir dicha información acerca de las categorías.
- Proponer un marco de trabajo estándar para la comparación de representaciones de páginas web. En este trabajo hemos intentado mantener nuestra comparación tan estándar como ha sido posible. Sin embargo, no hay una manera estándar de realizar este tipo de comparaciones. Éste es un punto muy importante para garantizar la compatibilidad entre diferentes trabajos y haría posible comparar fácilmente los resultados obtenidos por diferentes propuestas. Este marco de evaluación debería estar compuesto por diferentes colecciones de páginas web con diferentes características y una metodología específica de evaluación para evaluar diferentes aproximaciones.
- Estudiar nuevas maneras de considerar el criterio `posición`. Ahora es el menos importante en la combinación, pero quizá su definición podría ser

[1] La colección completa puede descargarse en `http://nlp.uned.es/~alpgarcia/wad.php`

modificada. El uso del árbol DOM o el análisis visual podrían ser métodos alternativos para hallar las partes de la página en las que las palabras podrían ser consideradas más importantes. Existen propuestas en la literatura que siguen aproximaciones de este tipo, pero hasta donde llega nuestro conocimiento, todas ellas utilizan información relativa a las categorías para establecer la importancia de partes de la página concretas.

- Estudiar nuevos criterios para incluir en la combinación. Sería interesante comparar el efecto de nuevos criterios en la combinación, añadiéndolos uno por uno. Esto no es un trabajo trivial, ya que añadir nuevos criterios implicaría modificar el conjunto de reglas para añadir nuevo conocimiento heurístico al sistema.

- Inferir el conjunto de reglas del sistema a partir de un conjunto de ejemplos. Resultaría muy interesante intentar encontrar un conjunto de reglas para representar una colección de documentos a partir de un conjunto preclasificado de documentos de ejemplo. Por supuesto, habría que analizar si las reglas resultantes son coherentes y podrían corresponder a algún tipo de conocimiento heurístico relacionado con los documentos. La idea inicial podría ser un sistema similar a las propuestas que podemos encontrar en la literatura sobre el ajuste de sistemas borrosos basados en reglas. A diferencia de esos trabajos, en nuestro caso la función objetivo no es directamente la salida del sistema borroso, sino el resultado de un proceso de agrupamiento aplicado sobre un conjunto de documentos que son representados mediante las salidas del sistema.