Development of Novel Fuzzy Clustering Techniques in the Context of *e*-Learning

Maria Eduarda Silva Mendes Rodrigues

A thesis submitted for the degree of Doctor of Philosophy

November 2004

Department of Electronic and Electrical Engineering
University College London

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

To my dear husband and daughter,

Miguel and *Isabel*

Acknowledgements

First, I would like to thank my supervisor, Dr. Lionel Sacks, for introducing me to this field of research and for his continuous guidance, encouragement and confidence in my work.

I would like to thank Professor Eurico Carrapatoso, Professor José Ruela and Professor Pedro Guedes de Oliveira, whose support enabled me to undertake my postgraduate studies. I am also extremely grateful to the Portuguese Foundation for Science and Technology for awarding me with a doctoral scholarship through the PRAXIS XXI programme.

I would also like to thank my friends and colleagues at UCL, including all the present and past members of the ACSE research group, for providing a pleasant research environment and for making these last years a very enriching experience. A particular acknowledgement goes to Dr. Ognjen Prnjat for his comments on this thesis.

I would like to give a very special thanks to my family and friends for their constant support and encouragement. In particular, I wish to acknowledge the unconditional love and constant dedication of my parents Lucília and Rui. Their support has always been essential in all my accomplishments.

Most importantly, I would like to express my deepest thanks and appreciation to my dear husband Miguel, for his everlasting love, understanding, encouragement and help, which have been fundamental for me to bring this thesis into being and for making this journey worthwhile. I hope I will make him as proud of my achievements, as I am proud of him.

Finally, to my beloved baby daughter Isabel I owe my inspiration for the very last stages of this endeavour. I would like to thank her for bringing so much joy into my life.

Abstract

This thesis investigates the performance of fuzzy clustering for dynamically discovering content relationships in *e*-Learning material based on document metadata descriptions. This form of knowledge representation is exploited to enable flexible content navigation in *e*-Learning environments. However, the methods and tools developed in this thesis have wider applicability.

The purpose of clustering techniques is to determine underlying structures and relations in data sets usually based on distance or proximity measures. A number of clustering methods to suit particular applications have been developed throughout the years. This thesis specifically considers the well-known Fuzzy c-Means (FCM) clustering technique as the basis for document clustering.

Initially, novel expressions are developed to extend the FCM algorithm, which is based on the Euclidean metric, to an algorithm based on other proximity measures more appropriate for quantifying document relationships. These include the cosine, Jaccard and overlap similarity coefficients. This novel algorithm works with normalised k-dimensional data vectors that lie in hyper-sphere of unit radius and hence has been named Hyper-Spherical Fuzzy c-Means (H-FCM).

Subsequently, the performance of the H-FCM algorithm is compared to that of the FCM as well as conventional hard (*i.e.* non-fuzzy) clustering algorithms with respect to four test document collections. Both the impact of different proximity measures as well as the impact of pre-processing the document vector representations for dimensionality reduction are thoroughly investigated. Results demonstrate that the H-FCM clustering method outperforms both the conventional FCM method as well as hard clustering techniques.

This thesis also considers the integration of fuzzy clustering techniques in an end-to-end *e*-Learning system. In particular, a tool to convert the H-FCM document clustering outcome into a knowledge representation, based on the Topic Maps standard, suitable for

Web-based environments is developed. Moreover, a tool to enable flexible navigation of *e*-Learning material based on the fuzzy knowledge space is also developed. This tool is deployed in a real *e*-Learning environment where user trials are carried out.

Finally, this thesis considers the important problem of defining a suitable number of clusters for appropriately capturing the concepts of the knowledge space. In particular, an hierarchical H-FCM algorithm is developed where the sought granularity level defines the number of clusters. In this algorithm, a novel heuristic based on asymmetric similarity measures is exploited to link document clusters hierarchically and to form a topic hierarchy.

Contents

List of Figures

List of Tables

List of Abbreviations

CANDLE	Collaborative And Networked Distributed Learning Environment
DC	Dublin Core
DCMES	Dublin Core Metadata Element Set
DTD	Document Type Definition
CDF	Cumulative Distribution Function
CL	Complete-Link hierarchical clustering method
CSS	Cascading Style Sheet
CWBS	Compose Within and Between Scattering validity index
FCM	Fuzzy c-Means
FS	Fukuyama-Sugeno validity index
GA	Group-Average hierarchical clustering method
H-FCM	Hyper-spherical Fuzzy c-Means
H^2-FCM	Hierarchical Hyper-spherical Fuzzy c-Means
IDF	Inverse Document Frequency
IMS	Instructional Management System
IR	Information Retrieval
ISO	International Organization for Standardization
IST	Information Society Technologies
LOM	Learning Objects Metadata
LSI	Latent Semantic Indexing
LTSC	Learning Technology Standards Committee
NIST	National Institute for Standards and Technology
ODP	Open Directory Project
OT	Overlap Test
PC	Partition Coefficient

PE	Partition Entropy
RDF	Resource Description Framework
RR	Relevant - Relevant
RNR	Relevant - Non-Relevant
SGML	Standard Generalized Markup Language
SL	Single-Link hierarchical clustering method
SP	Specificity
STC	Suffix Tree Clustering
SVD	Singular Value Decomposition
TF	Term Frequency
TM	Topic Map
UML	Unified Modelling Language
URI	Unique Resource Identifier
W3C	World Wide Web Consortium
XB	Xie-Beni validity index
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
XSLT	XSL Transformations

Chapter 1
Introduction

1.1 Motivation

1.1.1 Flexible *e*-Learning environments

In recent years *e*-Learning systems have become widespread with bespoke solutions by individual institutions and standardising initiatives for learning technologies, such as those coordinated by the IEEE Learning Technology Standards Committee (LTSC) [1]. The concept *e*-Learning is generally applied to describe the use of network technologies for creating, delivering and facilitating a broad range of learning experiences, such as online learning and Web-based training.

Although existing *e*-Learning systems present many interesting advantages, the most obvious one being the possibility to access learning facilities anytime and anywhere, there are still many unsolved problems that need to be addressed to improve the effectiveness of the learning process in these systems [2]. The ineffectiveness of most *e*-Learning systems is mainly due to the fact that such systems have been designed to mimic the traditional classroom teaching approach. To tackle the current limitations of existing systems, many research projects have recently emerged. In particular, the support of various pedagogical models for flexible *e*-Learning has been addressed by the cross-European project CANDLE (Collaborative And Networked Distributed Learning Environment) [3, 4], which has provided the context for the research work presented in this thesis.

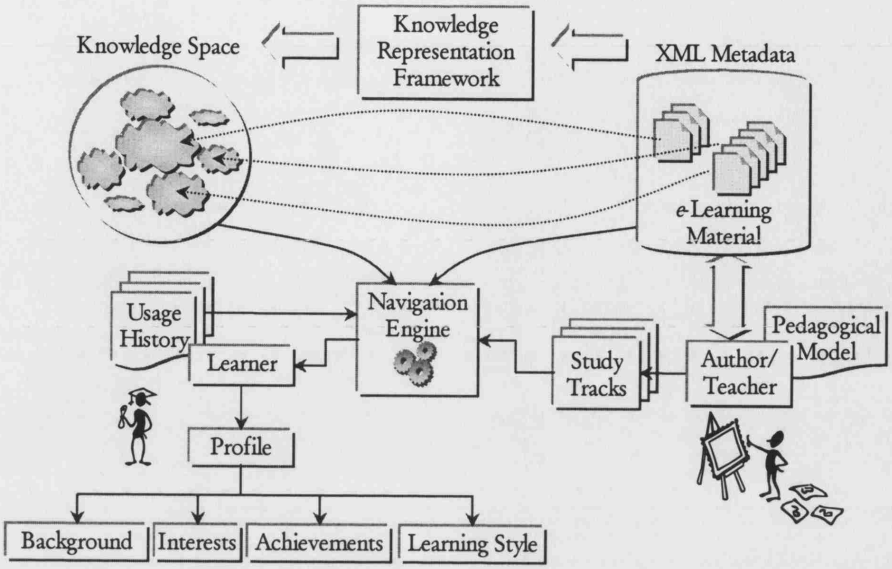


Figure 1.1: Adaptive knowledge-based content navigation for flexible *e*-Learning.

CANDLE has identified the need for flexible *e*-Learning environments through a literature study of pedagogical frameworks. Flexibility is required because each individual has specific needs and learning objectives. There are several flexibility aspects which should be considered. These include the provision of diverse communication facilities for learners to contact between themselves and with teachers/tutors, the support of different types of interactions within a given course to cope with different learning styles (*e.g.* learning in group or more isolated learning experiences), and the definition of flexible study programs in view of the learners background. The other important flexibility aspect, which is addressed in this thesis, is the support of different kinds of interactions with *e*-Learning material. This may range from relatively rigid training objectives to exploratory or research oriented interactions - the system should be flexible enough to accommodate the different learning contexts [5, 6]. Towards the more exploratory end of the learning spectrum, tools such as an adaptive navigation engine are required to determine which documents are the most relevant for a given user, who wants to learn a particular subject.

In Figure 1.1, a framework for adaptive knowledge-based content navigation is depicted. The learner's exploration of the available *e*-Learning material may be constrained by the pedagogical approach defined by the teacher. In a more instructional approach there would be a low degree of freedom as the learner would have to follow a specific sequence of links. More flexible approaches should also be supported. However, free exploration of

a vast number of information resources is not the best solution to this problem because of the ease with which learners can stray away from the original learning goals. A better option is to have a group of related material organised into an abstract knowledge space for guiding the learner in the browsing process. Such knowledge space can be visually represented for a better understanding of concept relationships and it can also be used by the navigation engine to derive links relevant to the learner's interests and objectives [7, 8, 9]. To achieve this, there needs to be a way to classify and organise *e*-Learning material in terms of knowledge domains. The research presented in this thesis is concerned with the representation of such knowledge relationships in *e*-Learning environments for flexible content navigation.

1.1.2 Dynamic knowledge representation

The CANDLE project has developed a repository of *e*-Learning material following recent standardising initiatives for learning technologies [10, 11, 12]. More specifically, each piece of content is tagged with XML (eXtensible Markup Language) [13] metadata according to an extended version of the IEEE LOM (Learning Objects Metadata) model [12]. Metadata can be defined as data about data and its fundamental roles are: i) to provide meaningful textual descriptions about resources (*e.g.* title, authors, technical requirements, conditions of use, etc.) and more importantly ii) to enable semantic interoperability, *i.e.*, to enable different systems to interpret and manipulate the same data [14].

Metadata is generally used to enhance the search and retrieval of content. However, it suits more advanced purposes such as knowledge-based content access. Particularly, educational metadata standards include a category to classify *e*-Learning material with keywords from a given taxonomy. The taxonomy provides a representation of some knowledge domain. Another emerging approach to represent knowledge is to develop a static ontology of the domain, defining a set of concepts and a set of relations between those concepts. By tagging each unit of *e*-Learning material with specific concepts, the relations defined in the ontology can be used to locate related material. This approach is a major feature of the more general enterprise of the Semantic Web [15, 16].

The rich semantic information captured by the ontology facilitates the access to *e*-Learning material both for authors and learners. Authors may start from one point in the ontology and follow the appropriate links to locate relevant material for their new courses. The learner may use the ontology for navigation and possibly automated location of

content. The key question with this approach is which ontology to use. Two problems can be foreseen. On the one hand, different experts in a given field are likely to disagree on the correct ontology. On the other hand, the actual ontology quickly changes through time as the field develops. Hence, the deployment and maintenance efforts are costly. Similar limitations have been identified in the context of the Semantic Web [17].

This problem motivates the development of a dynamic knowledge representation approach. In particular, this thesis investigates the use of fuzzy clustering techniques to dynamically discover the underlying knowledge structure and to identify knowledge-based relationships between *e*-Learning material based on the textual content of the XML metadata documents. We argue that the discovery of such relationships can be approached as a document clustering problem.

Clustering techniques fit in the broad area of pattern recognition and are generally applied for finding unobvious relations and structures in data sets. Specifically, document clustering methods are usually applied to group related documents based on similarities between their textual content. Fuzzy clustering techniques are explored in this thesis since they allow documents to have membership in multiple clusters. The degree of membership acknowledges that documents may contain information that is relevant to different knowledge domains. Thus, fuzzy clustering allows uncovering useful associations between domains. Furthermore, one of the limitations with formal ontologies is the need for the consensual definition of the "right" ontology. For most systems, knowledge can be ambiguous, can vary depending on the expert in the area and evolves through time. The imprecision and uncertainty associated with the underlying knowledge structures can be dealt with the theory of fuzzy sets, which is associated with fuzzy clustering techniques.

The details about the thesis organisation and an outline of the thesis contributions are given in the following sections.

1.2 Thesis organisation

Following this introductory chapter, an overview of the principles concerning the document clustering process, including a detailed discussion of each of the main phases of this process, are presented in Chapter 2. These phases consist of the representation of documents, the numerical encoding of document vector representations, pre-processing for dimensionality reduction, and the actual clustering phase. Suitable similarity measures to

determine document relationships as well as traditional document clustering methods are also reviewed in this chapter. Cluster validity measures are also discussed. The chapter concludes with a discussion on applications of document clustering algorithms that are related to the research in this thesis.

In Chapter 3, the selection of a similarity measure and of a fuzzy clustering method for document clustering is addressed. Initially, the typical properties of document collections are analysed and clustering tendency tests are applied to determine which similarity measure is more likely to produce better clustering performances. Subsequently, fuzzy clustering techniques are reviewed. In particular, the Fuzzy c-Means (FCM) clustering algorithm [18] is explored for document clustering. New mathematical expressions are derived for this clustering method in order to cluster unit length document vectors based on similarity coefficients. The Hyper-spherical Fuzzy c-Means (H-FCM) algorithm is developed as a result. Finally, some considerations about the characteristics of the H-FCM algorithm are presented.

In Chapter 4, the performance of the H-FCM algorithm is investigated through a set of experiments with test document collections. Initially, the new algorithm is compared with the original FCM algorithm in terms of internal performance and also in terms of the actual quality of the generated clusters. Then, the impact of pre-processing the document vectors on the H-FCM clustering outcome is analysed. Subsequently, a comparison of the H-FCM performance for different similarity coefficients is carried out. Finally, the H-FCM algorithm is compared with traditional hard clustering methods that have long been applied for document clustering.

In Chapter 5, tools that integrate the H-FCM algorithm for clustering and browsing *e*-Learning material are developed and evaluated in an end-to-end *e*-Learning system. The chapter starts with an overview of the CANDLE project. Then, suitable technologies for formal knowledge representation in Web-based applications are reviewed. In particular, the use of the Topic Maps standard is proposed for modelling in a platform-independent way the set of document and concept relationships that result from the document clustering process. Then, a tool that has been implemented for the dynamic generation of a topic map of the fuzzy knowledge space is described. Subsequently, the Knowledge Navigator system is described. This prototype tool has been implemented for flexible access to *e*-Learning material based on the fuzzy knowledge space relationships. The functionalities and user interface of this tool are also presented in detail. Finally, the deployment of the Knowledge

Navigator in a real *e*-Learning system is described and user trials are carried out in the context of the CANDLE project.

In Chapter 6, the issue of defining a suitable number of clusters for browsing large repositories of *e*-Learning material and for appropriately capturing the concepts of the knowledge space is addressed by developing a new hierarchical clustering algorithm, the Hierarchical Hyper-spherical Fuzzy c-Means (H^2-FCM). The chapter starts with a detailed description of the new algorithm. In particular, the use of an asymmetric similarity measure for creating a hierarchy of H-FCM fuzzy clusters is justified and the heuristic criteria the algorithm employs to generate the hierarchy are explained in detail. The performance of the H^2-FCM is then evaluated considering the clusters quality as well as its computational cost. The chapter concludes with a visual representation of the topic hierarchy that results from the hierarchical organisation of the cluster centroids.

Finally, Chapter 7 concludes this thesis summarising the main contributions and suggesting topics for future research.

1.3 Contributions

The primary goal of this research work was to evaluate the performance of fuzzy clustering methods for dynamically discovering content relationships in *e*-Learning material. This knowledge representation approach was explored to enable flexible content navigation in *e*-Learning applications. The main contributions of this thesis are summarised as follows:

- **Hyper-spherical Fuzzy c-Means algorithm (H-FCM)**: a modified Fuzzy c-Means (FCM) algorithm that applies similarity coefficients rather than the Euclidean distance for clustering unit length document vectors has been developed. In particular novel mathematical expressions have been developed for computing the cluster centroids in three different cases where three different similarity coefficients are applied. These are: cosine, Jaccard and overlap coefficients.
- **Hierarchical Hyper-spherical Fuzzy c-Means algorithm (H^2-FCM)**: a scalable hierarchical fuzzy clustering algorithm that takes an heuristic approach to organise H-FCM centroids hierarchically has been developed. The linking heuristic is based on the concept of asymmetric similarity and the number of

H-FCM clusters is selected according to the required granularity of the topics of each document cluster. The quality of the H^2-FCM cluster hierarchy has been evaluated and it has been shown that this algorithm presents good performance.

- **Evaluation of the performance of the H-FCM algorithm:** through a set of experiments it has been demonstrated that the new H-FCM algorithm performs better than the conventional FCM algorithm and that it also outperforms hard clustering algorithms, traditionally applied for document clustering. It has also been shown that the algorithm produces better results when document vectors are encoded with the TF term weighting scheme rather than with the TF-IDF scheme. It has been shown that these differences in performance were due to the fact that the TF-IDF scheme was de-emphasising term weights corresponding to the main topics of the document clusters. Furthermore, pre-processing effects on the performance of the algorithm have been analysed and it has been demonstrated that discarding very specific terms (*i.e.* that only appear in very few documents) leads to huge dimensionality reductions while maintaining or even improving the H-FCM clustering performance.
- **Implementation and deployment of the Knowledge Navigator:** tools based on standard and platform-independent technologies that integrate the H-FCM algorithm for clustering and browsing *e*-Learning material have been implemented. Furthermore, these tools have been deployed and evaluated in a real *e*-Learning environment where user trials were carried out. It has been shown how the knowledge space is obtained through the fuzzy clustering process and it has been demonstrated how the discovered relationships can be used to browse related material and to explore related topics.

These contributions have led to the following publications:

1. M.E.S. Mendes and L. Sacks, "Assessment of the performance of fuzzy cluster analysis in the classification of RFC documents," In: *Proceedings of the* 2000 *London Communications Symposium*, LCS 2000, London, UK, September 2000.

2. M.E.S. Mendes and L. Sacks, "Dynamic knowledge representation for e-Learning applications," In: *Proceedings of the* 2001 BISC *International Workshop on Fuzzy Logic and the Internet*, FLINT 2001, Memorandum No. UCB/ERL M01/28, pp. 176-181, U. C. Berkeley, USA, August 2001.

3. M.E.S. Mendes and L. Sacks, "From metadata to fuzzy knowledge representation," In: *Proceedings of the* 2001 *London Communications Symposium*, LCS 2001, pp. 135-138, London, UK, September 2001.

4. L. Sacks, A. Earle, O. Prnjat, W. Jarrett and M. Mendes, "Supporting variable pedagogical models in network based learning environments," In: *Proceedings of the* IEE 2nd *Annual Symposium on Engineering Education: Professional Engineering Scenarios*, ref. no.02/056, vol. 1, pp. 22/1-22/6, London, UK, January 2002.

5. M.E.S. Mendes, E. Martinez and L. Sacks, "Knowledge-based content navigation in e-Learning Applications," In: Proceedings of the London Communications Symposium 2002, LCS 2002, pp. 93-96, London, UK, September 2002.

6. M.E.S. Mendes and L. Sacks, "Evaluating fuzzy clustering for relevance-based information access," In: *Proceedings of the* 12th IEEE *International Conference on Fuzzy Systems*, FUZZ-IEEE 2003, pp. 648-653, St. Louis, Missouri, USA, May 2003.

7. M.E.S. Mendes, W. Jarrett, O. Prnjat and L. Sacks, "Flexible searching and browsing for telecoms learning material," In: *Proceedings of the* 2003 *International Symposium on Telecommunications*, IST'2003, Isfahan, Iran, August 2003.

8. M.E.S. Mendes and L. Sacks, "Dynamic knowledge representation for e-Learning applications," In: M. Nikravesh, L. A. Zadeh, B. Azvin and R. Yager (editors). *Enhancing the Power of the Internet - Studies in Fuzziness and Soft Computing*, Springer, vol. 139, pp. 255-278, January 2004.

9. M.E.S. Mendes Rodrigues and L. Sacks, "A scalable hierarchical fuzzy clustering algorithm for text mining," In: *Proceedings of the* 4th *International Conference on Recent Advances in Soft Computing*, RASC 2004, pp. 269-274, Nottingham, UK, December 2004.

Chapter 2
Document clustering

2.1 Introduction

In the previous chapter, the use of fuzzy techniques for clustering e-Learning material has been proposed. As this can be seen as a document clustering problem, in this chapter we present the relevant document clustering concepts that will be applied throughout this thesis.

In section 2.2, an overview of clustering and its application to information retrieval and related areas is presented. The basic steps of the document clustering process are then described in the subsequent sections. In section 2.3, suitable ways of representing and numerically encoding documents for clustering purposes are introduced and the use of pre-processing techniques for dimensionality reduction is discussed. Section 2.4 presents suitable measures of inter-document relationship and section 2.5 addresses traditional document clustering methods. In section 2.6, an overview of cluster validity measures is presented. Finally, a discussion on the application of fuzzy clustering methods in the e-Learning context is presented in section 2.7.

2.2 Clustering overview

Cluster analysis, or simply clustering, is one of the techniques used in the broad field of pattern recognition. Its purpose is to group a given set of unclassified objects into a number of clusters such that objects from the same cluster are in some way similar to each

other and dissimilar to objects from other clusters [19]. The clustering task should not be confused with the classification task. Classification or categorisation techniques are used to organise objects into pre-defined classes, whereas clustering methods discover the classes themselves [20]. Classification requires an initial training stage in which sample objects are classified to serve as a reference to new samples in the automated classification stage.

Clustering has long been used in a broad range of applications in many scientific fields like engineering, medical, natural and social sciences [20, 21, 22]. Information retrieval (IR) is a particular area where clustering techniques have been explored. The application of such methods in this field is particularly relevant to us because there is an overlap between the problems addressed in the IR context and in our application context.

Two main clustering approaches have been described in the IR literature: *term clustering* and *document clustering*. Term clustering addresses the grouping of related terms usually based on the documents in which they co-occur [23]. It has been investigated as a mechanism to increase the recall of document retrieval systems (*i.e.* to increase the proportion of relevant documents that are retrieved) either by: i) mapping terms occurring in queries and documents to their clusters identifiers or by ii) expanding queries with additional terms that belong to the same clusters as the terms in the original query. These methods were mostly abandoned in favour of document clustering ones, as they did not prove very successful in increasing retrieval effectiveness [24, 25].

The research presented in this thesis addresses specifically document clustering methods. In the context of IR, the original goal of document clustering was to improve search and retrieval efficiency by reducing the number of comparisons between documents and queries [20, 26]. Jardine and van Rijsbergen also suggested that retrieval effectiveness could be potentially improved with document clustering [26]. Their argument was based on the *cluster hypothesis* which states that documents relevant to a query tend to be more similar to each other than to irrelevant documents to the same query and thus are likely to be clustered together [27]. But this hypothesis has been contested [20, 28, 29] as no evidence was found that it generally holds, and the cluster-based approach to document retrieval has been found less effective than direct searches [28]. Hearst and Pedersen [30] took another approach and re-examined the cluster hypothesis for post-retrieval document clustering, concluding it was valid in this case. They presented evidence to support that dynamically generated clusters, as opposed to static document clustering, are tailored to the characteristics of the query leading to increased effectiveness.

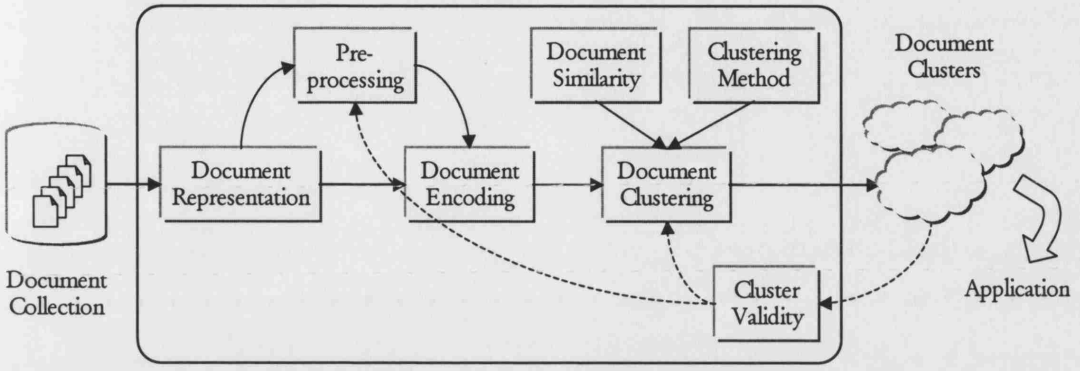


Figure 2.1: Phases of the document clustering process.

More recently, document clustering techniques have been applied in other contexts still related to information retrieval, such as organising Web search results [31, 32, 33, 34], browsing large document collections and browsing search results [35, 36, 37] or discovering frequently asked questions by clustering user query logs [38]. Document clustering has also been explored in data mining applications, more specifically as a text mining tool [39].

Regardless of the application, the document clustering process can be characterised by a typical set of steps [22, 40] that are illustrated in Figure 2.1. In this process each document is represented by a set of attributes, usually numerical, that are appropriately encoded for handling by the clustering algorithm. Pre-processing may be applied to control the exhaustivity of the document representations, by keeping only a subset of the original attributes. The clustering method assesses document relationships based on some similarity measure and both the clustering method and the similarity measure need to be tailored for a specific application. By applying the clustering algorithm documents are structured into a set of clusters, which are used according to the application purposes. Finally, the clusters generated can be validated through appropriate tests. Each of the phases of the document clustering process are overviewed in the subsequent sections of this chapter with special emphasis given to issues relevant to the research presented in this thesis.

2.3 Document representation and encoding

To apply a clustering algorithm to a document collection an appropriate document representation scheme needs to be defined. Depending on the type of clustering method, the nature of the attributes used to represent a document can be conceptual, numerical or even mixed. Most document clustering techniques work with numerical representations in the form of a k-dimensional vector,

$$x_i = [x_{i1}\ x_{i2} \ldots x_{ik}] \tag{2.1}$$

where k corresponds to the total number of terms used to index the document collection and x_{ij} represents the weight of term j in document x_i. This type of representation is typical of the classical models of IR [41].

There are several approaches for obtaining indexing terms, the most popular one being the use of single words contained in the text documents (the so-called *bag-of-words* approach). Other approaches include the use of syntactic [42, 43, 44] and statistical [45, 46, 47] information where phrases or groups of words are selected as indexing features. Such linguistic approaches have been extensively used in text classification systems and also in IR, but are far less common in document clustering applications, where the bag-of-words approach prevails. In recent years, document representation models based on phrase indexing in the form of suffix trees [31] and sentence graphs [48] have been proposed for document clustering with results pointing to slightly increased performance when compared to single word indexing. However, in the context of this thesis document representations will follow the bag-of-words approach because it requires less complex indexing procedures and its performance for document clustering is generally good.

Single words are usually extracted from the document texts through automatic indexing of the whole document collection as shown Figure 2.2. Common words like 'a', 'and', 'where', tend to exhibit high frequency of occurrence and are normally discarded [49]. Such terms, known as *stop words*, are useless for identifying the documents content and can be easily eliminated by keeping them in a dictionary or *stop list*. Stemming, *i.e.* the removal of word affixes such as 'ing', 'ion', 's', is also frequently applied since it allows to group terms with the same conceptual meaning (*e.g.* 'network', 'networking' and 'networked' are reduced to the same concept 'network') [49]. Both stop word elimination and stemming lead to a reduction in the size of the indexing structure. In particular, stop word elimination

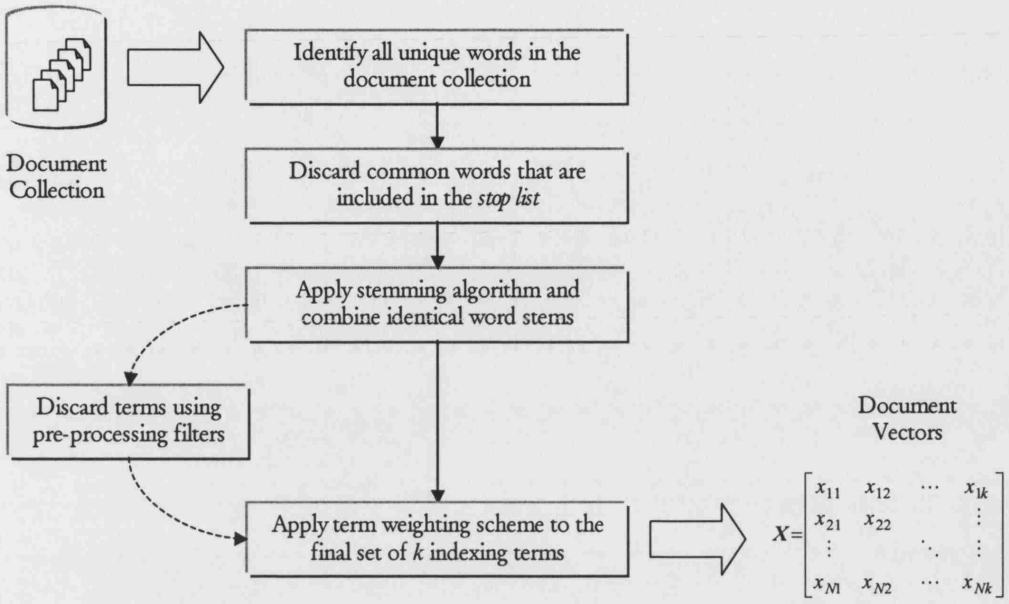


Figure 2.2: Procedure for automatic indexing of text documents.

results in a very high compression (forty percent or more) [41]. Further reduction in the number of indexing terms can be achieved through pre-processing filters, which can control the exhaustivity and specificity of the document representations. The remaining indexing terms that compose the final document vector representations are encoded by selecting an appropriate term weighting scheme. The following sub-sections address the main issues regarding pre-processing and term weighting.

2.3.1 Pre-processing

In general the total number of indexing terms for a given document collection is very large. Document representations of high dimensionality can be problematic for document clustering due to computational and storage costs. Dimensionality reduction techniques are thus applied because some of the indexing terms are likely to be useless to identify the documents content.

There are two main approaches for dimensionality reduction. One of them consists of the re-parameterisation of the original document representations, where new indexing variables are generated either by combining or transforming the existing indexing terms. An example of this approach is Latent Semantic Indexing (LSI), which has received much

attention in recent years both in IR and in text classification research. The idea behind LSI is that there is some latent structure between the original terms that can be captured in a reduced dimensionality space obtained through truncated Singular Value Decomposition (SVD) [50].

The other approach to dimensionality reduction, by far the most used one, consists in eliminating insignificant terms through filtering methods. Ideally, the remaining indexing terms should be able to describe all the relevant concepts expressed in the documents and as precisely as possible [27]. But in practice, a trade-off between exhaustivity and specificity of the document representation is required.

The question that arises is how to determine which terms are significant and which are irrelevant. It has been observed that when terms are arranged in decreasing order of their frequency of occurrence, the frequency of any given term multiplied by its rank order is approximately equal to the frequency of any other term multiplied by its rank [49]. This is known as Zipf's rank-frequency law [51] and it is expressed as:

$$frequency \times rank \approx constant. \tag{2.2}$$

Using this law it is possible to derive different methods for measuring term significance based on the frequency patterns. According to Luhn [52], the most significant terms are those that fall in the mid-frequency range (see Figure 2.3) and thus, low and high-frequency cut-offs can be set to eliminate all terms which fall beyond those thresholds.

Two related measures of term significance are *entropy* and *specificity* [49]. The entropy measure is derived from Shannon's information theory [53] and it acknowledges that the higher the probability of occurrence of a given term in the document collection, the less information it contains. The specificity measure acknowledges that the least important terms are those that just appear in a small percentage of documents (*i.e.* very specific terms) or that are present in almost every document (*i.e.* too general) and can hence be discarded.

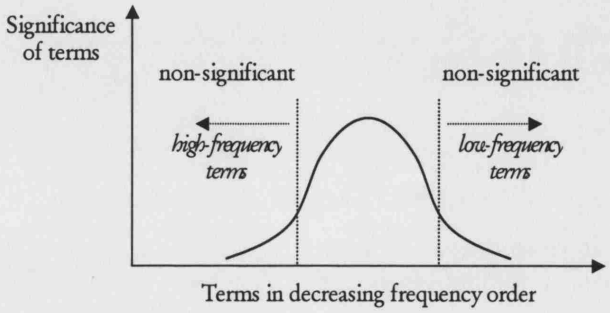


Figure 2.3: Significance of indexing terms as a function of their frequency of occurrence in the document collection.

Threshold levels for term significance measures, such as entropy and specificity, are usually selected based on heuristic criteria. Different cut-offs may lead to distinct performance of the document clustering application and in many cases the choice of the best cut-off is not straightforward.

Shaw [54, 55] has investigated the effects of indexing exhaustivity (*i.e.* the number of selected indexing terms) on the effectiveness of cluster-based IR. The studies were conducted with the Single-Link (SL)[1] hierarchical clustering method for a single document collection. The indexing exhaustivity was determined by setting different term weight thresholds. The results suggest that the SL method is significantly sensitive to the cut-offs used and that the best retrieval performance is achieved at relatively low levels of indexing exhaustivity, *i.e.* when more terms are discarded.

A study of the effects of indexing exhaustivity on the effectiveness of cluster-based IR was also carried out by Burgin [56]. Five hierarchical clustering algorithms, including SL, were applied to four document collections. The results confirmed once again that the SL method is rather sensitive to the pre-processing threshold used. However, the other four clustering methods not only proved to be quite insensitive to the indexing exhaustivity, but also performed significantly better than the former.

The effects of pre-processing through term filtering methods have also been analysed in text classification systems, where dimensionality reduction techniques are extensively employed. A comparison between term frequency thresholding and four other filtering methods (information gain, χ^2-statistic, mutual information and term strength) and their impact on the performance of two classifiers (nearest-neighbour (kNN) and linear least squares fit mapping (LLSF)) have been investigated in [57]. The outcome of the evaluation has revealed that term frequency thresholding, information gain and χ^2-statistic methods perform similarly and can be used with either classifier to eliminate a very high percentage of terms without losing classification accuracy. However, with the other filtering methods for equivalent performance more terms have to be kept.

Term filtering thresholding is a simple yet effective technique for dimensionality reduction which can be used in our application to reduce the computational cost of the document clustering process.

[1] The Single-Link clustering method is detailed in section 2.5.

2.3.2 Term weighting schemes

The document vector representation defined in (2.1) has its origins in classical IR. Two of the main models of IR, the Boolean model and the Vector model, use binary and non-binary term weights to represent documents, respectively [41]. These weights are a measure of the relative importance of the k indexing terms for identifying the document content. In the first case, binary weights simply acknowledge the presence or absence of terms in the document, whereas in the Vector model the weights are a function of the terms frequency within the document.

Non-binary document representations are usually favoured in IR because they usually lead to better retrieval effectiveness. Several weighting schemes have been proposed in the literature [58]. The simplest one assumes that the importance of a term is proportional to its frequency of occurrence within the document, *i.e.* $x_{ij} = f_{ij}$. This scheme is known as TF (Term-Frequency). Another scheme, known as TF-IDF (Term Frequency - Inverse Document Frequency), assumes that the importance of a term is not only proportional to its frequency but also a function of the total number of documents that contain the term [59]. With this scheme, terms that occur frequently in a small number of documents are attributed higher weights. The TF-IDF weights are usually given by the formula in (2.3) or variations of it,

$$x_{ij} = f_{ij} \cdot \log\left(N / n_j\right) \tag{2.3}$$

where f_{ij} represents the frequency of term j in document i, N the total number of documents and n_j the number of documents that have been indexed with term j.

Weighting schemes based on term frequencies lead to document representations with higher term weights in longer documents, as these contain more words than shorter ones. In IR this effect is undesirable because documents should have an equal chance of being retrieved in response to some query independently of their length [60]. This effect is eliminated by normalising the document length, *i.e.* $\|x_i\|=1$.

Another characteristic of these weighting schemes is that they assume information is homogeneously distributed in documents. However, such assumption may not hold in documents that are naturally structured into several sub-parts or sections (such as scientific papers, reports, etc.) or in documents structured into sub-parts by means of tags (such as HTML and XML documents). Some indexing approaches that consider different levels of

importance for different document parts have been proposed for improving Web retrieval [61] and for clustering HTML documents [48]. In [61], terms weights are firstly defined according to the importance associated with the tags in which the terms appear and then the individual weights are aggregated. In [48], three significance levels are used for different document sub-parts. Phrase frequencies are then weighted according to those significance levels for calculating document similarity.

Many document clustering applications have adopted the TF-IDF scheme, rather than TF or binary representations, mainly due to its performance in non-cluster based IR. However, very few studies on the performance of different weighting schemes for document clustering have been carried out. One of such attempts has been reported in [62]. Five term weighting schemes were compared for cluster-based IR, using the SL clustering method for three document collections. The retrieval effectiveness results provided no substantial evidence in favour or against a specific term weighting scheme.

2.4 Document similarity

Clustering methods attempt to group a given set of unlabeled objects into a number of meaningful clusters based on some notion of similarity or distance between those objects. Document clustering techniques usually rely on textual similarity (*i.e.* shared terms or phrases between documents), co-citation analysis or link structure to form document clusters. Co-citation analysis [63, 64, 65] and hyper-link structure [34, 65, 66] have been recently explored based on the assumption that documents sharing the same citations or hyper-links are very likely to be related. However, most applications employ methods which consider textual similarity as the basis for clustering documents. Here we focus specifically on this type of relationship between documents.

Distance functions and similarity coefficients (expressing the relationship between documents) have been widely used in the field of IR and in text classification systems. The formal definition of distance function is as follows: given a set of documents X to be clustered, a distance function $d: X \times X \rightarrow R_o^+$ is a measure of association between two documents belonging to a set X, that satisfies all the following properties [67]:

Reflexivity: $d(x_A, x_A) = 0, \forall x_A \in X$

Symmetry: $d(x_A, x_B) = d(x_B, x_A), \forall x_A, x_B \in X$

Triangle inequality: $d(x_A, x_B) + d(x_B, x_C) \geq d(x_A, x_C), \forall x_A, x_B, x_C \in X$

Indiscernibility: $d(x_A, x_B) = 0 \Rightarrow x_A = x_B, \forall x_A, x_B \in X$

Similarity measures, in general, do not satisfy the triangle inequality but they satisfy the reflexivity and symmetry properties. However, there are also some models of similarity that challenge the symmetry property. Tversky [68] argued that similarities based on human judgment are often quite asymmetric. He also argued that common attributes tend to increase the perceived similarity between objects more so than distinct attributes can diminish it. Tversky's model of similarity is able to capture inclusion relations between documents. In particular, a document vector with many indexing terms is considered to be less similar to a sparser document vector, rather than the opposite. Asymmetrical similarity measures that acknowledge inclusion relations have been recently proposed for document clustering [69] and for ranking documents in personal information filtering systems [70].

In general, the selection of the association measure for document clustering is not restricted, however, some clustering methods have theoretical requirements for a specific measure. For example, Ward's clustering method [71] requires the use of the Euclidean distance.

In IR, similarity coefficients are preferred to distance functions and are considered better measures of the inter-document relationship. In the next sub-section, an overview of such similarity coefficients is presented.

2.4.1 Similarity coefficients

A measure of the similarity $S(x_A,x_B)$ between two document vectors, x_A and x_B, represented according to (2.1) and encoded using binary or non-binary term weighting schemes, can be obtained using a number of functions. Table 2.1 contains the most common similarity coefficients used for document vector representations [27, 40, 49, 72, 73].

The inner product coefficient considers terms shared by both documents. In case of binary term weights, this measure counts the number of terms present in both documents. The other coefficients are also functions of the terms shared between documents.

Table 2.1: Similarity coefficients for document vector representations.

Coefficient	Similarity function: $S(x_A, x_B)$	
Inner product	$\sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}$	(2.4)
Cosine	$\frac{\sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}}{\left[\sum_{j=1}^{k} x_{Aj}^2 \cdot \sum_{j=1}^{k} x_{Bj}^2\right]^{1/2}}$	(2.5)
Dice	$\frac{2 \cdot \sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}}{\sum_{j=1}^{k} x_{Aj}^2 + \sum_{j=1}^{k} x_{Bj}^2}$	(2.6)
Jaccard	$\frac{\sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}}{\sum_{j=1}^{k} x_{Aj}^2 + \sum_{j=1}^{k} x_{Bj}^2 - \sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}}$	(2.7)
Overlap	$\frac{\sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}}{\min\left(\sum_{j=1}^{k} x_{Aj}, \sum_{j=1}^{k} x_{Bj}\right)}$	(2.8)

The cosine coefficient represents the cosine of the angle between the two document vectors. This measure is insensitive to different document lengths, since it is normalised by the length of the document vectors. The Dice coefficient is also normalised to range in the unit interval, but by the average length of the two document vectors. For binary weights this measure represents the ratio between shared terms and mean number of terms in both documents. The Jaccard coefficient is also normalised to range in the unit interval. For binary weights this measure gives the ratio between shared terms and the number of terms that occur in either documents. Finally, the overlap coefficient measures the overlap between two documents, considering the size of the shortest document. For non-binary weights, this coefficient may not range in the unit interval.

The cosine coefficient is the most widely used document similarity measure in IR. A study has compared the performance of the this coefficient to that of other coefficients in cluster-based IR [62]. The results have shown that the cosine and the Jaccard coefficients generally produce better levels of retrieval effectiveness, with the former performing slightly better than the later. This comparative study was limited though to the SL hierarchical clustering algorithm. A comparative study of document similarity measures has also been carried out in the context of visual IR interfaces [74]. The aim of the investigation was to determine how good the measures would be in expressing the known characteristics of the document space, *i.e.* the relevance of the documents to a given set of topics. Multi-dimensional scaling was used as the scaling method for visual presentation of relevant and non-relevant documents. Furthermore, the visual separation between groups was used to assess the performance of the similarity measures. The results have shown that only the cosine and overlap coefficients satisfactorily recover the structure of the document space.

2.5 Clustering methods

Clustering methods can be broadly divided into hierarchical and non-hierarchical or partitional methods. Hierarchical methods generate a cluster hierarchy whereas non-hierarchical methods generate a flat partition of the data objects into a set of clusters. Additionally, methods from each of these categories can be classified as monothetic or polythetic, as incremental or non-incremental, as deterministic or stochastic and as hard or fuzzy [22, 27].

Monothetic algorithms consider one attribute at a time to determine relationships between data objects during the clustering process whereas polythetic algorithms consider all attributes. The vast majority of clustering methods fall in the polythetic category.

Non-incremental algorithms require all data objects to be available from the very beginning, while incremental ones require only a small set of objects to generate clusters, which are subsequently refined incrementally.

Regarding the distinction between deterministic and stochastic algorithms, the former follow deterministic steps in the clustering process whereas the later include randomised steps, usually in the optimisation task.

Finally, the assignment of data objects to clusters can be either hard or fuzzy. In the first case each data object can only belong to a single cluster while in the fuzzy case objects can have different degrees of membership in more than one cluster. This thesis considers in detail the performance of fuzzy clustering methods.

A high number of clustering techniques, that suit a high variety of applications, can be found in literature [20, 21, 22]. Clustering techniques for document clustering usually generate hard clusters and are almost always classified as hierarchical or partitional. The characteristics of these two categories are presented in the next sub-sections.

2.5.1 Hierarchical methods

Hierarchical clustering techniques are perhaps the most widely used for document clustering [20]. There are mainly two approaches for hierarchical clustering: divisive and agglomerative. Divisive methods start with all data objects in a single cluster and sequentially split the clusters according to some clustering criterion until a pre-defined stopping condition is reached. Agglomerative methods start with each data element in a distinct cluster, called singleton, and sequentially merge the two most similar clusters until a pre-defined stopping condition is met. A possible termination condition is the final number of clusters.

The output of the hierarchical clustering algorithm, *i.e.* the divisions or fusions made at each successive stage of the algorithm, is normally represented by a tree-like graph called dendogram [19]. An example of such graphical representation for the agglomerative clustering case is shown in Figure 2.4, where a set of six objects (x_1 to x_6) are clustered into two groups (C_1 and C_2). The advantage of such a representation is that the dendogram can be cut at various levels of similarity, resulting in different number of clusters for different thresholds.

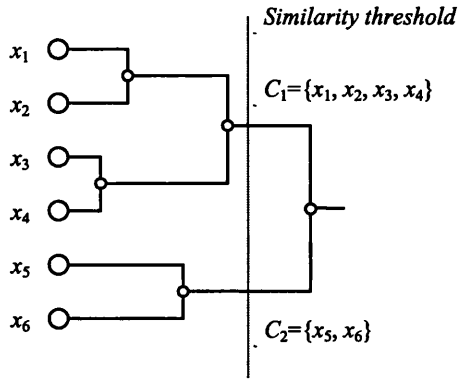


Figure 2.4: Dendogram representation of an agglomerative hierarchical algorithm.

Research on hierarchical document clustering, which has been mainly carried out by the IR community, predominantly considers agglomerative algorithms [20, 27]. These can be generically described by the pseudo-code presented in Figure 2.5. The various agglomerative methods available just differ on the way the distance (or the similarity) between clusters is defined (step 5 of the algorithm).

```
1. Start by determining all inter-document relationships (N×N
   calculations are required, N being the number of documents)
   using some distance function or similarity measure.
2. Generate N clusters and allocate a single document to each
   of those clusters.
3. Repeat
4.       Form a new cluster by merging the two closest clusters.
5.       Determine the distance/similarity between the new
         cluster and all other existing clusters.
6. Until all documents are in one cluster or until some other
   stopping criterion is satisfied.
```

Figure 2.5: Summary description of agglomerative hierarchical clustering algorithms.

Several linkage methods for cluster merging exist but the most common ones are the Single-Link method (SL), the Complete-Link method (CL), the Group-Average method (GA) and Ward's method [19, 20, 22, 71]:

- In the SL method (or nearest-neighbour) the distance between two clusters is the minimum distance between any pair of objects, one from each cluster (see Figure 2.6. a). This method tends to produce large elongated clusters with low internal cohesion.

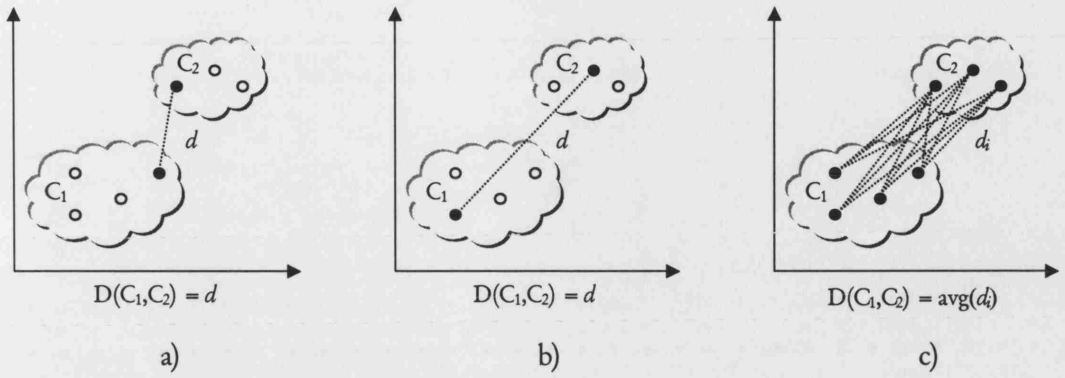


Figure 2.6: Distance between clusters in the a) Single-Link, b) Complete-Link, and c) Group-Average agglomerative hierarchical algorithms.

- In the CL method (or furthest neighbour), the distance between two clusters is the maximum distance between any pair of objects (see Figure 2.6. b). This method tends to produce a large number of compact clusters.
- In the GA method, the distance between two clusters is the average of all pairwise distances between objects of the two clusters (see Figure 2.6. c).
- In Ward's method, the closeness between two clusters is assessed based on the increase of the error sum of squared distances between the objects and the centroid of their cluster, due to the union of the clusters. Every pair of clusters is considered at each stage of the algorithm and the pair that minimises the error increase is selected and merged.

The main advantages of these algorithms are the ease of handling of any distance or similarity measure (except Ward's method that specifically employs the Euclidean distance), their flexibility regarding the level of granularity required, and their ability to generate clusters of arbitrary shapes since they do not assume any model for the data set. Their main disadvantage is their high computational cost, typically $O(N^2)$ to $O(N^3)$, which makes them very inefficient when applied to large document collections.

Divisive hierarchical clustering methods are far less common than agglomerative ones and their application to document clustering has been quite limited. However, some algorithms of this type have been recently applied to document collections [75, 76, 77]. The algorithm proposed in [75] uses a division criterion based on the clusters leading principal direction (*i.e.* eigenvector with the largest singular value). Clusters are split in two by the

hyperplane orthogonal to the leading principal direction. The algorithm is somehow vulnerable to undesirable partitions of natural clusters in the data set because it uses just the first principal direction. To overcome this problem, an improved version of the same algorithm was proposed in [76], which takes into account other principal directions.

The division criterion followed in [77] is based on the *topic binder* hypothesis, which states that if a given term is unique to a subset of documents, then those documents are likely to be related to each other. The strongest topic binder term is selected at each stage of the algorithm and a term frequency threshold is used to split documents into two clusters.

In general, divisive hierarchical methods can be implemented using any appropriate cluster splitting criterion. Alternatively, partitional methods can be applied recursively to implement divisive hierarchical clustering [78].

2.5.2 Partitional methods

Unlike hierarchical methods, partitional methods generate a flat clustering structure. These methods are in most cases model-based, *i.e.* they have *a priori* assumption about the model describing the data set. Furthermore, they typically require prior definition of the number of clusters.

Partitional methods usually involve the optimisation of an objective function through an iterative process that only guarantees convergence to a local minimum or maximum. Data elements are successively reallocated to existing clusters at each iteration step until an optimal partition of the data set is achieved. Different clustering criteria and different initialisation methods give rise to different clustering methods [19].

The most popular clustering algorithm in this category is the k-Means [79], which splits the data set into k distinct clusters. It has been applied for document clustering [80], although to a less extent than agglomerative hierarchical clustering methods.

A summary description of the algorithm is given in Figure 2.7. The clustering criterion consists in minimising the error sum of squared distances between the data objects and the centroid of their cluster. The k-Means algorithm usually employs the Euclidean metric for measuring the distance between data objects and cluster centroids, but other metrics [81] and similarity functions can be used [82, 83]. These include the cosine and Jaccard similarity coefficients introduced in section 2.4.1.

```
1. Start with N data vectors, set the number of clusters to k
2. Select k objects as the initial cluster centroids, according
   to some initialisation criterion
3. Loop
4.       Assign each data object to the closest cluster centroid
5.       Compute the new cluster centroids as the weighted
         average of all objects in the corresponding clusters
7. Until (none of the current centroids differs from those in
   the previous iteration)
```

Figure 2.7: Summary description of the *k*-Means partitional clustering algorithm.

Several procedures have been suggested for selecting the initial cluster centroids (step 2 of the algorithm). Traditional examples include:

1. the division of the data set into k clusters at random,
2. the random selection of k objects from the data set (cluster seeds) and assignment of the remaining objects to the nearest cluster [84],
3. the random selection of k objects from the data set (cluster seeds) and sequential assignment of each new object to the nearest cluster followed by immediate recalculation of the cluster centroid [79],
4. and the successive selection of k objects from the data set (cluster seeds), each of them being the most centrally located object to the remaining non-seed objects. After all seeds have been selected the remaining non-seed objects are assigned to the nearest cluster [85].

The comparative study in [86] showed that the first and fourth initialisation approaches lead to the best clustering performances.

The main advantage of the *k*-Means algorithm, and of most partitional clustering methods, is its low computational cost. The algorithm presents linear time complexity with the number of documents, $O(N)$, and hence it is very attractive for clustering large document collections. Another advantage is that, unlike hierarchical methods, it allows relocation of documents and hence, a poor initial partition can be rectified at a later iteration of the algorithm. However, the *k*-Means presents also some disadvantages: the clustering outcome is very sensitive to the initial selection of centroids; the algorithm only finds a local optimum and not a global one; the number of clusters has to be predefined; and it is somewhat sensitive to outliers, since it assumes that clusters are spherical due to the metric used.

Other partitional algorithms that have been applied for document clustering are the *k*-Medoids, Buckshot and Fractionation algorithms. *k*-Medoids methods [85, 87, 88] use documents as cluster representatives (the so-called medoids) rather than cluster centroids. Consequently, these methods are more robust than the *k*-Means in the presence of outliers. However, the selection of the best medoids implies an extensive search over all possible documents, thereby making these algorithms computationally expensive.

The Buckshot and Fractionation [36] are algorithms linearly complex in time that try to overcome the initialisation sensitivity of the *k*-Means method by selecting the initial cluster centroids in a different way. The former applies an agglomerative hierarchical clustering method to a small random sample of documents to find the centroids. Such random sampling procedure may produce different partitions to different calls of the algorithm. The Fractionation algorithm also applies an agglomerative hierarchical method but it proceeds in a different manner. Equal size document groups are defined and the hierarchical method is applied to each of those groups independently. This process is then iterated, by treating each group as an entity, until *k* clusters remain. Fractionation is more accurate than Buckshot in finding the cluster centroids. However, the later is significantly faster than the former.

Agglomerative hierarchical methods are often thought to produce better clusters than partitional methods. However, recent comparisons between agglomerative hierarchical clustering and partitional clustering methods, that addressed specifically the quality of the clusters, challenged such assumption [78, 89]. Experimental results with various document sets showed that the *k*-Means algorithm outperformed the GA method, which was the best among agglomerative hierarchical methods. Furthermore, divisive hierarchical methods implemented through the recursive use of the *k*-Means [89] or some of its variants [78] have even led to slightly better clustering results than the *k*-Means.

2.6 Cluster validity

Clustering algorithms generate clustering structures regardless of the existence or non-existence of an intrinsic structure in the data set. *Clustering tendency* tests serve to assess the clustering properties of the data set rather than the performance of the clustering algorithm. In particular, these tests seek to determine whether the data has a non-random clustering structure that justifies the use of clustering techniques in the first place [90].

Clustering tendency tests have also been used in the context of cluster-based IR to examine the cluster hypothesis [27, 91].

Assuming the data set actually contains natural groups, some clustering methods may perform better than others in finding those groups. *Cluster validity* tests serve to assess whether the clustering output is meaningful or not. These tests enable comparative evaluations of the performance of different clustering methods [22, 90].

Objective evaluation measures of cluster validity can be classified as internal or external measures. Internal validity measures do not assume any external knowledge about the actual clustering structure of the data set and hence, they are independent of the data being clustered. External validity measures, on the other hand, simply rely on prior knowledge on how clusters should be formed to assess the performance of the clustering algorithms and are consequently algorithm independent. The next sub-sections overview these two types of measures.

2.6.1 Internal validity measures

Internal validity measures are in most cases algorithm dependent. For instance, the objective functions that partitional clustering algorithms attempt to optimise are often used as measures of validity of the clustering output. Generically, partitional methods can be evaluated both considering the entire partition or considering each cluster individually [90]. A given partition of the data set can be validated by comparing it to random partitions and checking if it is sufficiently distinct from the random case. Individual clusters are usually evaluated through measures of compactness and separation. The more compact the individual clusters are and the more separated from each other clusters are the better.

The output of the hierarchical clustering methods can also be evaluated based on the validity of individual clusters in the hierarchy, by measuring the clusters compactness and isolation [90]. In the agglomerative case, a cluster is considered good if it forms early in the dendogram and lasts for a relative long period before being merged with other clusters. The evaluation of hierarchical clustering methods can also be based on the validity of the complete cluster hierarchy. Quantitative measures that indicate how well the generated dendogram matches the original similarity relationships between data objects, *i.e.* how well it fits the $N \times N$ proximity matrix, are often applied [20, 90].

2.6.2 External validity measures

External validity measures can be used when the data objects are pre-classified. Such measures are commonly applied in the evaluation of text classification systems. In real document clustering applications, information about the clustering structure is not usually available *a priori*. However, test data sets with reference classes have been assembled for research purposes enabling comparative studies on the performance of different clustering algorithms.

Examples of external measures include the confusion matrix, *F*-measure, average purity, average entropy and mutual information. The confusion matrix is a square matrix where the row labels correspond to known classes, the column labels correspond to discovered clusters and the matrix element in row i and column j contains the number of data objects from reference class i attributed to cluster j. The more the confusion matrix resembles a diagonal matrix the better the performance of the algorithm, unless clusters are known to overlap in which case off-diagonal entries are expected.

The *F*-measure was initially proposed by the IR community [27] and it is a function of precision and recall, which are two popular measures for evaluating the performance of IR systems [27, 41]. Precision gives the fraction of relevant documents out of those retrieved in response to a query and recall represents the fraction of retrieved documents out of the relevant ones. Precision and recall have also been applied for evaluating text classification systems [92, 93, 94]. In this context, precision represents the fraction of elements assigned to a pre-defined class that indeed belong to the class and recall represents the fraction of elements that belong to a pre-defined class that were actually assigned to the class.

The purity of an individual cluster measures the percentage of objects from the best represented pre-defined class in that cluster and the entropy of an individual cluster measures the uncertainty of it representing a single pre-defined class [83, 95]. If all objects in a given cluster have the same class label then the cluster purity is maximised and the cluster entropy is minimised. Minimum purity and maximum entropy occur when the cluster contains an equal number of objects from each pre-defined class. As these two measures are biased to favour small clusters, the use of normalised mutual information has been proposed for an unbiased overall performance evaluation [83].

2.7 Other approaches

This thesis proposes the use of fuzzy clustering techniques to dynamically discover content relationships between learning resources in *e*-Learning systems. The purpose of this knowledge discovery task is to enable exploratory or research oriented interactions with the available *e*-Learning material.

In recent years, research on document clustering has gone beyond pursuing increased efficiency in document retrieval systems. The huge amount of information available on the World Wide Web has encouraged the use of clustering techniques for enhanced browsing and visualisation of Web documents. Both hierarchical and partitional methods have been used in this context [31, 36, 37].

The Suffix Tree Clustering (STC) algorithm was proposed by Zamir and Etzioni [31] as a post-retrieval tool for organising Web search results (*i.e.* search engine snippets). The STC algorithm clusters documents incrementally based on shared phrases. A suffix tree structure is used to organise the initial base clusters, which are represented by phrases. Documents containing a phrase from a given cluster are attributed to that cluster. Then, the algorithm merges base clusters that have high degrees of overlap in their document sets. The experiments reported in [31] have shown that the quality of the clusters produced by the STC algorithm is comparable with to quality of the clusters obtained with the *k*-Means method, when single words are used instead of phrases.

Cutting *et al.* [36] proposed the Scatter/Gather clustering algorithm for browsing large document collections. This method has also been used as a post-retrieval browsing technique [37]. Initially, the Scatter/Gather scatters documents into a small number of clusters that are presented to the user, who then chooses the relevant ones. In the next phase, the algorithm merges the selected clusters for scattering again the sub-collection of documents. The clustering process continues until the clusters are detailed enough from the user's perspective.

Although our motivation for applying clustering techniques is related to the task of enhancing the navigation of *e*-Learning material, the main focus of our research is instead on the discovery and representation of unobvious or unfamiliar knowledge about a domain rather than on facilitating the access to specific information resources through a set of document clusters. We seek to automatically identify associations between related documents and related topics for flexible knowledge exploration.

Specifically, the use of fuzzy clustering methods is proposed in this thesis. These methods allow documents to have membership in multiple clusters thereby acknowledging that knowledge domains may overlap and that *e*-Learning material from a given domain may also be relevant to other domains.

Recently, a number of authors have also explored fuzzy clustering techniques for document clustering applications [96, 97, 98, 99]. In [96, 97], fuzzy clustering has been applied for improving the performance of IR systems. Kraft *et al.* [96] applies the Fuzzy c-Means algorithm and fuzzy logic rules for traditional relevance-feedback in IR (*i.e.* user queries are expanded with new terms). Miyamoto [97] introduces a fuzzy clustering algorithm based on the concept of fuzzy multisets and suggests its application for cluster-based IR. Fuzzy multisets refer to the use of multiple weighted terms for each dimension of the document vector representations. The actual application of the algorithm in IR systems has not been carried out. In [98, 99], fuzzy versions of the hard *k*-Medoids clustering algorithm have been proposed for post-retrieval clustering of search engine snippets, just like the STC algorithm has been proposed.

However, these applications of fuzzy clustering methods are still targeted to the location of specific information resources. To the best of our knowledge, the use of fuzzy clustering for knowledge representation in the context of *e*-Learning, enabling flexible content access, has not been explored previously. Evaluating the performance of fuzzy clustering for knowledge representation in *e*-Learning applications constitutes the main objective of this thesis.

2.8 Summary

In this chapter the basic principles of the document clustering process have been presented. Each of the main steps of this process has been analysed in detail. Initially, we have addressed the representation of documents based on automatic indexing procedures. In particular, we have described traditional vector representations where each dimension consists of an indexing term. We have also reviewed term weighting schemes for reducing or augmenting the importance of terms based on their frequency of occurrence in the document collection. Furthermore, we have presented pre-processing methods that are applied for reducing the dimensionality of the document vector representations. This chapter also presented a review of suitable measures for assessing document relationships

as well as an overview of traditional document clustering methods. Measures of internal and external cluster validity for assessing the performance of clustering methods have also been addressed. Finally, we have presented applications of clustering methods in the Web search context and we have suggested the use of fuzzy clustering as a suitable technique for knowledge representation. In the next chapter, we specifically address the selection of a fuzzy clustering method that suits our purposes.

Chapter 3
Fuzzy clustering for document collections

3.1 Introduction

This thesis proposes the use of fuzzy clustering as a knowledge representation tool for *e*-Learning applications, through the dynamic discovery of content relationships among *e*-Learning material. The previous chapter focused on reviewing the document clustering process and traditional document clustering approaches. In this chapter, the use of fuzzy clustering techniques for document clustering is specifically addressed.

As in every clustering application it is essential to understand the nature of the data set before deciding upon a particular similarity measure and a particular clustering method. Fuzzy clustering techniques are being explored to cluster *e*-Learning material based on the textual content of the metadata documents. Hence, this chapter starts by analysing the typical properties of text document collections. Section 3.2 presents the characteristics of document collections that have been used in our research experiments. In sections 3.3 and 3.4, we discuss the selection of a similarity measure and of a fuzzy clustering method capable of creating overlapping clusters. In particular, we describe the Fuzzy c-Means (FCM) algorithm, which we have chosen for document clustering. In section 3.5, we modify the original FCM algorithm by developing new mathematical expressions so that the proximity between documents can be assessed based on similarity coefficients rather than on the Euclidean distance. In section 3.6, we discuss the characteristics of the

modified algorithm, which has been named Hyper-spherical Fuzzy c-Means (H-FCM). Finally, section 3.7 summarises the main contributions of this chapter.

3.2 Properties of text document collections

In this section the intrinsic properties of text document collections are described. In particular, the document sets that have been used in our clustering experiments are presented: two subsets of the Reuters-21578[2] text categorisation collection, a subset of the Open Directory Project (ODP)[3] metadata and a set of scientific abstracts obtained from the INSPEC database[4]. The reasons for using test collections in the investigation instead of the actual *e*-Learning metadata documents are the following: i) these test collections are pre-classified, therefore providing a benchmark for an objective evaluation of the clustering quality; ii) they are heterogeneous in terms of collection size and type of documents, preventing a biased analysis of the clustering results towards a particular type of document collection; iii) and only a small set of *e*-Learning metadata documents was available at the time of the experiments, for which there was no clustering benchmark. Moreover, the *e*-Learning metadata documents present the same format as the ODP documents.

The Reuters-21578 text collection consists of newswire articles classified into 135 topic categories. Each article is represented as a structured SGML (Standard Generalized Markup Language) [100] document that contains not only the article's text itself but also a set of metadata fields that capture classification information. This classification information includes topics that are associated with the article, if any, whether the article is part of the "training" set or the "test" set, etc. We have selected two subsets of articles which were classified with at least one topic. Only the most frequent topics in the collection were considered: REUTERS1, a subset of articles classified with a single topic - "trade", "acq" or "earn" - and REUTERS2, a subset of articles classified with one or more topics - "crude", "interest", "money-fx", "ship" and "trade".

The ODP is a human-edited directory of the World Wide Web, where Web sites are categorised into a topic hierarchy. Each site is represented by metadata in the RDF (Resource Description Framework) [101] format □ and structured as an XML document.

[2] Reuters-21578 test collection: http://www.daviddlewis.com/resources/testcollections/reuters21578/
[3] Open Directory Project (ODP) : http://dmoz.org/
[4] INSPEC database: http://www.iee.org/publish/inspec/

The metadata includes information about the location of the site in the topic hierarchy and also a short textual description of its contents, instead of the full-text. A subset of the directory was selected, the *Kids and Teens* topic hierarchy. The ODP test collection was created with the short metadata descriptions of Web sites related to the following topics: "game", "lego", "math", "safety" and "sport".

The INSPEC database is a scientific database of abstracts in the fields of physics, electronics and electrical engineering, computers and control, and information technology. We have generated a test set INSPEC by downloading all the abstracts published since 2000 that contained the following keywords: "back-propagation", "fuzzy control" and "pattern clustering".

The document collections have been automatically indexed according to the process described in section 2.3 leading to weighted vector document representations. The bag-of-words approach that has been used in the majority of document clustering applications, has also been adopted in this thesis. Stop words have been eliminated by matching them against the SMART system English stop list [80] and stemming has been performed using Porter's stemming algorithm [102]. Two document representations have been generated, one with the TF encoding scheme and the other with the TF-IDF scheme.

The properties of the document collections are characterised by presenting some basic statistics about each collection and also details about the pre-defined reference classes. The size of each document collection, *i.e.* the total number of documents N and total number of indexing terms k, is given in Table 3.1. The average (avg) document length and average sparsity and the respective standard deviations (stdev) are also given. The length of a document is the total number of words the document contains and its sparsity corresponds to the percentage of indexing terms out of the total number k that do not occur in the document.

Table 3.2 presents the number of reference classes in each document collection and the corresponding topics. The distribution of documents in each class and the overall class overlap are also presented. Such overlap is calculated as the percentage of documents out of the total number of documents N that contain two or more reference topics, *i.e.* that belong to more than one class.

Table 3.1: Characteristics of the test document collections.

Collection	Size		Document length		Document sparsity	
	N	k	avg	stdev	avg	stdev
REUTERS1	1708	15744	73.45	63.97	99.67 %	0.26 %
REUTERS2	1374	11778	102.65	86.37	99.39 %	0.47 %
ODP	556	620	15.14	5.07	97.69 %	0.50 %
INSPEC	7473	11803	93.28	32.79	99.59 %	0.14 %

Table 3.2: Reference classes of the test document collections.

Collection	No. of classes	Topics	No. of docs. per topic	No. of docs. only with this topic	Class overlap
REUTERS1	3	"acq" "earn" "trade"	610 847 251	610 847 251	0.00 %
REUTERS2	5	"crude" "interest" "money-fx" "ship" "trade"	337 440 488 188 293	253 190 206 108 251	26.64 %
ODP	5	"game" "lego" "math" "safety" "sport"	206 187 84 99 59	134 180 59 63 42	14.03 %
INSPEC	3	"back-propagation" "fuzzy control" "pattern clustering"	2271 3401 1920	2174 3302 1879	1.58 %

The heterogeneity of the collections is apparent from the data in Table 3.1 and Table 3.2. The collections have diverse sizes, average document lengths and degrees of class overlapping. A common characteristic to all of them is the very high sparsity of the document vectors. Such sparsity pattern is due to the high dimensionality of the problem space, which is a characteristic of virtually any document collection of a realistic size. Even the smallest test collection, which has just 620 unique terms, has an average sparsity of 97.69%. The understanding of the typical characteristics of document collections is

important for selecting particular metrics or similarity measures (based on which document relationships can be found) as well as for selecting particular clustering algorithms. For example, the selection of a clustering algorithm itself needs to acknowledge the potential large size of the document repository. Time and memory requirements are also important issues to consider. Furthermore, the clustering algorithm may take advantage of the high sparsity of the problem space for reducing memory requirements.

The selection of an appropriate similarity measure and clustering algorithm is the subject of the following sections.

3.3 Selection of a similarity measure

In the previous chapter we have reviewed a number of similarity measures typically used in IR. Such measures have proved to work well in that context and they are natural choices to assess document relationships in the clustering context. Here, we address the selection of a similarity measure for document clustering.

An important issue to consider in the selection process is the interval in which the similarity measure ranges and the meaning of the lower and upper bounds of that interval. The similarity coefficients presented in section 2.4, with the exception of the inner product and overlap, are normalised to range in the unit interval. The use of non-normalised measures has been criticised by van Rijsbergen [27] because they can provide counter intuitive results. For example, the inner product between two identical documents x_A=[1 0 0] and x_B=[1 0 0] equals 1. Yet, the inner product between either of these documents and a very different document x_C=[1 5 2] also equals 1.

This problem is eliminated if document vector representations are normalised to unit length prior to similarity calculation. In such case, all coefficients range in the unit interval and, in particular, the expressions for the inner product (2.4), cosine coefficient (2.5) and Dice coefficient (2.6) are equivalent. In fact, in IR document length normalisation is often carried out so that shorter documents have the same chance to be retrieved as longer ones in response to some query [60]. Likewise, in document clustering applications document length normalisation should also be carried out so that the length of the documents does not impact on the way clusters are formed. For example, two documents with different lengths sharing the same indexing terms should be judged as quite similar to each other and thus clustered together, despite the fact that some terms occur more frequently in the

longer document. In this case, normalising document vectors to unit length prior to the clustering phase should improve the clustering performance. This issue is discussed in section 3.3.1. In particular, the clustering tendency of the document collections with and without normalising document vectors to unit length is analysed. Moreover, the clustering tendency with different similarity coefficients and different term weighting schemes is also analysed.

Another important issue to consider in the selection process of appropriate similarity measures relates to the characteristics of the document collections (*i.e.* high dimensionality and sparsity). This issue is discussed in section 3.3.2.

3.3.1 Clustering tendency

Here we address the problem of selecting an appropriate similarity measure through clustering tendency tests. Clustering tendency tests were originally proposed to examine the cluster hypothesis in the context of cluster-based IR [27, 91]. One of such tests is the Overlap Test (OT) that considers the degree to which documents relevant to the same query are more similar to each other rather than to non-relevant documents [26]. Initially, the pairwise similarities between relevant documents (RR) and the pairwise similarities between relevant and non-relevant (RNR) documents to a given query are calculated. Subsequently, the relative frequency distributions of the two sets of similarity values are compared. If a particular document collection has a clustering tendency then the two distributions should be sufficiently separated. The higher the separation the higher the likelihood of the clustering method producing better clusters.

In [62], the OT has been applied to determine which term weighting scheme and which similarity coefficient would maximise the separation between RR and RNR similarity distributions. The best retrieval effectiveness results have indeed been obtained with the term weighting scheme and similarity coefficient for which the maximum separation has was verified. The separation between RR and RNR distributions has been analysed based on the quantitative measure of overlap proposed in [103], which calculates the number of elements contained in the intersection of the RR and RNR histograms (see Figure 3.1).

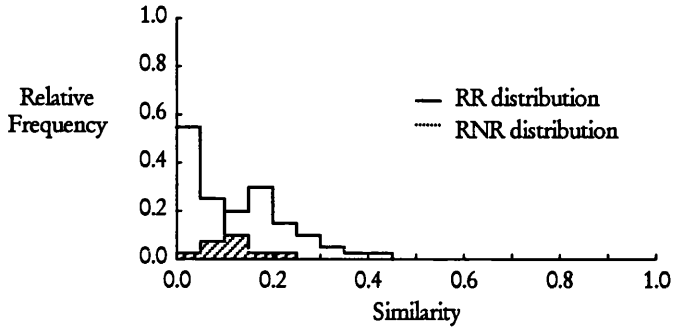


Figure 3.1: Overlap of RR and RNR similarity distributions.

We have adapted the OT to measure the clustering tendency of our pre-classified document collections. Specifically, we determine RR and RNR distributions by calculating the pairwise similarities between documents of the same class, *i.e.* intra-class, and the similarities between each document and documents from different classes, *i.e.* inter-class. In the investigation, we apply the modified OT to the test collections and we consider the effect of document length normalisation, different term weighting schemes (TF and TF-IDF) and different similarity coefficients. The TF-IDF scheme that has been used is defined in equation (2.3). The results of this investigation are presented in Tables 3.3 to 3.6.

The lowest overlap values corresponding to the best separation between intra- and inter-class similarity distributions for each document collection are indicated in bold. We will concentrate on comparisons: i) between non-normalised and normalised document vectors, ii) between TF and TF-IDF term weighting schemes, and iii) between different similarity coefficients (inner product, cosine, Dice, Jaccard and overlap). We note that the OT results with the inner product, cosine and Dice coefficients after normalisation of the document vectors are equal to the OT cosine results prior to normalisation.

Regarding the normalisation of document vectors to unit length hypothesis, from the comparison between Table 3.3 and Table 3.4 and between Table 3.5 and Table 3.6, we observe that length normalisation leads to a slightly better separation between intra- and inter-class distributions. However, such differences in the distributions separation are not statistically significant.

Table 3.3: Results of the OT obtained with non-normalised TF document vectors.

	Similarity Coefficient				
Collection	Inner product	Cosine	Dice	Jaccard	Overlap
REUTERS1	0.96	**0.59**	0.61	0.61	0.67
REUTERS2	0.94	**0.58**	0.59	0.63	0.67
ODP	0.57	**0.21**	**0.21**	0.26	0.39
INSPEC	0.81	**0.45**	0.46	**0.45**	0.53

Table 3.4: Results of the OT obtained with normalised TF document vectors.

	Similarity Coefficient	
Collection	Jaccard	Overlap
REUTERS1	**0.58**	**0.58**
REUTERS2	**0.57**	0.58
ODP	0.23	**0.21**
INSPEC	**0.45**	**0.45**

Table 3.5: Results of the OT obtained with non-normalised TF-IDF document vectors.

	Similarity Coefficient				
Collection	Inner product	Cosine	Dice	Jaccard	Overlap
REUTERS1	0.99	**0.82**	0.84	0.94	0.98
REUTERS2	0.88	**0.78**	0.82	0.91	0.86
ODP	0.94	**0.66**	0.70	0.82	0.85
INSPEC	0.99	**0.77**	0.81	0.93	0.99

Table 3.6: Results of the OT obtained with normalised TF-IDF document vectors.

	Similarity Coefficient	
Collection	Jaccard	Overlap
REUTERS1	0.93	0.86
REUTERS2	0.89	0.90
ODP	0.80	0.74
INSPEC	0.91	0.92

Comparing the data in Table 3.3 and Table 3.4 with the data in Table 3.5 and Table 3.6, we can conclude that the TF scheme leads to a better separation between intra- and inter-class distributions, for all collections and for all similarity coefficients. In fact, the overlap of the two distributions in the TF-IDF case is always considerably high.

Finally, a comparison between the various similarity coefficients reveals that overall the cosine measure produces the lowest overlap between intra- and inter-class distributions. However, we note that for normalised TF vectors the Jaccard and overlap coefficients also present similar performance.

3.3.2 Similarity in high-dimensional low density vector spaces

Here we address the problem of selecting appropriate similarity measures taking into account the typical characteristics of document collections, namely, high dimensionality and sparsity of the document vector representations. We start by analysing the similarity distributions of random vectors for different sparsity levels and for different number of dimensions and then we consider the document vectors from the test collections.

Figure 3.2 presents the histograms of the pairwise similarities between N randomly generated k-dimensional vectors (N=100, k=100) obtained with the cosine, Jaccard and overlap coefficients, for varying sparsity levels. The random vectors have been normalised to unit length before computing the similarities. The plots show that, for all coefficients, the higher the vectors sparsity the lower the vectors similarities. Furthermore, the plots show that the cosine and Jaccard coefficients tend to produce a broad range of similarity values, in particular in the lowest sparsity case. On the other hand, the overlap coefficient produces a narrow range of similarity values (more than 80% of those values are below 0.1), even in the lowest sparsity case.

Figure 3.3 presents the histograms of the pairwise similarities between N randomly generated vectors (N=100) obtained with the cosine, Jaccard and overlap coefficients, for varying number of dimensions k. The random vectors have been normalised to unit length before computing the similarities. The plots show that for a fixed sparsity level (95%) the impact of varying the space dimensionality on the similarity distributions is not very high. The maximum pairwise similarity between +90% of the random vectors is below 0.1, with all coefficients.

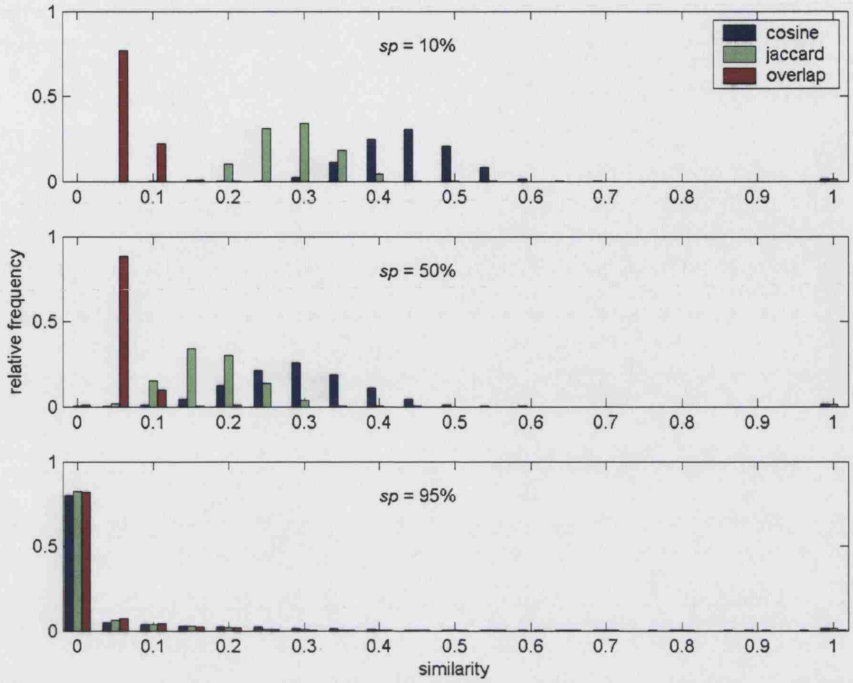


Figure 3.2: Histograms of the pairwise similarities between N=100 random unit-length vectors of k=100 dimensions, for varying sparsity levels.

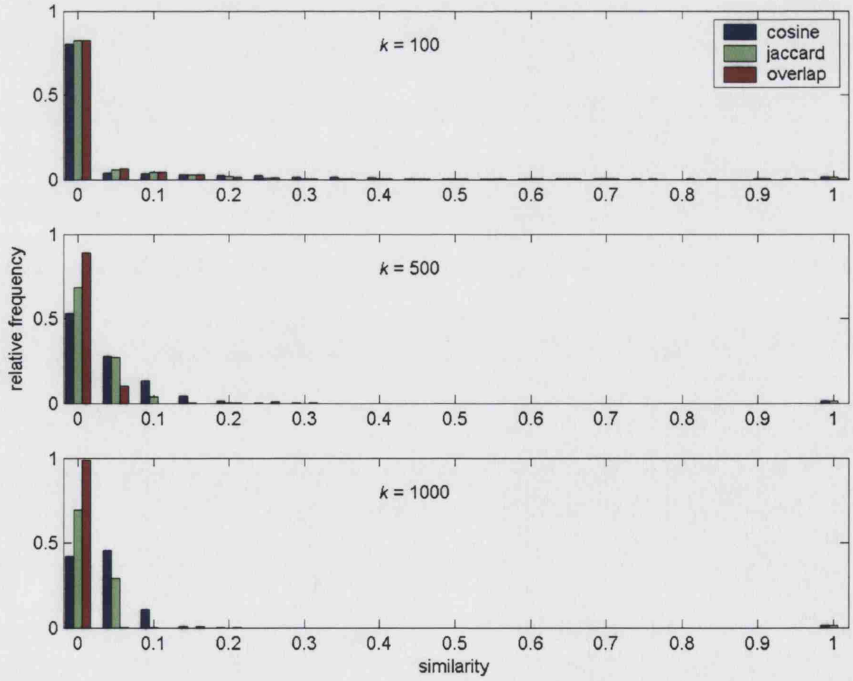


Figure 3.3: Histograms of the pairwise similarities between N=100 random unit-length vectors of k varying dimensions, for a sparsity level of 95%.

Document vectors typically exhibit very high sparsity (in most cases >99%) and are high-dimensional, as verified with our test collections. The previous observations suggest that the pairwise similarity between document vectors will be considerably low, independently of the similarity coefficient used. This is due mainly to the high sparsity rather than the high dimensionality. However, we note that even with such typically low similarity values clustering algorithms should still be able to find document clusters, provided that intra-class similarities are higher than inter-class similarities. The separation between intra- and inter-class similarity distributions was indeed confirmed for our test collections through clustering tendency tests.

The cumulative distribution functions (CDFs) of the intra- and inter-class document similarities obtained with cosine, Jaccard and overlap coefficients are now shown in Figure 3.4, Figure 3.5 and Figure 3.6, respectively. These graphs were obtained for normalised TF document vectors, so they correspond to the data presented in Table 3.4. In all cases, the separation between intra- and inter-class distributions is clear, which confirms once again that clustering is possible. Like with the high-dimensional low density random vectors, the pairwise intra-class document similarities are generally low and they vary with the similarity coefficient used. They also vary with the test collection.

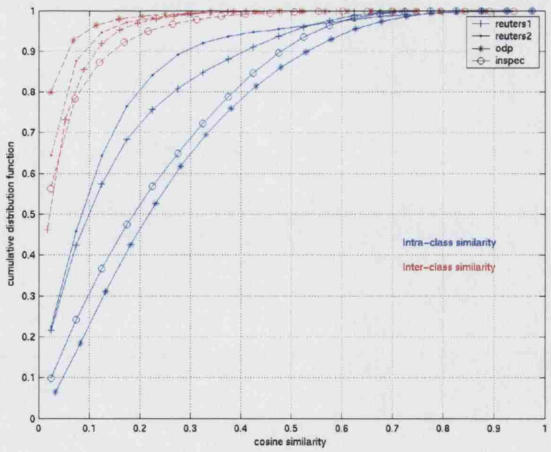


Figure 3.4: Cumulative distribution functions of the intra- and inter-class cosine similarities for the REUTERS1, REUTERS2, ODP and INSPEC document collections.

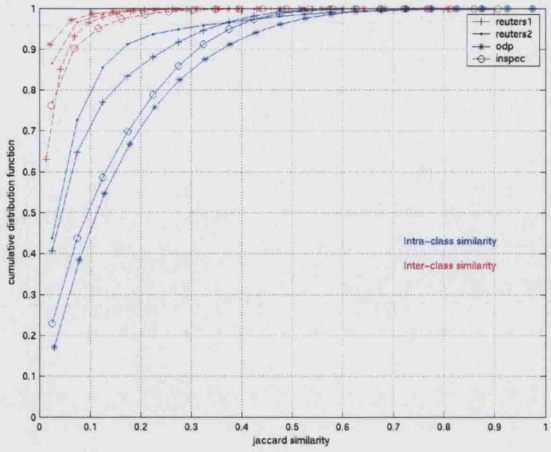


Figure 3.5: Cumulative distribution functions of the intra- and inter-class Jaccard similarities for the REUTERS1, REUTERS2, ODP and INSPEC document collections.

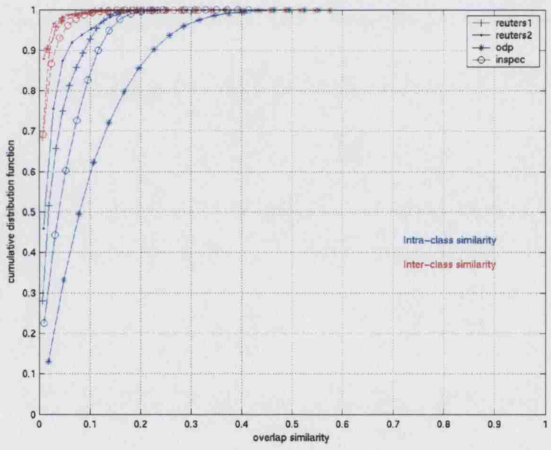


Figure 3.6: Cumulative distribution functions of the intra- and inter-class overlap similarities for the REUTERS1, REUTERS2, ODP and INSPEC document collections.

In general, the cosine coefficient provides the highest similarity values and the overlap coefficient gives the lowest. We can also observe that for the REUTERS1, REUTERS2 and INSPEC collections, the intra-class similarities obtained with the overlap measure are always very low (<0.2). For the ODP collection, which exhibits much lower dimensionality, the overlap measure results in a broader range of intra-class similarities (but still <0.4). These observations suggest that the overlap coefficient may only be suitable for clustering low-dimensional data sets and that overall, the cosine coefficient is the best choice for document clustering.

3.4 Selection of a fuzzy clustering algorithm

In the previous chapter we have reviewed a number of traditional hard document clustering methods. These essentially generate non-overlapping clusters. Here we address the selection of a fuzzy clustering method for discovering document relationships in overlapping knowledge domains.

Fuzzy clustering brings together the theory of fuzzy sets and traditional clustering techniques. The first developments in this area started not long after Zadeh had introduced the concept of fuzzy logic and the theory of fuzzy sets [104]. Research on fuzzy clustering methods is quite vast, but most applications of such methods generally deal with low-dimensional data sets, often of relatively small size, such as data mining [105, 106] and image segmentation [107, 108]. Consequently, the applicability of many of those methods to large high-dimensional data sets is yet to be considered.

Like hard clustering methods, fuzzy methods can be classified as hierarchical or partitional. There are many algorithms of both kinds but fuzzy partitional algorithms are much more common than hierarchical ones. This is mainly due to the extensive use of fuzzy clustering in particular application areas that do not require hierarchical organisation of the data objects (*e.g.* image processing).

The Fuzzy c-Means algorithm (FCM) was one of the first fuzzy clustering methods to emerge and it still is an extremely popular one [18]. The algorithm generalises the hard *k*-Means clustering method [79] by producing a fuzzy partition of the data set instead of a hard partition. Several variations of the original FCM algorithm have been developed to cope with different cluster shapes, sizes and densities [109].

Fuzzy relational clustering is a different family of partitional methods that work with relational data instead of feature vectors. The main relational algorithms are the Fuzzy Non-Metric Model (FNM), the Assignment Prototype (AP), the Relational Fuzzy c-Means (RFCM) and the Non-Euclidean Relational Fuzzy c-Means (NERFCM) algorithm [110]. These methods require $N \times N$ distance calculations which makes them computationally expensive for large N. Other types of relational algorithms use data objects as cluster representatives (the so-called medoids) instead of computing cluster centroids. Examples include the Fuzzy c-Medoids (FCMdd) and the Fuzzy c Trimmed Medoids (FCTMdd) [98]. These methods can still have a computational cost of $O(N^2)$.

Hierarchical fuzzy clustering methods can be classified as agglomerative or divisive, like in the hard clustering case (see Chapter 2). An agglomerative method that produces hard clusters, but that has been classified as fuzzy, since it takes as input fuzzy relations between data objects, is described in [111]. Divisive hierarchical methods usually involve the repeated use of a partitional fuzzy clustering algorithm at each level of the hierarchy. A method that generates a binary fuzzy hierarchy based on the successive refinement of fuzzy partitions is described in [112]. Another algorithm for divisive hierarchical fuzzy clustering, that instead of a binary division attempts to determine the optimum number of clusters at each level of the hierarchy, was proposed in [113]. This algorithm repeats each partitional step for a different number of clusters and determines the optimum number by means of an internal validity test.

The typical large size of document collections which in many applications is ever growing - as it happens with *e*-Learning content repositories - raises the issue of scalability when it comes to the selection of a clustering algorithm. Relational partitional methods suffer from the same disadvantage as agglomerative hierarchical clustering methods, *i.e.* high computational cost. Consequently, they do not scale well to very large data sets. The divisive hierarchical methods mentioned above are also computationally expensive, thus presenting scalability problems. On the other hand, algorithms such as the FCM are quite scalable since they exhibit linear time complexity with the number of documents and the number of dimensions, $O(Nk)$. As mentioned in the previous chapter, the *k*-Means algorithm has long been used for document clustering. Due to its simplicity and linear time complexity, we propose to use its fuzzy counterpart, the FCM algorithm, in our research experiments. This algorithm is described in the next sub-section.

3.4.1 Fuzzy c-Means algorithm (FCM)

The Fuzzy c-Means algorithm can be described as follows [18]. Given a data set $X = \{x_1, x_2, ..., x_N\} \subset \boldsymbol{R}^k$ with N elements each represented by a k-dimensional feature vector x_i, the algorithm runs iteratively to obtain c cluster prototypes or centroids – $V = \{v_1, v_2, ..., v_c\} \subset \boldsymbol{R}^k$ – and a partition matrix – $U=[u_{\alpha i}]$: $(c \times N)$ – where $u_{\alpha i}$ represents the membership of data element x_i in cluster α. The algorithm requires the prior definition of the final number of clusters c $(1<c<N)$, the choice of the fuzzification parameter m $(m>1)$ and the selection of a distance function $\|\cdot\|$, traditionally the Euclidean norm. The

fuzzification parameter controls the fuzziness of the partition matrix. When $m \to 1$ the clusters tend to be hard, *i.e.* $u_{\alpha i} \to 1$ or $u_{\alpha i} \to 0$, and when $m \to \infty$ maximum fuzziness is approached, *i.e.* $u_{\alpha i} \to 1/c$.

Both the cluster centroids and the partition matrix are computed to minimise the weighted within group sum of squared error objective function J_m:

$$J_m(U,V) = \sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}{}^{m} \left\| x_i - v_\alpha \right\|^2 = \sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}{}^{m} d_{i\alpha}{}^{2} \tag{3.9}$$

The FCM algorithm starts with a random initialisation of the partition matrix subject to the following constraints:

1. $u_{\alpha i} \in [0,1], \ \forall_{\alpha \in \{1,\ldots,c\}} \ \forall_{i \in \{1,\ldots,N\}}$ (3.10)

2. $\sum_{\alpha=1}^{c} u_{\alpha i} = 1, \ \forall_{i \in \{1,\ldots,N\}}$ (3.11)

3. $0 < \sum_{i=1}^{N} u_{\alpha i} < N, \ \forall_{\alpha \in \{1,\ldots,c\}}$ (3.12)

At each iteration t the memberships and the cluster centroids are updated according to equations (3.13) and (3.14), respectively.

$$u_{\alpha i} = \left[\sum_{\beta=1}^{c} \left(\frac{\left\| x_i - v_\alpha \right\|^2}{\left\| x_i - v_\beta \right\|^2} \right)^{\frac{1}{(m-1)}} \right]^{-1} = \left[\sum_{\beta=1}^{c} \left(\frac{d_{i\alpha}{}^{2}}{d_{i\beta}{}^{2}} \right)^{\frac{1}{(m-1)}} \right]^{-1} \tag{3.13}$$

$$v_\alpha = \frac{\sum_{i=1}^{N} u_{\alpha i}{}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{\alpha i}{}^{m}} \tag{3.14}$$

The algorithm ends when U^t differs from the previous U^{t-1} by a small quantity ε, or when the maximum number of iterations is reached.

3.4.2 Considerations about the Euclidean distance

The distance function applied in the FCM algorithm is usually the Euclidean distance. Although this metric has been used in some document clustering applications [96, 114, 115, 116], it is not the most suitable document association measure. In fact, the use of the Euclidean distance has been criticised by Willet [20] and its inappropriateness for high-dimensional text clustering has been verified by Strehl *et al.* [83]. The problem with this metric is that the non-occurrence of the same terms in both documents is handled in a similar way as the co-occurrence of terms.

To illustrate the weakness of the Euclidean distance we give the following example. Let us consider two documents x_A and x_B indexed with a set of k terms (see Figure 3.7). Let us assume that most terms, say $k'<k$, appear neither in x_A nor in x_B. This is a reasonable assumption considering previous observations for real document collections (see section 3.3). Let us also assume that x_A and x_B have no terms in common.

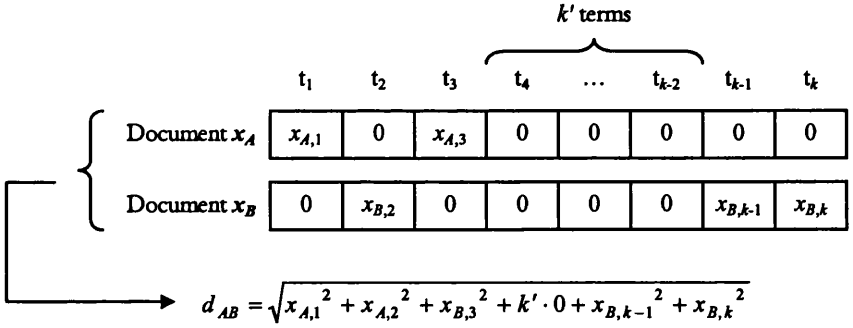


Figure 3.7: Euclidean distance between two totally dissimilar documents.

Since the two document vectors agree in k' dimensions in which they both have zero term weights, their Euclidean distance may be relatively small when in fact x_A and x_B are totally dissimilar. Therefore, in situations like this the Euclidean distance fails to produce an acceptable measure of the documents relationship.

In this context, the similarity measures discussed in section 3.3, are much more appropriate for assessing the proximity of documents vectors. In the above example, these measures would correctly indicate zero similarity between the two documents.

It is appropriate now to consider the properties of the Euclidean distance when used in conjunction with normalised vectors rather than non-normalised vectors. This metric is an inner product induced norm given by the following equation:

$$d_{AB} = \|x_A - x_B\| = \langle x_A - x_B, x_A - x_B \rangle^{1/2} = \left[\sum_{j=1}^{k} (x_{Aj} - x_{Bj})^2 \right]^{1/2} \tag{3.15}$$

When the vectors x_A and x_B are normalised to unit length it follows that:

$$d_{AB}{}^2 = \langle x_A - x_B, x_A - x_B \rangle = \langle x_A, x_A \rangle - 2\langle x_A, x_B \rangle + \langle x_B, x_B \rangle = 2 - 2\langle x_A, x_B \rangle \tag{3.16}$$

This equation reveals that the squared Euclidean distance between two unit length vectors is directly related to their inner product. Note that the similarity coefficients presented in Table 2.1 are also functions of the inner product. If we define a dissimilarity function through a simple transformation of the similarity function,

$$D(x_A, x_B) = 1 - S(x_A, x_B) \tag{3.17}$$

which ranges in the unit interval given that,

$$0 \le S(x_A, x_B) \le 1, \forall_{A,B} \tag{3.18}$$

and if we consider the best performing similarity coefficient, *i.e.* the cosine coefficient, which reduces to the inner product when x_A and x_B are normalised, then the relation between Euclidean distance and the dissimilarity function becomes straightforward:

$$D(x_A, x_B) = 1 - \frac{\sum_{j=1}^{k} x_{Aj} \cdot x_{Bj}}{\left[\sum_{j=1}^{k} x_{Aj}{}^2 \cdot \sum_{j=1}^{k} x_{Bj}{}^2 \right]^{1/2}} = 1 - \langle x_A, x_B \rangle = \frac{d_{AB}{}^2}{2} \tag{3.19}$$

This observation might suggest that applying the FCM algorithm to a document collection, having the documents normalised beforehand, will produce equivalent results either using the Euclidean norm or the cosine dissimilarity function. However, such assumption is incorrect because the cluster centroid vectors are not themselves normalised in the original algorithm. Consequently, the squared Euclidean distance between document vectors and cluster centroids will not follow equation (3.19).

3.5 Hyper-spherical Fuzzy c-Means (H-FCM)

In the previous section we have shown that similarity coefficients are much more appropriate than the Euclidean distance for assessing the proximity between document vectors. We have also mentioned that documents sharing the same indexing terms should be deemed related to each other, regardless of their length. Thus, document vectors should be normalised to unit length. In order to cluster normalised document vectors with the FCM method but employing (dis)similarity coefficients rather than the Euclidean distance, novel mathematical expressions are developed in this section.

We have modified the objective function in equation (3.9) by replacing the squared norm with the function defined in equation (3.17):

$$J_m(U,V) = \sum_{i=1}^{N}\sum_{\alpha=1}^{c} {u_{\alpha i}}^m D_{i\alpha} = \sum_{i=1}^{N}\sum_{\alpha=1}^{c} {u_{\alpha i}}^m (1 - S_{i\alpha}) \tag{3.20}$$

Following the same reasoning behind the normalisation of the document vectors, the cluster centroids $V = \{v_1, v_2, ..., v_c\} \subset R^k$, which in fact represent virtual documents, should also be normalised to unit length. Consequently, an extra constraint for the optimisation of J_m has to be introduced:

$$\langle v_\alpha, v_\alpha \rangle = \sum_{l=1}^{k} {v_{\alpha l}}^2 = 1, \forall_\alpha \tag{3.21}$$

Using the method of the Lagrange multipliers (see Appendix A) it is possible to minimise (3.20) subject to constraints in a straightforward manner, as this method converts constrained optimisation problems into unconstrained ones. Besides constraint (3.21), constraints (3.10), (3.11) and (3.12) still have to be considered. Note that the optimisation is made with respect to U and V separately. Minimisation of the function (3.20) with respect to $u_{\alpha i}$ (v_α fixed) leads to a result similar to (3.13) because the new constraint (3.21) is not a function of $u_{\alpha i}$. The only difference is the replacement of $d_{i\alpha}^2$ and $d_{i\beta}^2$ by $D_{i\alpha}$ and $D_{i\beta}$. The expression for $u_{\alpha i}$ is now:

$$u_{\alpha i} = \left[\sum_{\beta=1}^{c} \left(\frac{D_{i\alpha}}{D_{i\beta}} \right)^{\frac{1}{(m-1)}} \right]^{-1} \tag{3.22}$$

To minimise (3.20) with respect to v_α ($u_{\alpha i}$ fixed) subject to constraint (3.21), the Lagrangian function is defined as:

$$L_m(v_\alpha,\lambda_\alpha) = J_m(U,v_\alpha) + \lambda_\alpha \cdot \left[\langle v_\alpha, v_\alpha \rangle - 1\right]$$
$$= \sum_{i=1}^{N}\sum_{\alpha=1}^{c} {u_{\alpha i}}^m D_{i\alpha} + \lambda_\alpha \left(\sum_{l=1}^{k} {v_{\alpha l}}^2 - 1\right) \tag{3.23}$$

where λ_α is the Lagrange multiplier. Note that constraints (3.10) to (3.12) are not considered in this Lagrangian function because they are not a function of v_α.

To convert this optimisation problem into an unconstrained problem the derivative of the Lagrangian function is taken,

$$\frac{\partial L_m(v_\alpha,\lambda_\alpha)}{\partial v_\alpha} = \frac{\partial J_m(U,v_\alpha)}{\partial v_\alpha} + \lambda_\alpha \cdot \frac{\partial\left[\langle v_\alpha, v_\alpha \rangle - 1\right]}{\partial v_\alpha} = 0 \tag{3.24}$$

In the next sub-sections, we develop this equation to derive new expressions for the cluster centroid vectors for three similarity measures: cosine, Jaccard and overlap coefficients.

3.5.1 Case I - Cosine coefficient

The cosine dissimilarity function for normalised vectors is given by the expression in equation (3.25). Substituting $D_{i\alpha}$ in the objective function (3.20) and also in the cluster membership equation (3.22) yields the modification of the FCM algorithm for employing the cosine similarity coefficient[5].

$$D_{i\alpha} = 1 - \frac{\sum_{j=1}^{k} x_{ij} \cdot v_{\alpha j}}{\left[\sum_{j=1}^{k} {x_{ij}}^2 \cdot \sum_{j=1}^{k} {v_{\alpha j}}^2\right]^{1/2}} = 1 - \sum_{j=1}^{k} x_{ij} \cdot v_{\alpha j} \tag{3.25}$$

The new update expression for the cluster centroids is derived by substituting $D_{i\alpha}$ in the Lagrangian function (3.23),

[5] Since the publication of our paper [117] we came across a paper by Klawonn and Keller [118] which employs the inner product in the FCM algorithm.

$$L_m(v_\alpha,\lambda_\alpha)=\sum_{i=1}^{N}u_{\alpha i}{}^m(1-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j})+\lambda_\alpha(\sum_{l=1}^{k}v_{\alpha l}{}^2-1) \tag{3.26}$$

Taking the derivative according to (3.24) and equating to zero it follows that,

$$\frac{\partial L_m(v_\alpha,\lambda_\alpha)}{\partial v_\alpha}=-\sum_{i=1}^{N}u_{\alpha i}{}^m x_i+2\lambda_\alpha v_\alpha=0 \tag{3.27}$$

which is equivalent to,

$$v_\alpha=\frac{1}{2\lambda_\alpha}\cdot\sum_{i=1}^{N}u_{\alpha i}{}^m x_i \tag{3.28}$$

Applying constraint (3.21) leads to,

$$\sum_{l=1}^{k}v_{\alpha l}{}^2=\left(\frac{1}{2\lambda_\alpha}\right)^2\cdot\sum_{l=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}{}^m x_{il}\right)^2=1\Leftrightarrow\frac{1}{2\lambda_\alpha}=\left[\sum_{l=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}{}^m x_{il}\right)^2\right]^{-1/2} \tag{3.29}$$

Finally, replacing $\frac{1}{2\lambda_\alpha}$ in (3.28) results in:

$$v_\alpha=\frac{\sum_{i=1}^{N}u_{\alpha i}{}^m x_i}{\left[\sum_{l=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}{}^m x_{il}\right)^2\right]^{1/2}} \tag{3.30}$$

3.5.2 Case II - Jaccard coefficient

The Jaccard dissimilarity function for normalised vectors is given by the expression in equation (3.31). Substituting $D_{i\alpha}$ in the objective function (3.20) and also in the cluster membership equation (3.22) yields the modification of the FCM algorithm for employing the Jaccard coefficient.

$$D_{i\alpha}=1-\frac{\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}{\sum_{j=1}^{k}x_{ij}{}^2+\sum_{j=1}^{k}v_{\alpha j}{}^2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}=1-\frac{\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}{2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}} \tag{3.31}$$

The new update expression for the cluster centroids is derived by substituting $D_{i\alpha}$ in the Lagrangian function (3.23),

$$L_m(\boldsymbol{v}_\alpha,\lambda_\alpha)=\sum_{i=1}^{N}{u_{\alpha i}}^m\cdot\left(1-\frac{\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}{2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}\right)+\lambda_\alpha\left(\sum_{l=1}^{k}{v_{\alpha l}}^2-1\right) \tag{3.32}$$

Taking the derivative according to (3.24) and equating to zero it follows that,

$$\begin{aligned}\frac{\partial L_m(\boldsymbol{v}_\alpha,\lambda_\alpha)}{\partial \boldsymbol{v}_\alpha}&=-\sum_{i=1}^{N}{u_{\alpha i}}^m\cdot\frac{\boldsymbol{x}_i\cdot\left(2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right)+\left(\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right)\cdot\boldsymbol{x}_i}{\left(2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right)^2}+2\lambda_\alpha\boldsymbol{v}_\alpha=\\&=-\sum_{i=1}^{N}{u_{\alpha i}}^m\cdot\frac{2\boldsymbol{x}_i}{\left(2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right)^2}+2\lambda_\alpha\boldsymbol{v}_\alpha=\\&=0\end{aligned} \tag{3.33}$$

which is equivalent to,

$$\boldsymbol{v}_\alpha=\frac{1}{2\lambda_\alpha}\cdot\sum_{i=1}^{N}{u_{\alpha i}}^m\cdot\underbrace{\frac{2\boldsymbol{x}_i}{\left(2-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}\right)^2}}_{F_i} \tag{3.34}$$

Given that the FCM is an iterative algorithm and that the cluster centroids do not change significantly from one iteration to the next, the solution to this equation can be simplified by computing F_i using the cluster centroid given by the previous iteration.

Applying constraint (3.21) leads to,

$$\sum_{l=1}^{k}{v_{\alpha l}}^2=\left(\frac{1}{2\lambda_\alpha}\right)^2\cdot\sum_{l=1}^{k}\left(\sum_{i=1}^{N}{u_{\alpha i}}^m\cdot F_{il}\right)^2=1\Leftrightarrow\frac{1}{2\lambda_\alpha}=\left[\sum_{l=1}^{k}\left(\sum_{i=1}^{N}{u_{\alpha i}}^m\cdot F_{il}\right)^2\right]^{-1/2} \tag{3.35}$$

where,

$$F_{il} = \frac{2x_{il}}{\left(2 - \sum_{j=1}^{k} x_{ij} \cdot v_{\alpha j}\right)^2} \tag{3.36}$$

Finally, replacing $\frac{1}{2\lambda_\alpha}$ in (3.34) results in,

$$v_\alpha = \frac{\sum_{i=1}^{N} {u_{\alpha i}}^m \cdot F_i}{\left[\sum_{l=1}^{k}\left(\sum_{i=1}^{N} {u_{\alpha i}}^m \cdot F_{il}\right)^2\right]^{1/2}} \tag{3.37}$$

3.5.3 Case III - Overlap coefficient

The overlap dissimilarity function for normalised vectors is given by the expression in equation (3.38). Substituting $D_{i\alpha}$ in the objective function (3.20) and also in the cluster membership equation (3.22) yields the modification of the FCM algorithm for employing the overlap coefficient.

$$D_{i\alpha} = 1 - \frac{\sum_{j=1}^{k} x_{ij} \cdot v_{\alpha j}}{\min\left(\sum_{j=1}^{k} x_{ij}, \sum_{j=1}^{k} v_{\alpha j}\right)} = \begin{cases} 1 - \dfrac{\sum_{j=1}^{k} x_{ij} \cdot v_{\alpha j}}{\sum_{j=1}^{k} x_{ij}} & , \sum_{j=1}^{k} x_{ij} \le \sum_{j=1}^{k} v_{\alpha j} \\ 1 - \dfrac{\sum_{j=1}^{k} x_{ij} \cdot v_{\alpha j}}{\sum_{j=1}^{k} v_{\alpha j}} & , \sum_{j=1}^{k} x_{ij} > \sum_{j=1}^{k} v_{\alpha j} \end{cases} \tag{3.38}$$

The new update expression for the cluster centroids is derived by substituting $D_{i\alpha}$ in the Lagrangian function (3.23),

$$L_m(v_\alpha, \lambda_\alpha) = \sum_{\substack{r=1 \\ \sum_{j=1}^{k} x_{rj} \le \sum_{j=1}^{k} v_{\alpha j}}}^{N} {u_{\alpha r}}^m \cdot \left(1 - \frac{\sum_{j=1}^{k} x_{rj} \cdot v_{\alpha j}}{\sum_{j=1}^{k} x_{rj}}\right) + \sum_{\substack{s=1 \\ \sum_{j=1}^{k} x_{sj} > \sum_{j=1}^{k} v_{\alpha j}}}^{N} {u_{\alpha s}}^m \cdot \left(1 - \frac{\sum_{j=1}^{k} x_{sj} \cdot v_{\alpha j}}{\sum_{j=1}^{k} v_{\alpha j}}\right) +$$

$$+ \lambda_\alpha \left(\sum_{l=1}^{k} {v_{\alpha l}}^2 - 1\right) \tag{3.39}$$

Taking the derivative according to (3.24) it follows that,

$$\frac{\partial L_m(v_\alpha,\lambda_\alpha)}{\partial v_\alpha} = \underbrace{-\sum_{\substack{r=1 \\ \sum_{j=1}^{k} x_{rj} \le \sum_{j=1}^{k} v_{\alpha j}}}^{N} {u_{\alpha r}}^m \cdot \frac{x_r}{\sum_{j=1}^{k} x_{rj}}}_{F_r} \underbrace{-\sum_{\substack{s=1 \\ \sum_{j=1}^{k} x_{sj} > \sum_{j=1}^{k} v_{\alpha j}}}^{N} {u_{\alpha s}}^m \cdot \frac{x_s \sum_{j=1}^{k} v_{\alpha j} - \sum_{j=1}^{k} x_{sj} v_{\alpha j}}{\left(\sum_{j=1}^{k} v_{\alpha j}\right)^2}}_{F_s} + 2\lambda_\alpha v_\alpha = 0 \tag{3.40}$$

which is equivalent to,

$$v_\alpha = \frac{1}{2\lambda_\alpha} \cdot (F_r + F_s) \tag{3.41}$$

Like in the Jaccard case, the solution to this equation can be simplified by computing F_r and F_s using the cluster centroid given by the previous iteration.

Applying constraint (3.21) leads to,

$$\sum_{l=1}^{k} {v_{\alpha j}}^2 = \left(\frac{1}{2\lambda_\alpha}\right)^2 \cdot \sum_{l=1}^{k} (F_{rl} + F_{sl})^2 = 1 \Leftrightarrow \frac{1}{2\lambda_\alpha} = \left[\sum_{l=1}^{k} (F_{rl} + F_{sl})^2\right]^{-1/2} \tag{3.42}$$

where,

$$F_{rl} = \sum_{\substack{r=1 \\ \sum_{j=1}^{k} x_{rj} \le \sum_{j=1}^{k} v_{\alpha j}}}^{N} {u_{\alpha r}}^m \cdot \frac{x_{rl}}{\sum_{j=1}^{k} x_{rj}} \tag{3.43}$$

and

$$F_{sl} = \sum_{\substack{s=1 \\ \sum_{j=1}^{k} x_{sj} > \sum_{j=1}^{k} v_{\alpha j}}}^{N} {u_{\alpha s}}^m \cdot \frac{x_{sl} \sum_{j=1}^{k} v_{\alpha j} - \sum_{j=1}^{k} x_{sj} v_{\alpha j}}{\left(\sum_{j=1}^{k} v_{\alpha j}\right)^2} \tag{3.44}$$

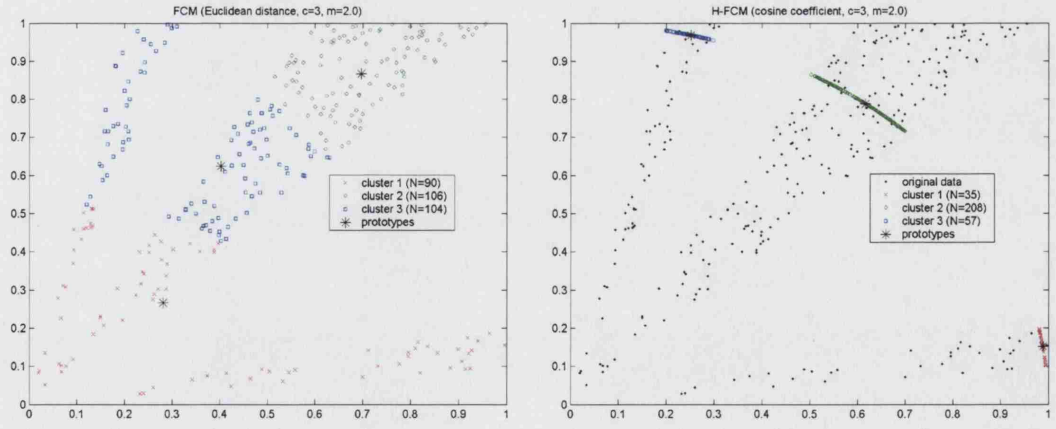
Finally, replacing $\frac{1}{2\lambda_\alpha}$ in (3.41) results in:

$$v_a = \frac{F_r + F_s}{\left[\sum_{l=1}^{k}(F_{rl} + F_{sl})^2\right]^{1/2}} \tag{3.45}$$

3.5.4 Summary of the H-FCM algorithm

We have named the modified algorithm as Hyper-spherical Fuzzy c-Means (H-FCM), since both data vectors and cluster centroids lie in a k-dimensional hyper-sphere of unit radius. To illustrate this, the H-FCM clustering of an example bi-dimensional data set using the cosine similarity measure is given in Figure 3.8.

The data set can be partitioned into three clusters of highly similar data elements (*i.e.* any pair of elements in the cluster has a high cosine similarity). The FCM algorithm, employing the Euclidean distance, was also run for the same data. The H-FCM graph shows both cluster centroids and normalised data elements located in a circle of unit radius. The H-FCM algorithm distributes the data elements among the clusters following a criterion of minimum cosine dissimilarity to the cluster centroids, whereas the FCM bases such distribution on the minimum Euclidean distance to the cluster centroids. Hence, we conclude that H-FCM places similar elements in the same cluster whereas FCM is susceptible to place elements that are quite dissimilar in the same cluster.



a) b)

Figure 3.8: Clustering of points in a bi-dimensional space using: a) FCM and b) H-FCM with cosine coefficient.

The centroid expression in the FCM case is related to the H-FCM centroid expression in the cosine case. In both cases, centroids are computed as the mean of the elements belonging to the respective clusters, weighted by the elements degree of membership to the cluster. However, in the H-FCM case the centroids are normalised to unit-length whereas in the FCM case they are not.

To summarise, the H-FCM algorithm runs in a similar manner to the original FCM, differing only on the update expression for v_α. The H-FCM pseudo-code description is given in Figure 3.9.

1. **Start** with a $N \times k$ data matrix X, with row vectors normalised to unit length and **select** the final number of clusters $c>1$, fuzzification parameter $m>1$, termination criterion $\varepsilon>0$ and **set** the maximum number of iterations t_max
2. **Initialise** $U^{(0)}$ randomly considering (3.10), (3.11) and (3.12)
3. For cases II (Jaccard) and III (overlap), **initialise** $V^{(0)}$ randomly considering (3.21)
4. **Set** iteration t=1
5. **While** (t <= t_max)
6. **Compute** the cluster centroids $V^{(t)}$ according to
   Case I: (3.30) using $U^{(t-1)}$
   Case II: (3.37) using $U^{(t-1)}$ and $V^{(t-1)}$
   Case III: (3.45) using $U^{(t-1)}$ and $V^{(t-1)}$
7. **Compute** $U^{(t)}$ according to (3.22) using $V^{(t)}$
8. **If** ($\|U^{(t)} - U^{(t-1)}\| < \varepsilon$) **stop while**
9. **Set** t=t+1
10. **Endwhile**
11. **Return** partition matrix U and cluster centroids V

Figure 3.9: Summary description of the H-FCM algorithm.

3.6 Considerations about the H-FCM algorithm

Important issues to consider regarding the properties of clustering methods, include scalability to large data sets, ability to work with high-dimensional data, computational complexity, handling of outliers, dependence on user defined parameters, sensitiveness to initialisation conditions, ability to find clusters of different sizes and shapes and ability to generate overlapping clusters. Although there are many algorithms available, there is no such thing as the perfect algorithm as they all have some weaknesses. Depending on the application specific requirements, a trade-off is usually established for the selection of the clustering algorithm.

It is known that the FCM presents the same limitations as the *k*-Means algorithm: it assumes the clusters shape is spherical with respect to the metric used and it is sensitive to initialisation conditions. Previously, we have justified the selection of the FCM algorithm based on its low time complexity, for scaling well with the number of documents and the number of dimensions, for its ability to generate fuzzy membership functions and for being the fuzzy counterpart of a well-established document clustering method (*i.e.* the *k*-Means). The H-FCM algorithm was derived by modifying the FCM to employ similarity coefficients that perform well with high-dimensional document vectors. However, there are two issues which remain to be considered: handling of outliers and dependence on user defined parameters.

3.6.1 Handling of outliers

Outliers are data elements that lie outside the region where most elements of the data set are located. Due to the probabilistic constraint imposed by the FCM on the cluster memberships (see equation (3.11)), the membership of outliers in the *c* clusters is never as small as desired. Therefore, the algorithm is not very robust in the presence of such elements. To overcome this problem possibilistic clustering algorithms have been proposed [119, 120, 121]. In such methods the cluster memberships are not constrained to add-up to one and consequently, each cluster is dissociated from the others. Conceptually, the membership values in possibilistic clustering represent the degree of compatibility between data objects and cluster centroids rather than degrees of sharing.

Although possibilistic methods are better than the FCM at handling outliers they present some drawbacks in the context our application. Firstly, these methods are extremely sensitive to the initial partition, thereby requiring elaborate initialisation procedures. Secondly, the cluster centroids are automatically attracted to dense regions. As shown in section 3.3.2, the pairwise similarities between sparse high-dimensional data points are typically low. This means that the document space will not present clear dense regions and hence, possibilistic algorithms may fail. Moreover, it has been observed that some possibilistic methods tend to produce coincident clusters even when dense regions exist [122]. However, we note that outliers in such high-dimensional space (*e.g.* documents that are not clearly related to any other documents) are not expected to have great impact on the final location of the cluster centroids. This is because their similarity to all centroids is generally extremely low. In such case, the H-FCM algorithm tends to assign outliers to all

clusters with equal membership, *i.e.* $u_{i\alpha}=1/c$. Thus, outliers can be handled by setting a membership threshold to isolate documents falling in this category.

3.6.2 Dependence on user defined parameters

Regarding user defined parameters, the H-FCM requires setting the final number of clusters c and the fuzzification parameter m. As mentioned before, m simply controls the fuzziness of the partition matrix and its choice is not usually critical. Regarding the choice of c, in most applications the optimum number of clusters is not known *a priori*. This has motivated research in clustering methods capable of automatically determining the optimum number of clusters [123, 124].

A typical approach is to run the clustering algorithm for a range of values of c and then apply validity measures based on the compactness and density of the clusters, to determine which c leads to the best partition. The clustering method described in [123] follows this approach. The drawback of this method is its computational cost because the algorithm needs to be repeated for each c.

In [124], a different approach based on the maximum entropy principle is followed. The proposed algorithm finds cluster centroids by maximising the entropy of clusters and assuming that the centroids are not biased towards any of the data objects. A resolution parameter β ranging in the unit interval has to be defined, which controls the number of obtained clusters. Like with the previous approach, the algorithm has to be run repeatedly for a range of β values, which makes it computationally unattractive.

The algorithms described above assume that valid clusters are compact, dense and well separated from each other. In low-dimensional spaces it is acceptable to make such assumption. The performance of the algorithms was indeed assessed with low-dimensional data sets. However, in high-dimensional spaces dense regions are not expected due to the typical low similarity patterns. This motivates an investigation of the H-FCM clustering structure in the high-dimensional case.

In section 3.3.2, we have analysed the similarity distributions of sparse high-dimensional vectors as well as the intra- and inter-class similarity distributions for our test document collections and we have verified that the pairwise similarities are generally low. Now, we analyse the clustering structure produced by the H-FCM algorithm when applied to the four document collections using the cosine coefficient as similarity measure.

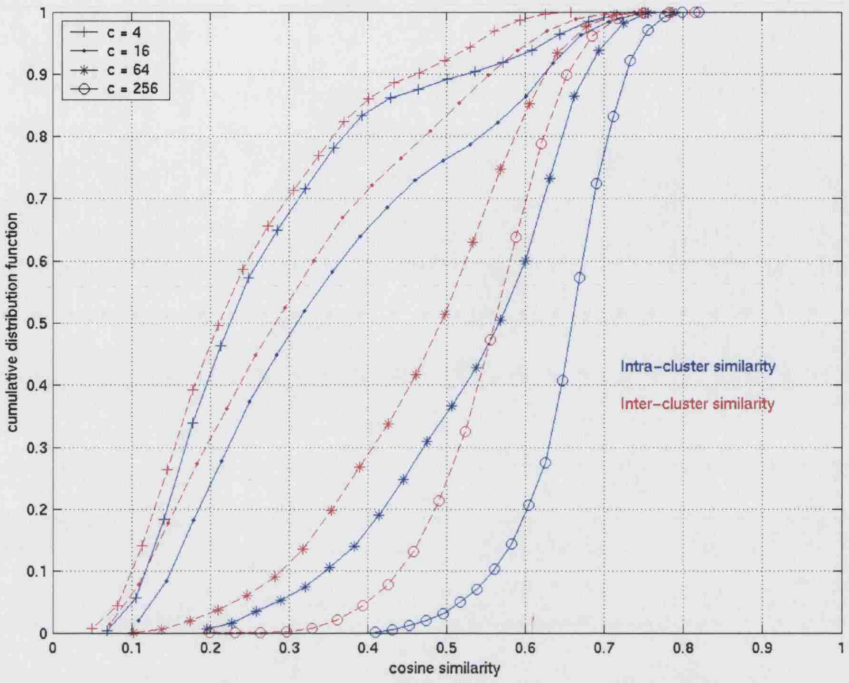


Figure 3.10: Cumulative distribution functions of the intra- and inter-cluster cosine similarity for the REUTERS1 document collection, obtained with H-FCM for $c = 4$, 16, 64 and 256.

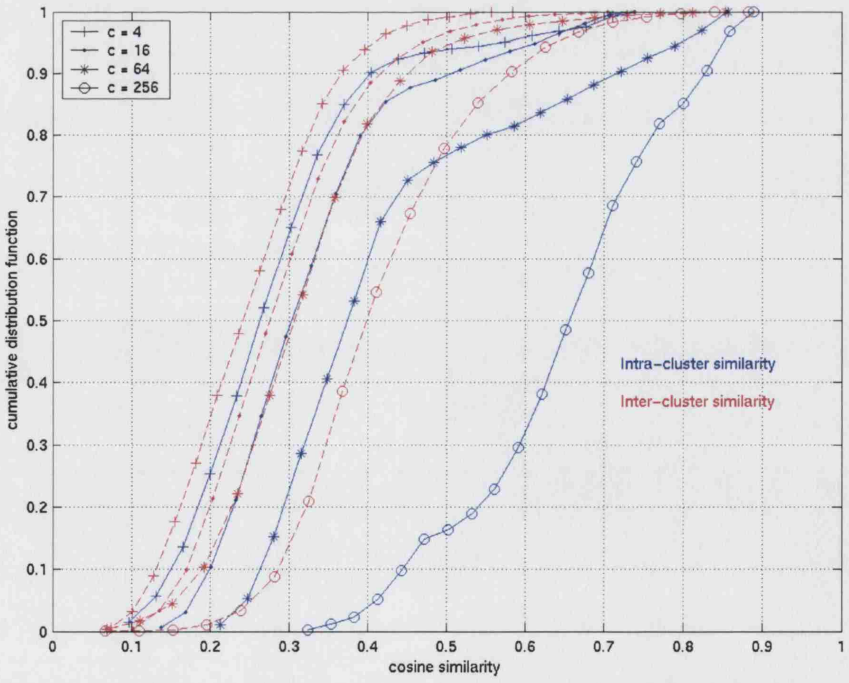


Figure 3.11: Cumulative distribution functions of the intra- and inter-cluster cosine similarity for the REUTERS2 document collection, obtained with H-FCM for $c = 4$, 16, 64 and 256.

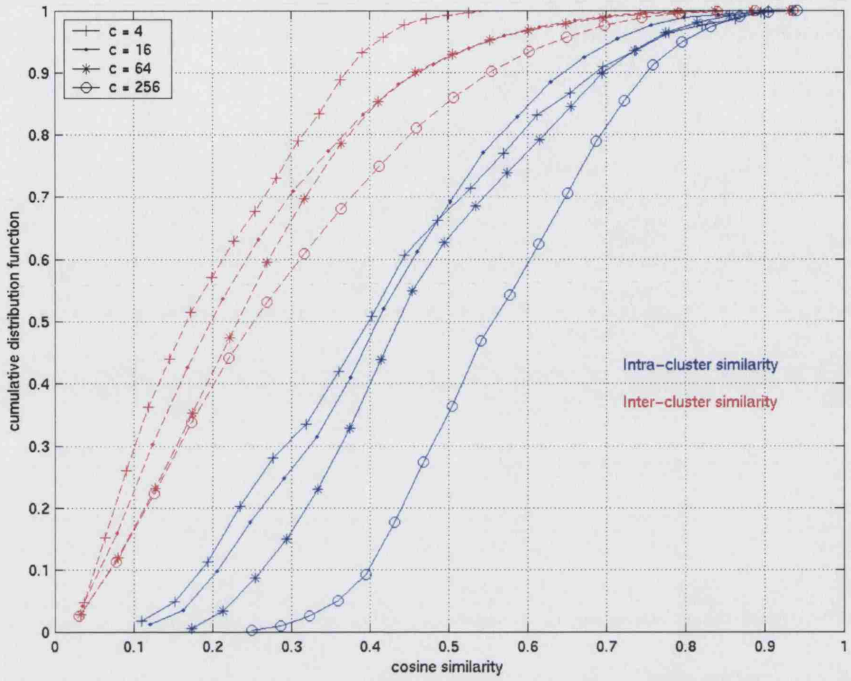


Figure 3.12: Cumulative distribution functions of the intra- and inter-cluster cosine similarity for the ODP document collection, obtained with H-FCM for c = 4, 16, 64 and 256.

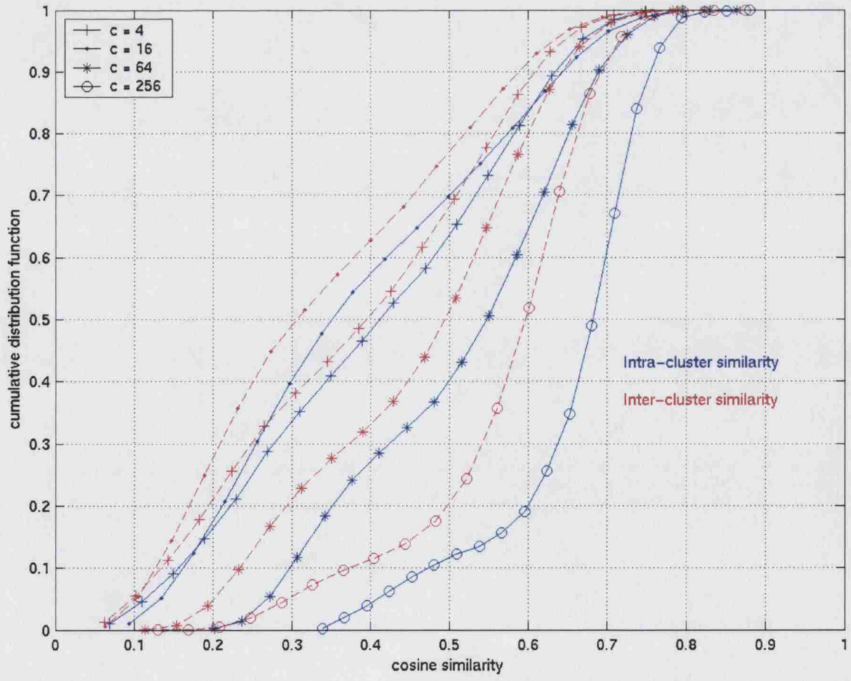


Figure 3.13: Cumulative distribution functions of the intra- and inter-cluster cosine similarity for the INSPEC document collection, obtained with H-FCM for c = 4, 16, 64 and 256.

The analysis is based on intra- and inter-cluster similarity distributions (the CDFs are shown in Figure 3.10 to Figure 3.13). Each document collection was clustered for c = 4, 16, 64 and 256 clusters. The intra-cluster distributions were obtained by calculating the similarities between the cluster centroids and documents belonging to the respective clusters and the inter-cluster distributions were obtained by calculating the similarities between the cluster centroids and documents belonging to other clusters.

From the plots we verify that regardless of the number of clusters there is a clear separation between intra- and inter-cluster distributions, that is, documents within a given cluster are always more similar to the corresponding centroid than documents outside that cluster. This relative difference proves that the H-FCM is able to meaningfully cluster high-dimensional sparse data.

It can also be observed that generically, the lower the number of clusters the more dissimilar the documents are to their cluster centroid. In fact, for c=4 at least 60% of the documents from all collections present an intra-cluster similarity below 0.5. This means that clusters are not compact and that most documents have just a few terms in common with the respective centroid vectors. However, as c increases the documents vectors become closer (in the cosine sense) to the centroid vectors, which means that the clusters become more compact. The separation between intra- and inter-cluster distributions also increases.

This leads us to conclude that the issue of finding the optimum number of clusters is not so relevant. The choice of c should rather reflect the level of refinement that is required in a given application. In Chapter 6, this issue is further investigated.

3.7 Summary

This chapter started by examining typical properties of real document collections. We have verified that in such problem space, data vectors exhibit high dimensionality and very low density. We have also addressed the selection of a suitable similarity coefficient for document clustering. The clustering tendency of each test document collection has been investigated through an analysis of the overlap between intra- and inter-class similarity distributions, for different similarity coefficients and for different term weighting schemes. The analysis has revealed that overall the cosine measure was the best performing coefficient and that document vectors encoded with the TF scheme led to a much better

separation between intra- and inter-class similarity distributions than when the TF-IDF scheme is used.

The selection of a fuzzy clustering method for document clustering has also been addressed in this chapter. We have reviewed various fuzzy clustering techniques and we have chosen the FCM algorithm, for its simplicity and linear time complexity, and also for being the fuzzy counterpart of a traditional document clustering method. We have pointed out the limitations of the Euclidean distance for assessing the proximity between document vectors. The FCM algorithm, though, originally applies the Euclidean distance to compute cluster memberships and centroid vectors. A replacement of this metric with similarity coefficients has also been proposed here and new mathematical expressions have been derived accordingly. The Hyper-spherical Fuzzy c-Means (H-FCM) algorithm is the result of this new development.

Finally, we have considered some characteristics of the H-FCM algorithm namely the handling of outliers and its dependence on user defined parameters, such as the number of clusters. We have argued that given the properties of our problem space, outliers are not expected to have great impact on the final location of the cluster centroids and that finding the optimum number of clusters c was not a central issue. Our reasoning indicates that the choice of c should rather reflect the desired granularity of document clusters.

In the next chapter, we thoroughly investigate the performance of the new H-FCM algorithm through a set of experiments with the test document collections.

Chapter 4
Evaluation of the H-FCM algorithm

4.1 Introduction

In the previous chapter, we have developed the H-FCM algorithm by modifying the original FCM algorithm so that the relationship between documents vectors of unit length could be assessed based on similarity coefficients rather than on the Euclidean distance. In this chapter, the suitability of the H-FCM algorithm for clustering document collections and for identifying meaningful content relationships is investigated.

In section 4.2, the evaluation measures that will be used throughout the investigation to assess the clustering performance are detailed. In section 4.3, a comparative study on the performance of the traditional FCM and of the H-FCM algorithm for document clustering is presented. In section 4.4, the effects of pre-processing the document vectors in terms of the H-FCM clustering outcome are analysed. In section 4.5, the performance of H-FCM with the cosine, Jaccard and overlap similarity coefficients are compared. In section 4.6, the H-FCM is compared with traditional hard clustering methods that have long been applied in cluster-based IR systems. Finally, a summary of the main contributions of this chapter is given in section 4.7.

4.2 Performance evaluation measures

In Chapter 2, we have introduced cluster validity measures for assessing the output of clustering methods. These can be classified into internal and external measures. The validity of fuzzy clustering algorithms is generally evaluated using internal measures, *i.e.* measures that are algorithm dependent and do not exploit any external knowledge about the actual structure of the data set. However, given that clustering benchmarks are available for our test document collections (see Table 3.2) external validity measures can also be applied to evaluate the H-FCM clustering results. These measures are algorithm independent since they simply rely on prior knowledge on how clusters should be formed. In the following sub-sections, the internal and external measures that will be used throughout this thesis are described.

4.2.1 Internal measures

There are several validity indexes that have been specifically developed for the FCM algorithm to evaluate the intrinsic quality of the generated clusters. Examples include Partition Entropy (*PE*) and Partition Coefficient (*PC*) indexes [18], Xie-Beni (*XB*) index [125], Fukuyama-Sugeno (*FS*) index [126] and Compose Within and Between Scattering (*CWBS*) index [127]. The *PE* and *PC* validity indexes indicate the closeness of a fuzzy partition to a hard one. These indexes are only a function of the cluster membership values. The *XB*, *FS* and *CWBS* indexes indicate compactness and separation of the fuzzy clusters. These indexes consider the membership values as well as the location of the data elements and cluster centroids.

The *PE* index is defined in equation (4.46). The possible values of *PE* range from 0, when the partition matrix is hard (*i.e.* $u_{\alpha i}=1$ if element x_i belongs to cluster α, and $u_{\alpha i}=0$ otherwise), to $\log_a(c)$ when every data object has equal membership in every cluster (*i.e.* $u_{\alpha i}=1/c$, $\forall_{\alpha\in\{1,\ldots,c\}}\ \forall_{i\in\{1,\ldots,N\}}$).

$$PE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}\log_a(u_{\alpha i}) \tag{4.46}$$

Dividing *PE* by $\log_a(c)$ normalises the value of *PE* to range in the [0,1] interval. Maximum *PE* is obtained in the case of maximum fuzziness, which indicates the algorithm

is unable to find a clustering structure. The *PC* index provides similar information to the *PE* index since minimum *PC* corresponds to maximum *PE* and vice-versa [128].

The *XB* index evaluates the quality of a fuzzy partition generated by FCM based on the compactness of each cluster and the separation between cluster centroids. It is based on the assumption that the more compact and separated the clusters are the better the clustering outcome. The *FS* and *CWBS* indexes provide similar information.

The *XB* index is defined in equation (4.47). If the minimum distance between any pair of clusters φ and γ is too low, then *XB* will be very high. Moreover, if a cluster α is compact then the distance between element x_i (belonging to that cluster with membership $u_{\alpha i}$) and the respective cluster centroid v_α will be quite small. Consequently, a good partition of the data set normally corresponds to a low value of *XB*.

$$XB = \frac{\sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}{}^{m}\left\|x_i - v_\alpha\right\|^2}{N \cdot \min\limits_{\substack{\varphi\neq\gamma \\ \varphi,\gamma\in[1,c]}} \left\|v_\varphi - v_\gamma\right\|^2} \tag{4.47}$$

The *PE* and *XB* validity indexes will be used for assessing the performance of the H-FCM algorithm. Although they were initially derived for the FCM clustering algorithm they are still applicable to the H-FCM. In the last case, however, a simple modification is required in the *XB* equation to replace the squared distance $\|\cdot\|^2$ with the dissimilarity function defined in equation (3.17),

$$XB = \frac{\sum_{i=1}^{N}\sum_{\alpha=1}^{c} u_{\alpha i}{}^{m} \cdot D_{i\alpha}}{N \cdot \min\limits_{\substack{\varphi\neq\gamma \\ \varphi,\gamma\in[1,c]}} D_{\varphi\gamma}} \tag{4.48}$$

No change is required in the *PE* expression since it is only a function of the fuzzy partition matrix.

4.2.2 External measures

As mentioned in section 2.6.2, precision and recall are two measures traditionally used for evaluating the performance of information retrieval systems [27, 41]. Similar measures have also been applied for the evaluation of text classification systems [92, 93, 94], whose purpose is to classify text documents given a known set of classes. Clustering

methods are in fact unsupervised classification systems and thus, the traditional definitions of precision and recall can be adapted to serve as external evaluation measures of clustering algorithms, whenever clustering benchmarks exist. Thus, we have adapted the definitions of precision and recall for the clustering context as follows.

Table 4.1: A two-way contingency table for discovered cluster γ and reference cluster Γ.

	In reference cluster Γ	Not in reference cluster Γ
Assigned to cluster γ	$\lvert\Omega_\gamma \cap \Omega_\Gamma\rvert$	$\lvert\Omega_\gamma \cap \overline{\Omega}_\Gamma\rvert$
Not assigned to cluster γ	$\lvert\overline{\Omega}_\gamma \cap \Omega_\Gamma\rvert$	$\lvert\overline{\Omega}_\gamma \cap \overline{\Omega}_\Gamma\rvert$

(Ω_γ - set of documents in cluster γ, Ω_Γ - set of documents in reference cluster Γ)

Given a discovered cluster γ and the associated reference cluster Γ and given the two-way contingency table above, precision ($P_{\gamma\Gamma}$) and recall ($R_{\gamma\Gamma}$) are now defined as,

$$P_{\gamma\Gamma} = \frac{\lvert\Omega_\gamma \cap \Omega_\Gamma\rvert}{\lvert\Omega_\gamma\rvert} = \frac{n_{\gamma\Gamma}}{N_\gamma} \qquad (4.49) \qquad R_{\gamma\Gamma} = \frac{\lvert\Omega_\gamma \cap \Omega_\Gamma\rvert}{\lvert\Omega_\Gamma\rvert} = \frac{n_{\gamma\Gamma}}{N_\Gamma} \qquad (4.50)$$

where $n_{\gamma\Gamma}$ is the number of documents from reference cluster Γ assigned to cluster γ, N_γ is the total number of documents in cluster γ and N_Γ is the total number of documents in reference cluster Γ.

Contingency tables such as the one above have long been used in ROC (Receiver Operating Characteristics) Analysis to measure the accuracy of diagnostic systems [129]. ROC curves plot true positive rates (TPR) vs. false positive rates (FPR). From Table 4.1, these metrics are define as: $TPR_{\gamma\Gamma} = n_{\gamma\Gamma}/N_\Gamma (= R_{\gamma\Gamma})$ and $FPR_{\gamma\Gamma} = (N_\Gamma - n_{\gamma\Gamma})/N_\Gamma = 1 - TPR_{\gamma\Gamma}$. In the field of IR, precision-recall curves are preferred to TPR-FPR curves, since they have been found better at evaluating IR systems effectiveness [27].

$P_{\gamma\Gamma}$ and $R_{\gamma\Gamma}$ can be combined into a single performance measure, the F-measure, which is defined as,

$$F^{\xi}_{\gamma\Gamma} = \frac{(\xi^2 + 1) \cdot P_{\gamma\Gamma} \cdot R_{\gamma\Gamma}}{\xi^2 \cdot P_{\gamma\Gamma} + R_{\gamma\Gamma}} \qquad (4.51)$$

where ξ is a parameter that controls the relative weight of precision and recall (ξ=1 is used for equal contribution).

To obtain overall performance measures that consider all c clusters, a weighted average of the individual $P_{\gamma\Gamma}$, $R_{\gamma\Gamma}$ and $F^{\xi}{}_{\gamma\Gamma}$ is applied:

$$P = \frac{\sum_{\Gamma=1}^{c} N_\Gamma P_{\gamma\Gamma}}{\sum_{\Gamma=1}^{c} N_\Gamma} \quad (4.52) \qquad R = \frac{\sum_{\Gamma=1}^{c} N_\Gamma R_{\gamma\Gamma}}{\sum_{\Gamma=1}^{c} N_\Gamma} \quad (4.53) \qquad F^{\xi} = \frac{\sum_{\Gamma=1}^{c} N_\Gamma F^{\xi}{}_{\gamma\Gamma}}{\sum_{\Gamma=1}^{c} N_\Gamma} \quad (4.54)$$

These measures consider the number of documents assigned to each cluster. As the H-FCM algorithm produces fuzzy clusters, it is possible that all documents belong to some degree to all clusters. In such case, precision is consequently low and recall is maximised. Hence, to obtain useful information with these evaluation measures, fuzzy clusters should be hardened before calculating precision and recall using, for instance, the maximum membership criterion (*i.e.* assigning the documents to the cluster where their membership value is higher). We follow this approach.

4.3 Comparison between FCM and H-FCM

In this section, the performance of the FCM and H-FCM algorithms is investigated both in terms of the intrinsic properties of the clustering partitions (*i.e.* based on internal performance measures) and of the actual quality of the discovered document clusters (*i.e.* based on external performance measures). The four test document collections presented in the previous chapter are used in all experiments to avoid a biased analysis towards a particular type of collection as explained earlier (see section 3.2). No pre-processing of the document vectors is performed at this stage. However, the effects of pre-processing are considered later in section 4.4. In this section, we use the cosine coefficient in the H-FCM algorithm because we have previously concluded that this similarity measure is the most suitable for document clustering (see section 3.3). Other similarity coefficients are considered later in section 4.5.

The main goal of these experiments is to establish whether the H-FCM algorithm succeeds in discovering meaningful clustering structures and to compare its performance to that of the original FCM algorithm. Both TF and TF-IDF term weighting schemes are considered to determine which of these schemes is more suitable for document clustering.

As mentioned in section 3.6.2, the algorithms require the definition of the final number of clusters c and of the fuzzification parameter m. The selection of c is in

accordance with the number of reference clusters of each collection, which is defined in Table 3.2 ($c_{REUTERS1}$=3, $c_{REUTERS2}$=5, c_{ODP}=5 and c_{INSPEC}=3). The selection of m needs to satisfy a compromise. On the one hand, document clusters have to exhibit a certain degree of fuzziness in order to handle uncertainty in the knowledge space. On the other hand, the degree of fuzziness must not be excessive, otherwise document relationships can be meaningless. These considerations restrict the range of values for m.

A suitable range of values for the fuzzification parameter has been proposed in [128] through empirical studies with a low-dimensional data set and various cluster validity indexes. The proposed range was $m \in [1.50, 2.50]$. However, studies with sparse high-dimensional data sets have not been carried out. In the previous chapter, we have shown that sparse high-dimensional spaces have distinct properties, namely the very low similarities between data vectors. Such similarity patterns restrict the range of suitable values for m. To support this statement we give the following example.

Figure 4.1 shows the impact of m in the shape of the membership functions for three clusters of points in the real line. It can be seen that the higher the value of m, the lower the membership of a given point in its closest cluster and the higher its membership in other clusters. Considering now the high-dimensional document space, we have previously verified that documents within a cluster can present relatively low similarities to the respective cluster centroid. Consequently, even relatively low values of m can lead to the undesirable situation of maximum fuzziness. Thus, in our experiments we restrict the values of m to the interval [1.10, 1.50].

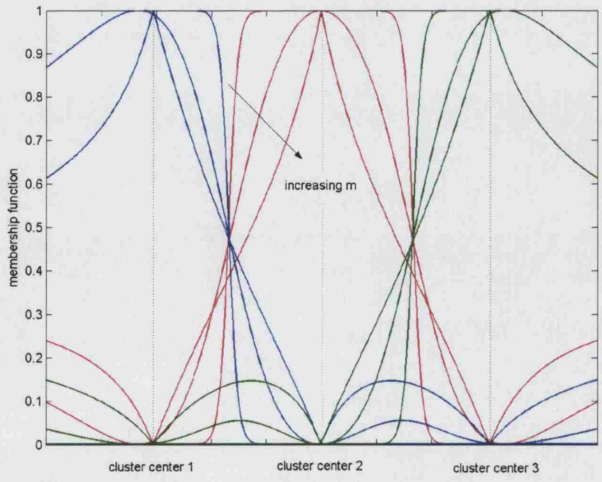


Figure 4.1: Membership of points in the real line in 3 clusters, obtained with FCM for increasing values of m (1.10, 1.50 and 2.00).

The results of the internal and external performance evaluation are presented and analysed in the next sub-sections.

4.3.1 Internal performance evaluation

Figures 4.2 through 4.9 compare the intrinsic performance of FCM and H-FCM. The plots in Figures 4.2, 4.4, 4.6 and 4.8 show the values of the normalised *PE* for increasing values of *m*, obtained for the REUTERS1, REUTERS2, ODP and INSPEC collections, respectively. The plots in Figures 4.3, 4.5, 4.7 and 4.9 show the values of the *XB* index also for increasing values of *m*, for the different collections.

From the results it can be seen that the TF weighting scheme always leads to better clustering performances than the TF-IDF scheme, both in the FCM case and in the H-FCM case. For a fixed value of *m*, the values of *PE* and *XB* are generally lower with TF data vectors, which means that the partition matrix is further away from the undesirable maximum fuzziness case and that the clusters are better separated from each other and more compact than in the TF-IDF case. In fact, it can be observed from the *PE* plots that with this weighting scheme the FCM algorithm is unable to find a clustering structure even for very low values of *m*. The poor performance of the TF-IDF scheme is also verified with the H-FCM algorithm. It can also be observed that with REUTERS2 and ODP collections the H-FCM performs slightly better in terms of *PE* for a broader range of *m* values. However, the *XB* plots show that the clusters are not well separated for most *m* values where $PE/\log(c)<1$.

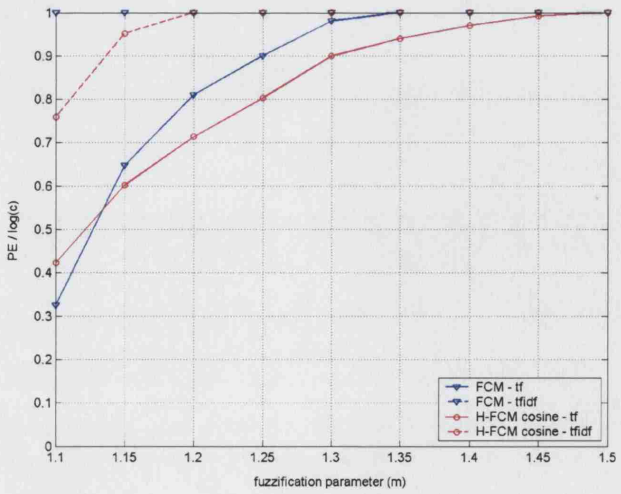


Figure 4.2: Normalised Partition Entropy for the REUTERS1 document collection.

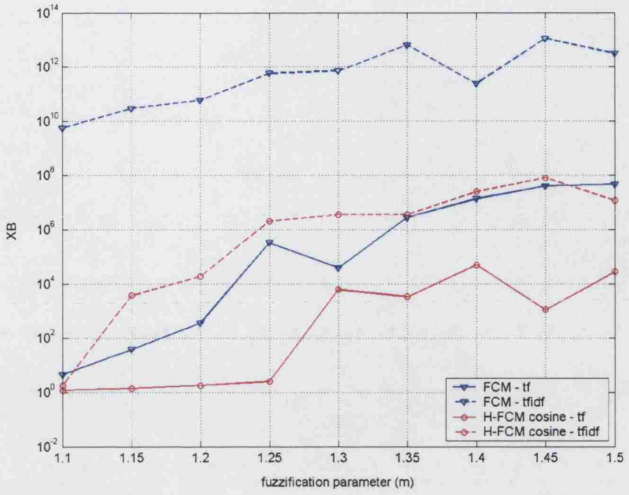


Figure 4.3: Xie-Beni index for the REUTERS1 document collection.

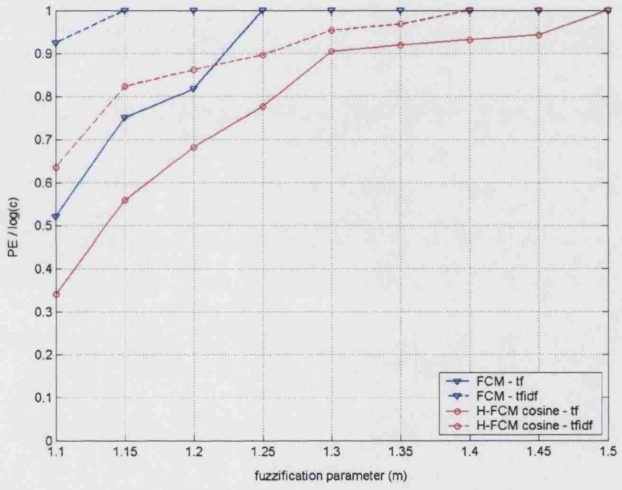


Figure 4.4: Normalised Partition Entropy for the REUTERS2 document collection.

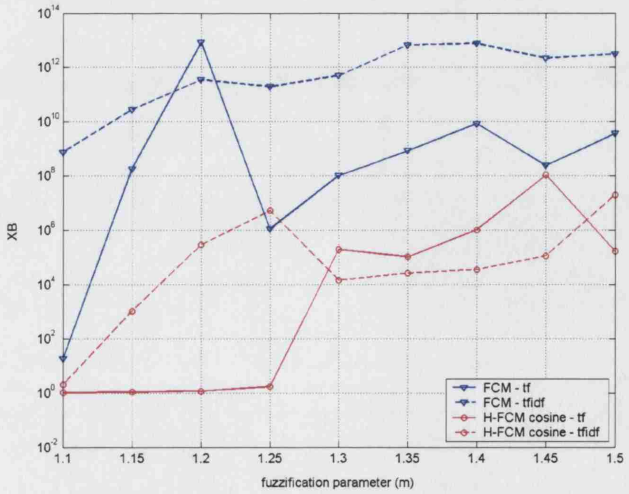


Figure 4.5: Xie-Beni index for the REUTERS2 document collection.

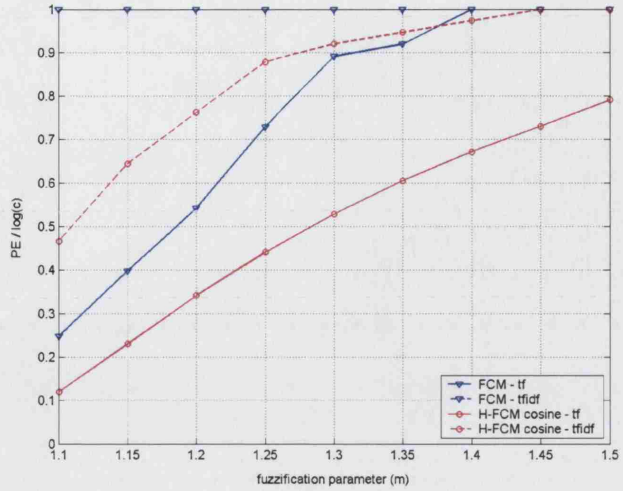


Figure 4.6: Normalised Partition Entropy for the ODP document collection.

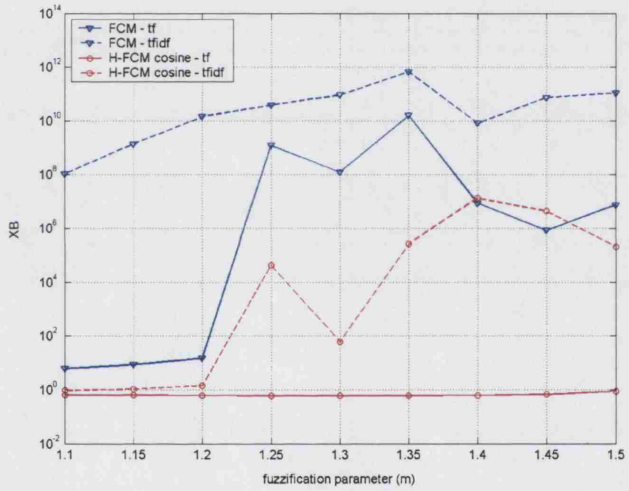


Figure 4.7: Xie-Beni index for the ODP document collection.

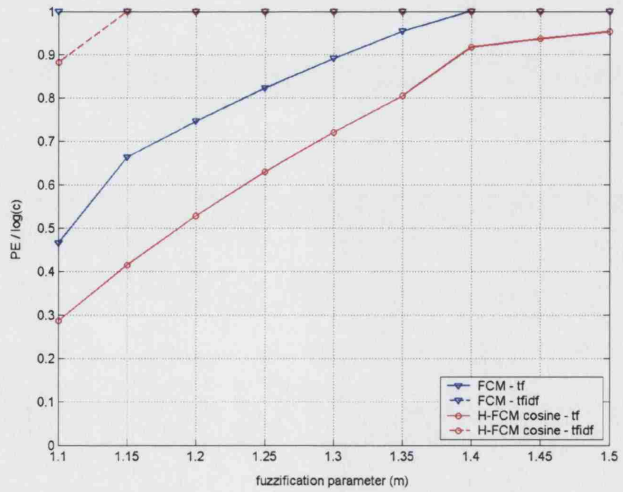


Figure 4.8: Normalised Partition Entropy for the INSPEC document collection.

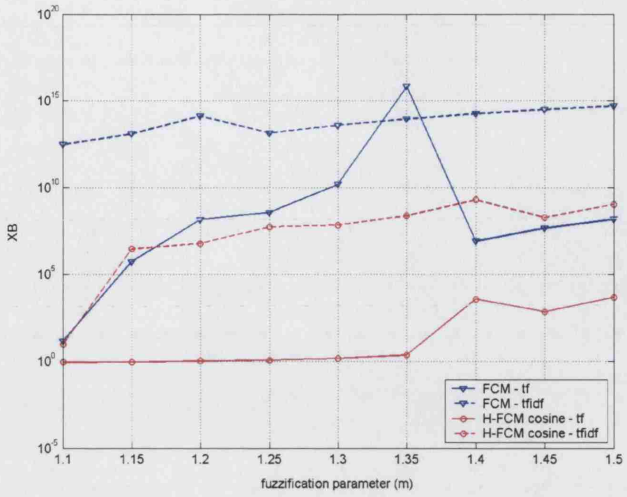


Figure 4.9: Xie-Beni index for the INSPEC document collection.

From this comparison between TF and TF-IDF weighting schemes we verify that these results are in accordance with those from the clustering tendency tests. In section 3.3.1, we have seen that the TF scheme outperformed the TF-IDF scheme in terms of separation between intra- and inter-class similarity distributions. Now, we have shown that the clustering algorithms also perform much better with the TF encoded data.

Comparing the performance of FCM and H-FCM with TF data vectors, the results show that H-FCM performs consistently better than the original FCM. It can be seen from the *PE* graphs (Figures 4.2, 4.4, 4.6 and 4.8) that for any given value of m the FCM always obtains a partition closer to the undesirable maximum fuzziness case, with a single exception for the REUTERS1 collection when m=1.10. However, the *XB* graphs (Figures 4.3, 4.5, 4.7 and 4.9) indicate that even in that case, the H-FCM clusters are more compact and more clearly separated. To obtain further insight into this result, Figure 4.10 shows the percentage of documents from the REUTERS1 collection that have cluster membership ≥0.5 in one of the clusters. It can be observed that the FCM algorithm achieves higher percentage for m=1.10 when the TF scheme is used. This justifies the lower *PE* value, but given the *XB* value we can conclude that in the FCM case more of those high membership documents are assigned to the wrong cluster.

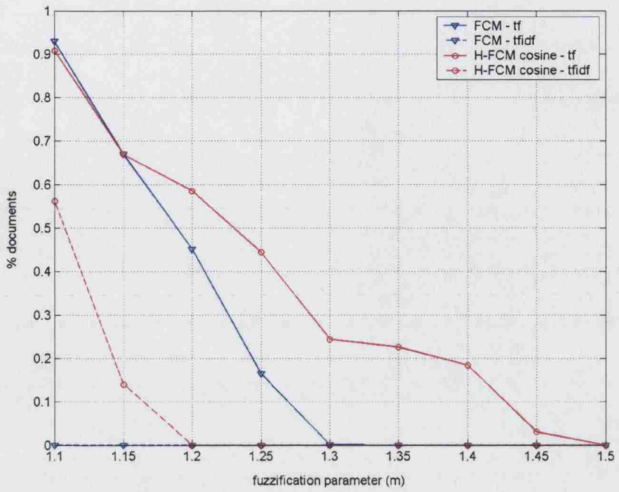


Figure 4.10: Percentage of the total number of documents from the REUTERS1 collection that have membership ≥0.5 in one of the clusters.

Analysing the behaviour of the algorithms for increasing values of the fuzzification parameter, it can be observed that H-FCM (with TF data vectors) is able to find good clusters for a wider range of values. As expected, *PE* increases as *m* increases but the clusters continue to be well separated for $m \leq 1.25$ in the REUTERS1 and REUTERS2 cases, $m \leq 1.35$ in the INSPEC case and for all $m \in [1.10, 1.50]$ in the ODP case. Even with the collection that provides the best results (ODP), the FCM algorithm only generates well separated clusters for very low values of *m* (≤ 1.20). To conclude, this analysis of internal performance shows that H-FCM clearly outperforms FCM.

4.3.2 External performance evaluation

Through internal validity measures we have verified that the H-FCM outperforms the FCM algorithm. In this sub-section, we exploit knowledge of the reference clusters of each test document collection to analyse the external performance of both algorithms. Figures 4.11 through 4.22 present the results of this comparison. The plots in Figures 4.11, 4.14, 4.17 and 4.20 show the values of the average precision of the generated clusters for increasing values of *m*, obtained for the REUTERS1, REUTERS2, ODP and INSPEC collections, respectively. The plots in Figures 4.12, 4.15, 4.18 and 4.21 show the values of the average recall of the generated clusters and the plots in Figures 4.13, 4.16, 4.19 and 4.22 show the values of the F'-measure, also for increasing values of *m*. The measures have been

calculated after hardening the clusters according to the maximum membership criterion, *i.e.* documents have been allocated to the cluster in which their membership was higher.

A good clustering performance corresponds to high values of both precision and recall. Ideally, they should both equal 1 which would mean that every cluster contained all and only the right documents. The F^1-measure is a combination of precision and recall and hence the higher it is the better. Low values of these measures imply poor algorithm performance. In particular, poor performance is verified in the extreme case of maximum fuzziness, *i.e.* when every document has the same membership in all clusters ($u_{\gamma i}=1/c$, $\forall_{\gamma\in\{1,\ldots,c\}}\ \forall_{i\in\{1,\ldots,N\}}$ and $N_\gamma = N,\ \forall_{\gamma\in\{1,\ldots,c\}}$), which means the algorithm was unable to find a meaningful clustering structure. So, before analysing the results we consider the theoretical limits for precision, recall and F^1-measure in such case. Given the definitions in equations (4.49), (4.50) and (4.51), for each individual cluster it follows that,

$$P_{\gamma\Gamma} = \frac{n_{\gamma\Gamma}}{N_\gamma} = \frac{N_\Gamma}{N},\ \forall_{\gamma\in\{1,\ldots,c\}} \qquad (4.55) \qquad R_{\gamma\Gamma} = \frac{n_{\gamma\Gamma}}{N_\Gamma} = \frac{N_\Gamma}{N_\Gamma} = 1,\ \forall_{\gamma\in\{1,\ldots,c\}} \qquad (4.56)$$

$$F^1{}_{\gamma\Gamma} = \frac{2\cdot P_{\gamma\Gamma}\cdot R_{\gamma\Gamma}}{P_{\gamma\Gamma}+R_{\gamma\Gamma}} = \frac{2\cdot N_\Gamma}{N_\Gamma + N},\ \forall_{\gamma\in\{1,\ldots,c\}} \qquad (4.57)$$

The weighted averages of the individual $P_{\gamma\Gamma}$, $R_{\gamma\Gamma}$ and $F^1{}_{\gamma\Gamma}$ measures result in,

$$P = \frac{1}{N}\frac{\sum_{\Gamma=1}^{c} N_\Gamma{}^2}{\sum_{\Gamma=1}^{c} N_\Gamma} \qquad (4.58) \qquad R = \frac{\sum_{\Gamma=1}^{c} N_\Gamma}{\sum_{\Gamma=1}^{c} N_\Gamma} = 1 \qquad (4.59) \qquad F^1 = \frac{\sum_{\Gamma=1}^{c} \frac{2\cdot N_\Gamma{}^2}{N_\Gamma + N}}{\sum_{\Gamma=1}^{c} N_\Gamma} \qquad (4.60)$$

Calculating these limits for the test document collections (using the class information in Table 3.2) yields the values presented in Table 4.2. Given these limits, we now proceed with the results analysis.

Table 4.2: Precision, recall and F^1 values for maximum fuzziness of the partition matrix.

Collection	Average *precision*	Average *recall*	Average F^1-measure
REUTERS1	0.40	1.00	0.55
REUTERS2	0.28	1.00	0.43
ODP	0.28	1.00	0.42
INSPEC	0.36	1.00	0.52

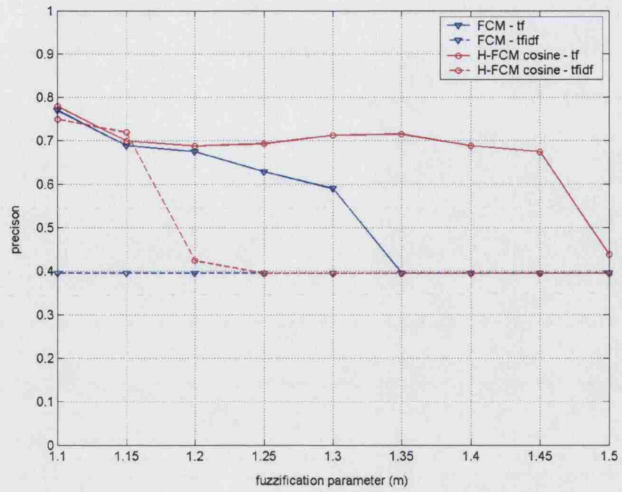


Figure 4.11: Average precision for the REUTERS1 document collection.

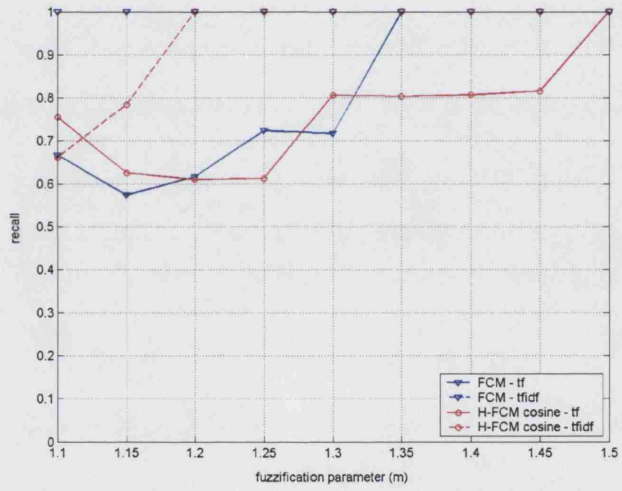


Figure 4.12: Average recall for the REUTERS1 document collection.

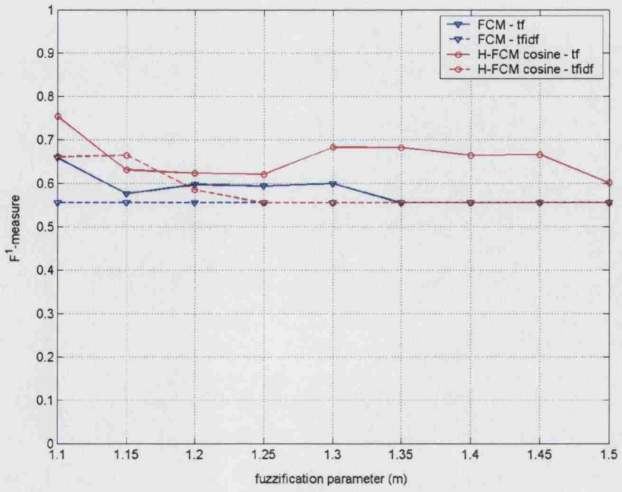


Figure 4.13: Average F^1-measure for the REUTERS1 document collection.

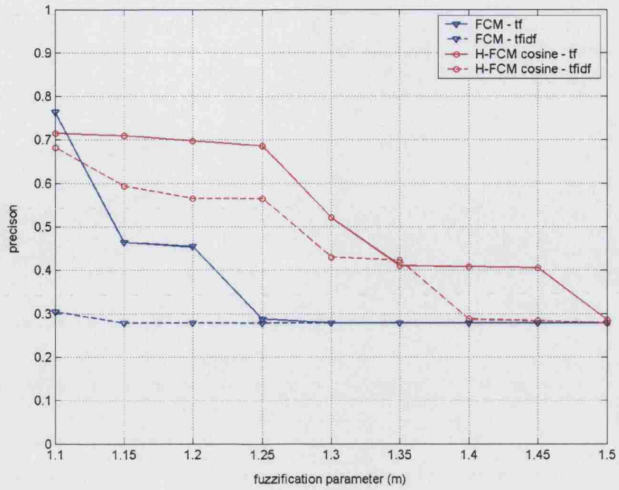


Figure 4.14: Average precision for the REUTERS2 document collection.

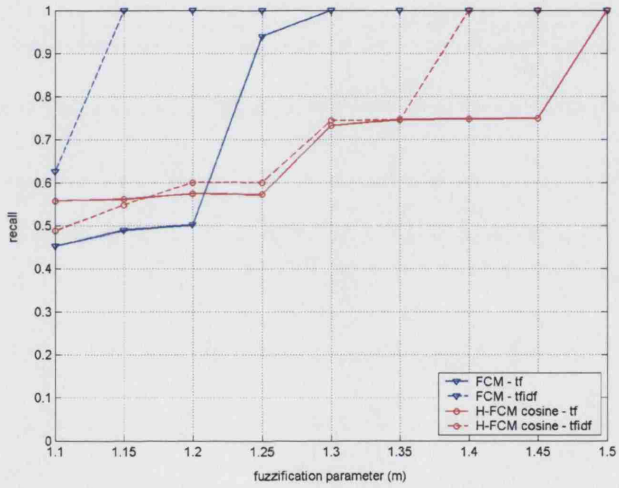


Figure 4.15: Average recall for the REUTERS2 document collection.

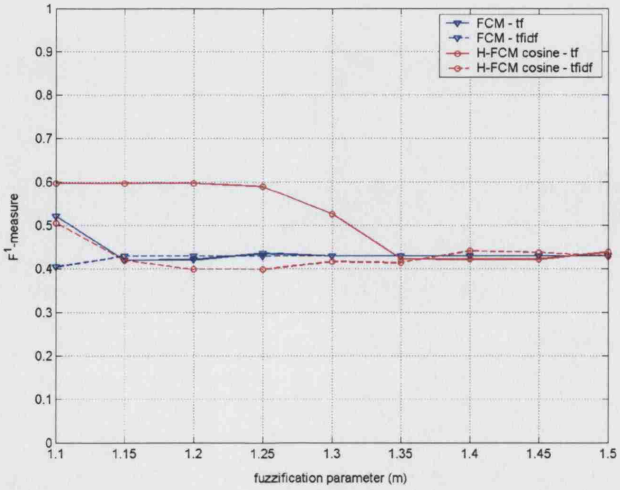


Figure 4.16: Average F^1-measure for the REUTERS2 document collection.

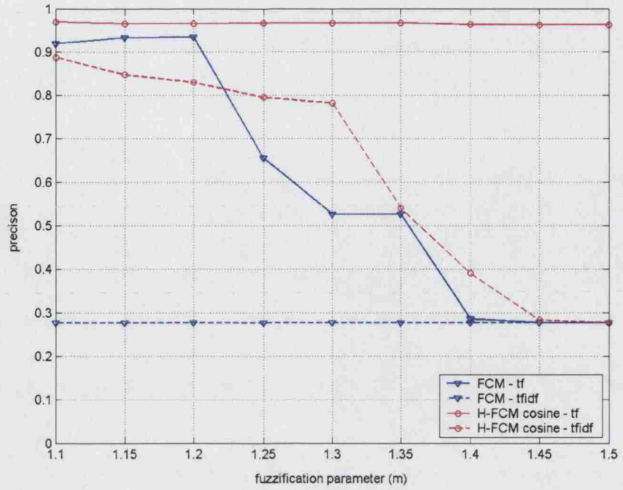


Figure 4.17: Average precision for the ODP document collection.

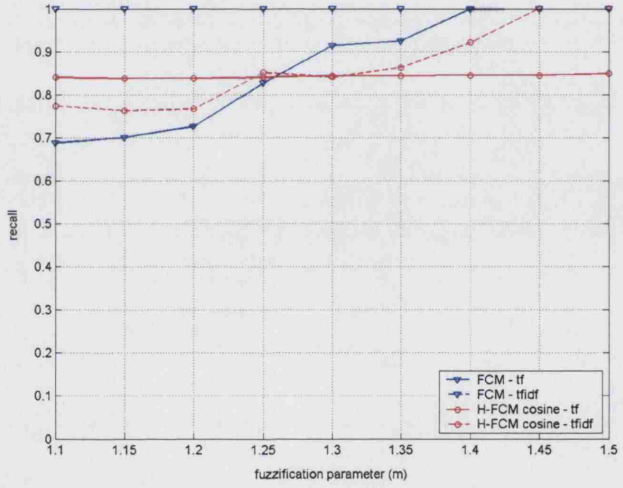


Figure 4.18: Average recall for the ODP document collection.

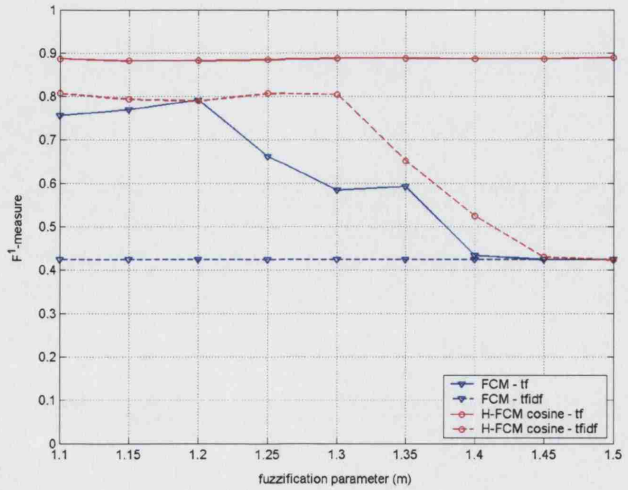


Figure 4.19: Average F^1-measure for the ODP document collection.

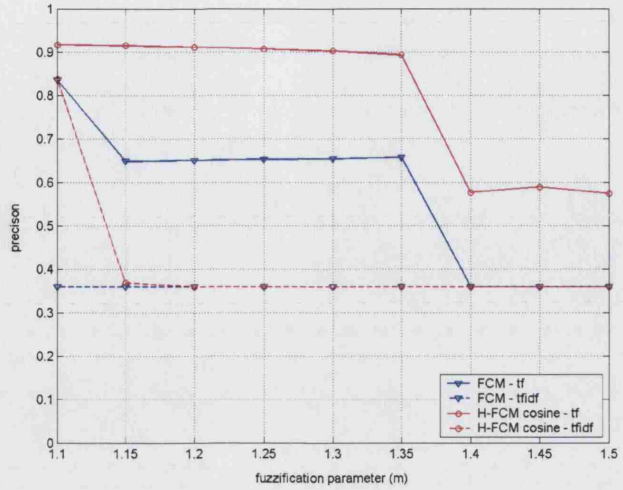


Figure 4.20: Average precision for the INSPEC document collection.

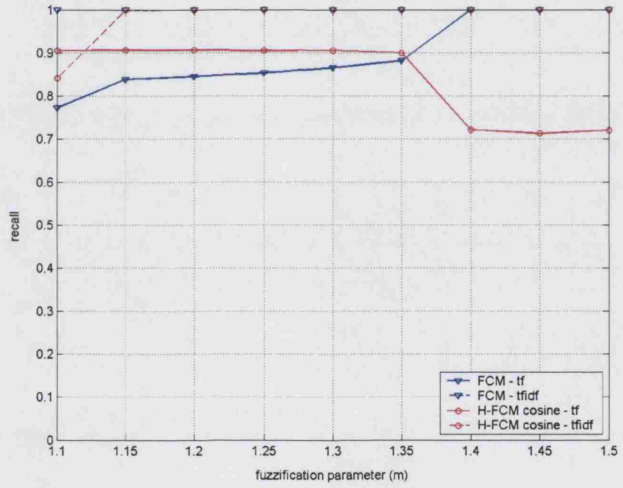


Figure 4.21: Average recall for the INSPEC document collection.

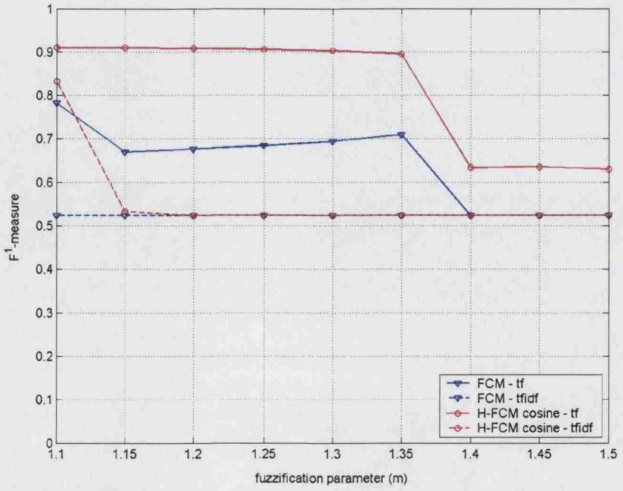


Figure 4.22: Average F^1-measure for the INSPEC document collection.

From the plots it can be verified that for the same range of m values in which maximum PE had been previously observed, precision, recall, and F^l correspond to the values in Table 4.2. This is expected as maximum PE corresponds to maximum fuzziness. Comparing the two term weighting schemes, it can be observed that the clustering performances of both algorithms when documents are encoded with TF-IDF are generally worse than when TF is used. Although not very significant, there is one exception with H-FCM when applied to the REUTERS1 collection. For m=1.15, the F^l-measure is slightly higher with the TF-IDF data essentially because of a higher recall level. However, the recall level is not due to a good clustering performance but rather due to a much higher PE (see Figure 4.2). The results from this comparison between TF and TF-IDF agree with the internal performance results from the previous sub-section.

Comparing the performance of FCM and H-FCM with TF data vectors, the results show that once again H-FCM performs consistently better than the original FCM. It can be seen from the F^l plots (Figures 4.13, 4.16, 4.19 and 4.22) that H-FCM always gives higher values for this measure, *i.e.* it gives the best compromise between precision and recall, except for $m \in [1.35,145]$ in the REUTERS2 case. However, in this case the clusters produced by FCM are not meaninful - Figure 4.4 shows that maximum fuzziness was reached.

In the previous sub-section, when comparing the performance of FCM and H-FCM with the REUTERS1 collection for m=1.10 we have argued that although FCM resulted in lower PE, that was due to assignment of documents to the wrong clusters. The precision and recall graphs in Figures 4.11 and 4.12, respectively, support that statement. It can be verified that for m=1.10, precision and recall are both lower with the FCM algorithm, which means that clusters contain more wrong documents and that a higher percentage of right documents are missing.

Analysing the behaviour of the algorithms for increasing values of m, we verify that the results are consistent with those from the internal evaluation. H-FCM is able to find good clusters for a wider range of m values than FCM. In fact, the quality of the clusters does not vary much as m is increases up to 1.25 in the REUTERS2 case, up to 1.35 in the INSPEC case and for all $m \in [1.10, 1.50]$ in the ODP case. In the REUTERS1 case, the H-FCM performance drops as m increases from 1.10 to 1.15, but it is still better than the FCM performance.

The results obtained with the four document collections are distinct in terms of the absolute values of precision and recall. In the previous chapter, we have presented the

intrinsic properties of the document collections, such as average document length, and we have shown that the collections exhibit differences in their intra- and inter-class cosine similarity distributions (see Figure 3.4). The differences observed in the properties of the document collections justify the differences observed in the clustering performances.

4.4 Pre-processing effects on the performance of H-FCM

In Chapter 2, we have reviewed common pre-processing techniques for reducing the dimensionality of document vectors. Such techniques fall under two main categories: re-parameterisation and filtering methods. Term frequency thresholding is a well established filtering method in the field of IR [49, 56] and in text classification systems [57]. Term entropy and term specificity filters are two common term frequency thresholding methods. In this section, we investigate the impact of pre-processing the document vectors through term frequency thresholding on the performance of the H-FCM clustering algorithm. The main goal of these experiments is to determine whether there are any benefits in pre-processing the document vectors regarding the quality of the discovered clusters. The TF representations of the four document collections are used in these experiments, since we have shown previously that the TF scheme leads to much better clustering performances than the TF-IDF scheme.

4.4.1 Term entropy and term specificity filters

In section 3.3.1, we have mentioned that entropy and specificity are two measures of term importance in a given document collection [49]. The entropy measure is derived from Shannon's information theory [53] and it acknowledges that the higher the probability of occurrence of a given term in the document collection, the less information it contains. Entropy of a term is defined as follows:

$$H_j = \sum_{i=1}^{N} p_{ij} \cdot \log_2 \frac{1}{p_{ij}}, \quad \text{with } p_{ij} = \frac{f_{ij}}{F_j} \text{ and } F_j = \sum_{i=1}^{N} f_{ij} \tag{4.61}$$

where p_{ij} is the probability of term j appearing in document i and F_j is the sum of the frequencies of term j (*i.e.* f_{ij}) across all N documents.

The specificity measure is generally applied to identify the least important terms as those that are present only in a small percentage of documents (*i.e.* very specific terms) or as those that are present in almost every document (*i.e.* too general). Specificity of a term is defined as follows [49]:

$$SP_j = \log(N / n_j) \tag{4.62}$$

where N is the total number of documents in the collection and n_j is the number of documents containing term j.

From equations (4.61) and (4.62) we can see that both measures include information about the occurrence of terms across the whole collection. However, the entropy measure takes into account term frequencies whereas the specificity measure only takes into account the presence or absence of terms.

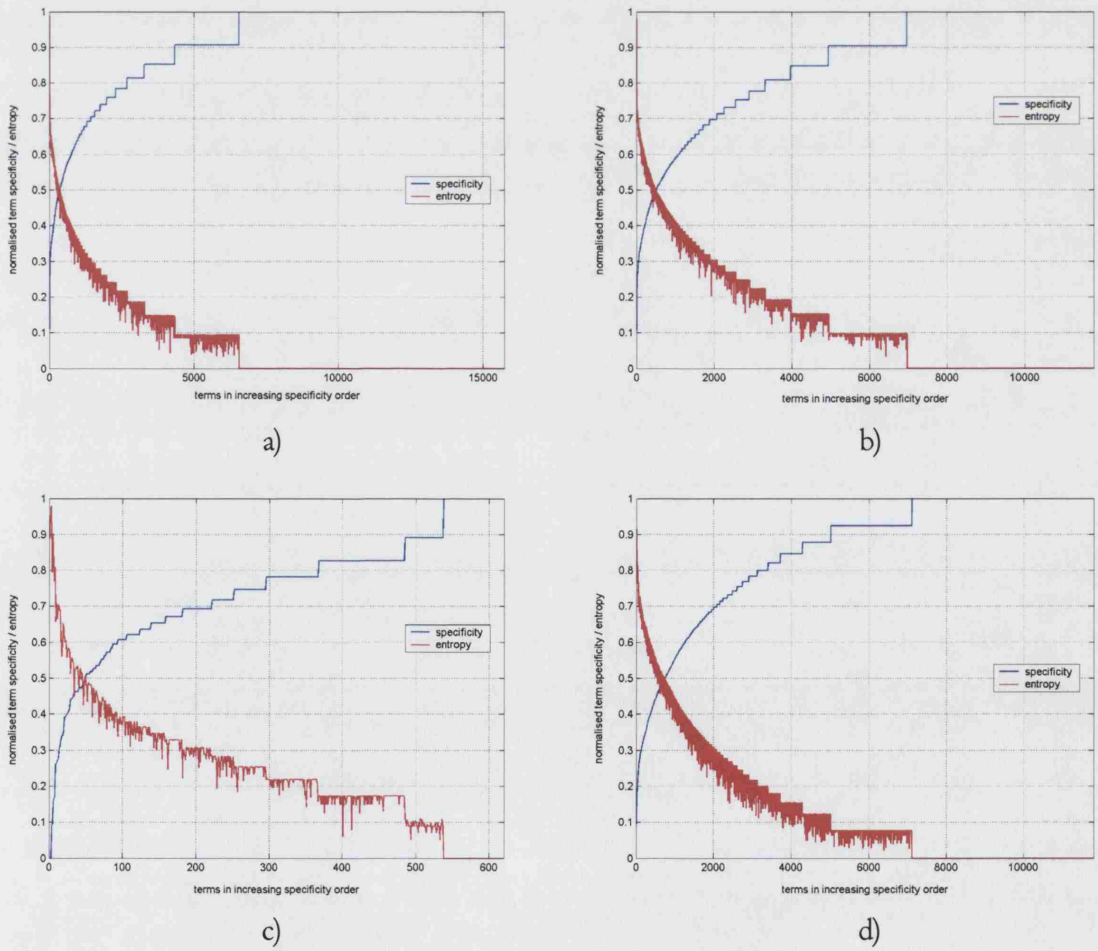


Figure 4.23: Normalised term specificity and entropy as a function of their increased specificity, for the a) REUTERS1, b) REUTERS2, c) ODP and d) INSPEC document collections.

Figure 4.23 shows the relationship between the two measures. Note that for the plots, these measures have been normalised to range in the unit interval. We have calculated the specificity and entropy of the indexing terms of the four document collections and we have sorted the terms according to increasing specificity. The plots show that high entropy terms correspond to the least specific terms in the collections and that low entropy terms correspond to the most specific terms. Therefore, applying pre-processing filters based on either of these measures should lead to identical results. Since it is less computationally demanding to compute specificity we choose this filter for our experiments.

4.4.2 Impact of the term specificity filter

In this sub-section, we apply the term specificity filter defined in equation (4.62) to reduce the dimensionality of the document vectors before clustering the collections with the H-FCM algorithm. Considering the full set of indexing terms in each collection, we define several thresholds for the specificity filter and discard terms according to the following criteria:

1. terms t_j with specificity $SP_j > \tau_{low} \cdot SP_{max}$ are removed (keeps common terms that occur in many documents)
2. terms t_j with specificity $SP_j < \tau_{high} \cdot SP_{max}$ are removed (keeps specific terms that occur in few documents)

where τ_{low} and τ_{high} are the threshold levels (that range between 0 and 1) and SP_{max} is the theoretical maximum specificity that occurs for terms which only occur in a single document,

$$SP_{max} = \log(N/1) = \log N \tag{4.63}$$

Each of these filters is considered separately in the experiment to analyse the effects each of them has on the clustering performance.

We have generated TF representations of the reduced document vectors for each threshold level of the low and high specificity filters. The H-FCM algorithm has been applied to each of the reduced data matrices and the evaluation of its performance is now carried out.

Table 4.3: Percentage of indexing terms filtered out from each document collection for several thresholds (τ_{low}) of the low specificity filter.

Collection	τ_{low}					
	0.90	0.80	0.70	0.60	0.50	0.40
REUTERS1	72.5%	82.9%	90.9%	95.1%	97.7%	99.0%
REUTERS2	58.0%	71.8%	82.3%	90.4%	95.6%	98.1%
ODP	13.4%	40.8%	64.4%	85.0%	92.1%	96.1%
INSPEC	57.4%	71.2%	82.4%	89.3%	93.6%	96.8%

The percentage of indexing terms that have been discarded with low specificity filters for different threshold levels is shown in Table 4.3. Figures 4.24 to 4.27 present the performance of the H-FCM algorithm, in terms of average F^1-measure, as a function of the number of indexing terms kept in each collection after applying the low specificity filter with the thresholds above. The experiment has been repeated by running the algorithm for several values of the fuzzification parameter m.

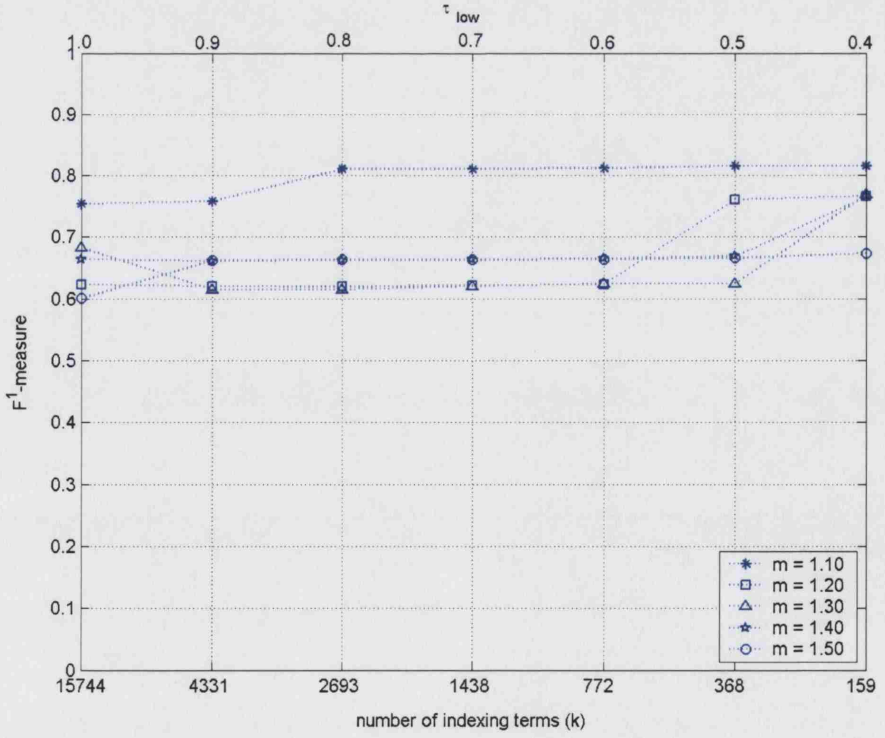


Figure 4.24: Impact of the low specificity filter on the external performance of the H-FCM for the REUTERS1 collection (average F^1-measure vs. number of indexing terms).

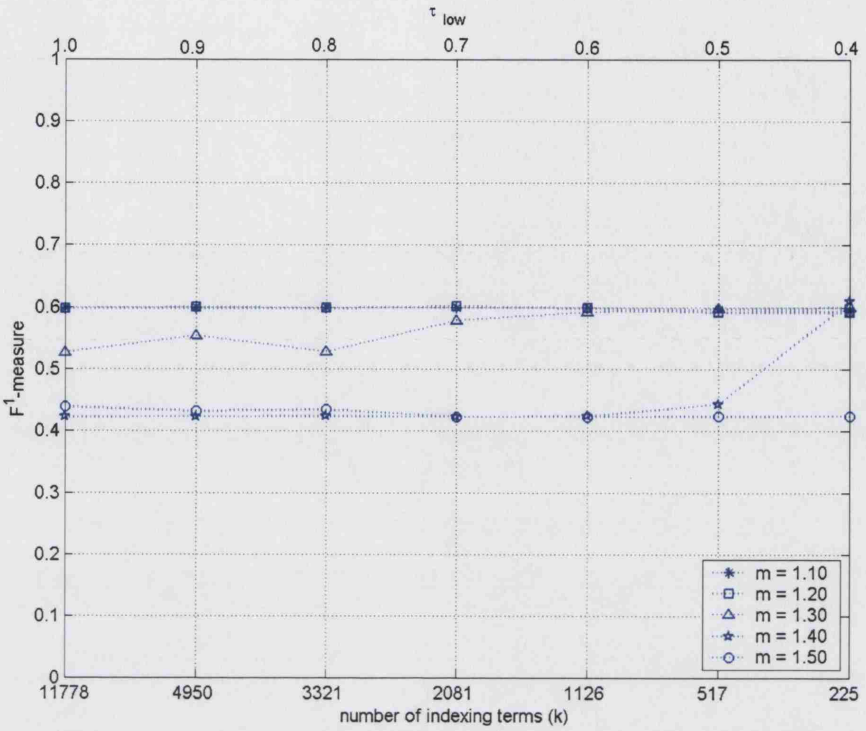


Figure 4.25: Impact of the low specificity filter on the external performance of the H-FCM for the REUTERS2 collection (average F^1-measure vs. number of indexing terms).

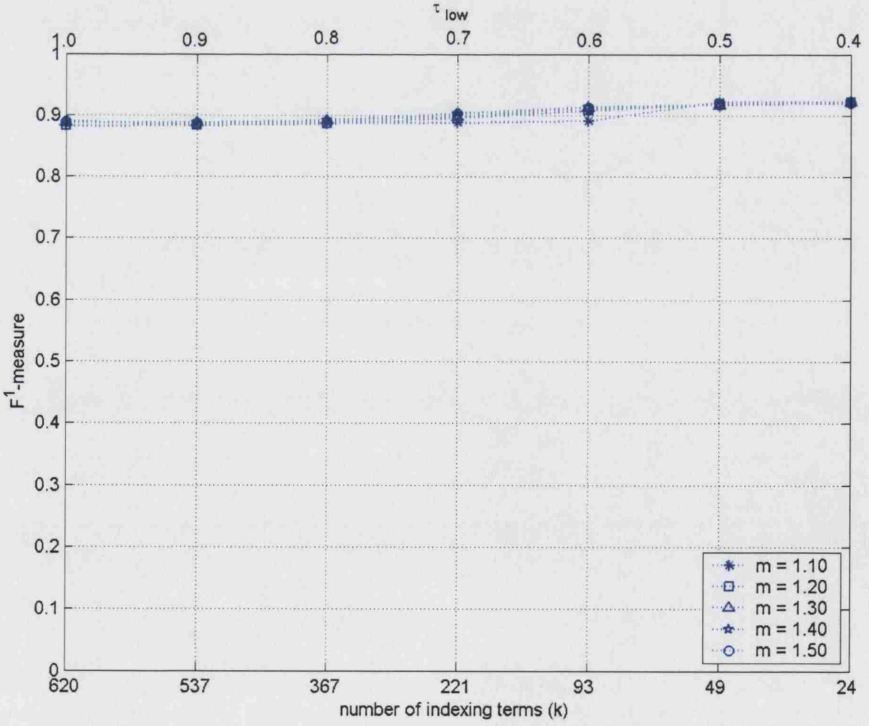


Figure 4.26: Impact of the low specificity filter on the external performance of the H-FCM for the ODP collection (average F^1-measure vs. number of indexing terms).

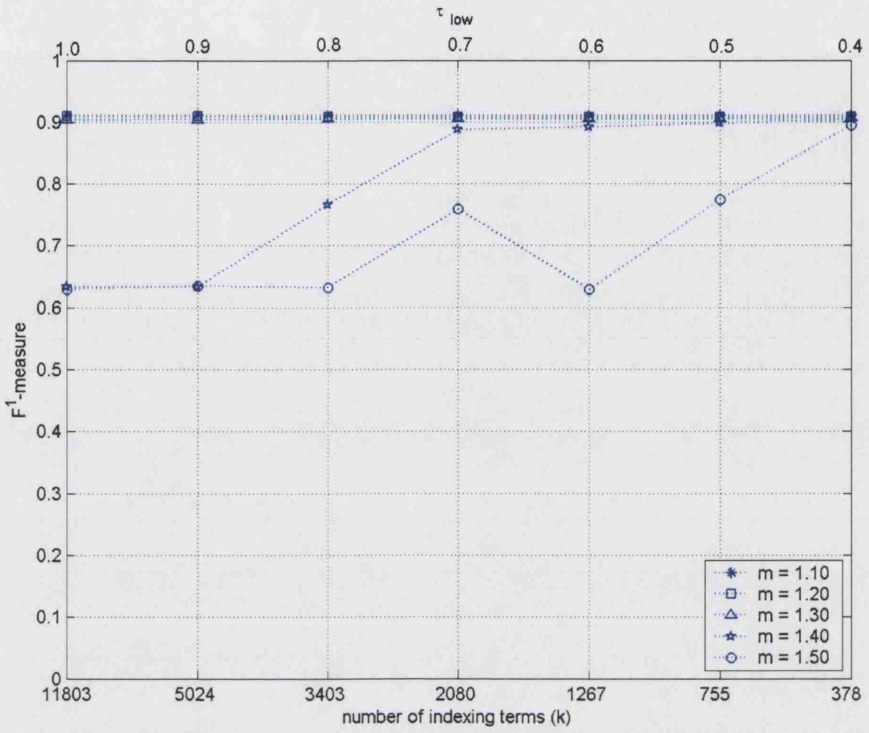


Figure 4.27: Impact of the low specificity filter on the external performance of the H-FCM for the INSPEC collection (average F^1-measure vs. number of indexing terms).

From the results it can be observed that removing terms which are very specific does not decrease the performance of the algorithm. In fact, the more terms are removed the better the H-FCM algorithm performs for m values which, without pre-processing, had led to situations close to maximum fuzziness (*e.g.* $m \geq 1.40$ in the INSPEC case). Therefore, higher m values can be used. This can be justified by the reduced sparsity of the document vectors when dimensions are eliminated and consequent increase of intra-cluster cosine similarities. We note that the cosine coefficient produces a broader range of similarity values when the sparsity of the problem space is lower (as it was shown in section 3.3.2).

Pre-processing the document collections with low specificity filters has an obvious great advantage in terms of memory requirements and computational cost, as drastic dimensionality reduction can be achieved (see Table 4.3).

Next, we consider the impact of the high specificity filters on the performance of the H-FCM. Figures 4.28 to 4.31 show the average F^1-measure as a function of the number of indexing terms kept in each collection after applying high specificity filters with the thresholds listed in Table 4.4. This table shows the percentage of terms that have been discarded in each case.

Table 4.4: Percentage of indexing terms filtered out from each document collection for several thresholds (τ_{high}) of the high specificity filter.

Collection	τ_{high}								
	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
REUTERS1	0.02%	0.08%	0.21%	1.0%	2.3%	4.9%	9.1%	17.0%	27.3%
REUTERS2	0.02%	0.12%	0.51%	1.9%	4.4%	9.5%	17.4%	27.8%	41.0%
ODP	0.8%	1.3%	2.1%	3.7%	6.9%	13.2%	27.3%	35.3%	38.4%
INSPEC	0.08%	0.34%	1.1%	3.2%	6.0%	9.3%	13.2%	19.4%	27.0%

The experiment has been once again repeated by running the H-FCM algorithm for several values of the fuzzification parameter m. From the results it can be seen that even discarding a very small percentage of the most common terms (*e.g.* τ_{high}=0.20) has great impact on the performance, especially with the ODP and INSPEC collections that present relatively low average document lengths. For most m values, after eliminating just 33 indexing terms from REUTERS1, 14 terms from REUTERS2, 13 terms from ODP and 40 terms from INSPEC, the F^1-measure decreases to values that indicate maximum fuzziness of the resulting clusters.

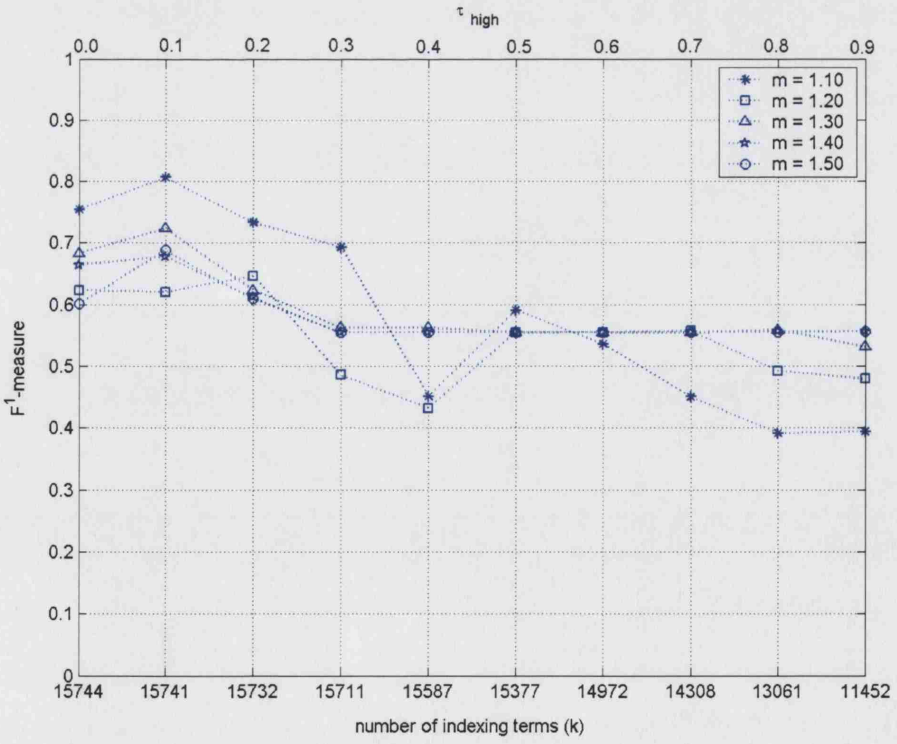


Figure 4.28: Impact of the high specificity filter on the external performance of the H-FCM for the REUTERS1 collection (average F^1-measure vs. number of indexing terms).

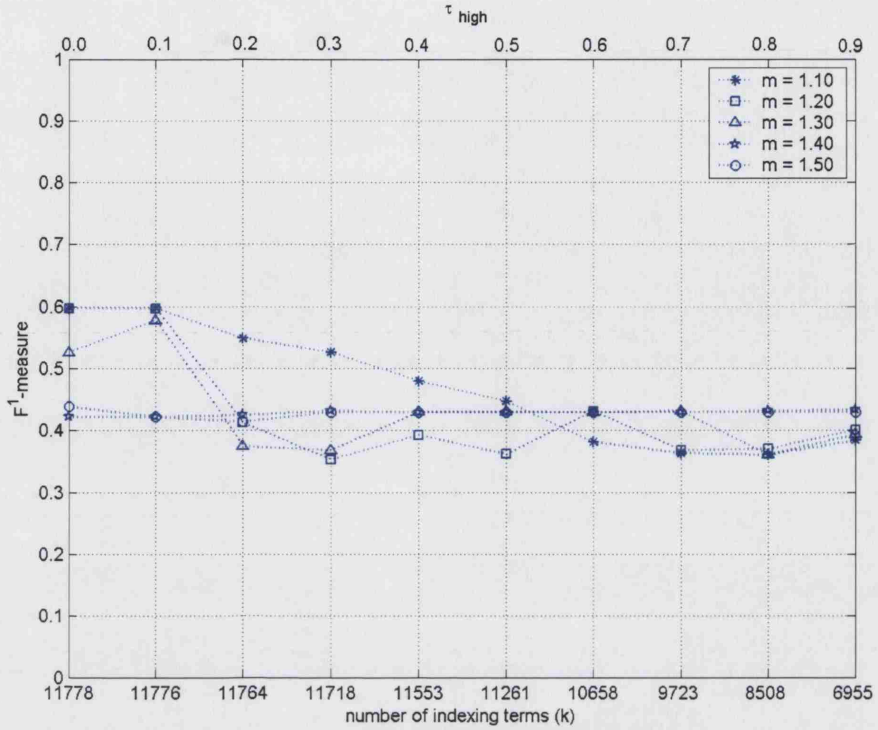


Figure 4.29: Impact of the high specificity filter on the external performance of the H-FCM for the REUTERS2 collection (average F^1-measure vs. number of indexing terms).

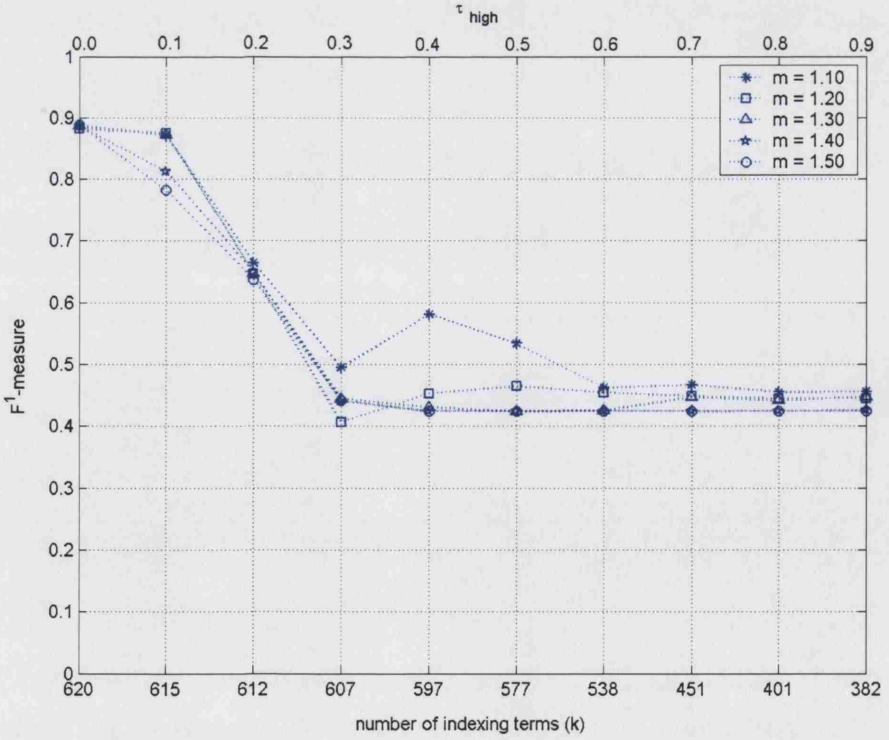


Figure 4.30: Impact of the high specificity filter on the external performance of the H-FCM for the ODP collection (average F^1-measure vs. number of indexing terms).

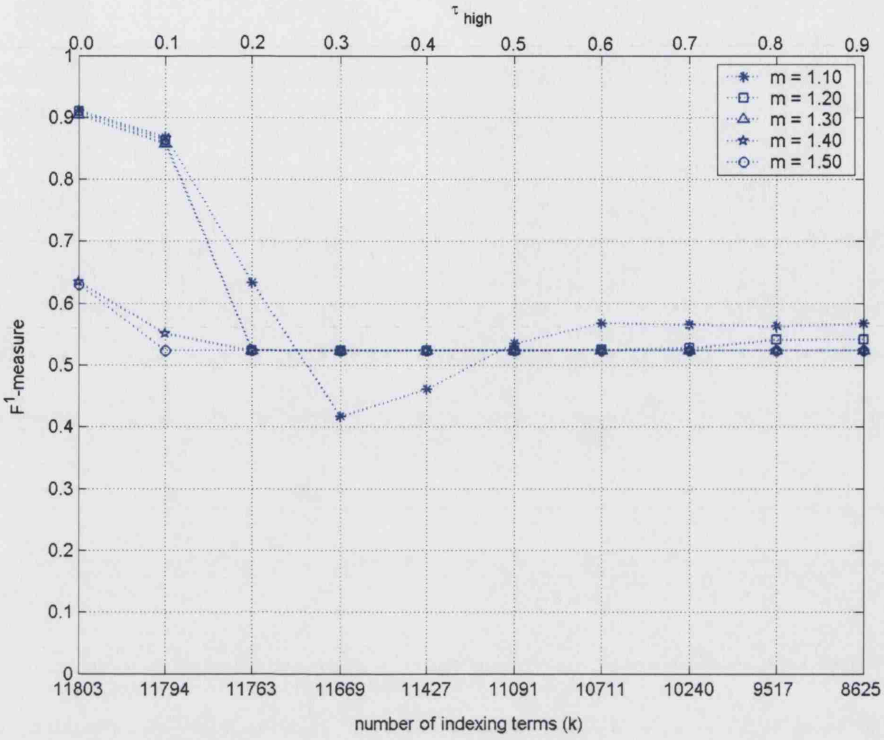


Figure 4.31: Impact of the high specificity filter on the external performance of the H-FCM for the INSPEC collection (average F^1-measure vs. number of indexing terms).

This observation suggests that among the discarded terms are those which describe the main topics of the collections and that serve to tie the documents together into the right clusters. To verify this assumption, we compare the most influential terms in the cluster centroids before and after applying the τ_{high}=0.20 pre-processing filter. To make the analysis we take for example the m value which generally led to the best performances (*i.e.* m=1.10). In Table 4.5, we show the top ten weights and the respective terms of each cluster centroid when no pre-processing filter had been applied to the document vectors. Moreover, the terms which have been removed by the τ_{high}=0.20 filter are shown in bold.

As mentioned in section 3.3.1, it is usually assumed in the literature that both low- and high-frequency terms are non-significant for describing the relevant concepts covered by the documents and thus they can be discarded [52]. In fact, terms appearing frequently in many documents are generally assumed to have little discrimination power for IR purposes. However, when the document collections contain an underlying clustering structure and the objective is to find that structure, it is expected that some of the high frequency terms are those that represent the main topics of each cluster. More so when the documents in the collection are not very long, like the ODP or INSPEC ones.

Table 4.5: Top ten weighted terms in the H-FCM cluster centroids (for m=1.10), without pre-processing the document vectors.

Collection	Cluster centroids				
REUTERS1 (3 clusters)	(0.55) **mln**	(0.35) **cts**	(0.35) **net**	(0.30) loss	(0.27) **dlrs**
	(0.22) **shr**	(0.19) ***reuter***	(0.17) ***march***	(0.17) ***year***	(0.15) revs
	(0.43) ***march***	(0.36) ***reuter***	(0.32) **cts**	(0.23) **april**	(0.21) **company**
	(0.19) **corp**	(0.18) **dlrs**	(0.17) record	(0.16) **pct**	(0.15) shares
	(0.51) trade	(0.47) blah	(0.20) japan	(0.18) ***reuter***	(0.18) ***march***
	(0.18) **dlrs**	(0.17) billion	(0.13) **mln**	(0.13) ***year***	(0.12) **pct**
REUTERS2 (5 clusters)	(0.64) **oil**	(0.23) ***march***	(0.21) ***reuter***	(0.17) **dlrs**	(0.16) crude
	(0.16) **mln**	(0.15) opec	(0.14) prices	(0.11) **pct**	(0.10) bpd
	(0.49) **pct**	(0.30) **rate**	(0.30) **bank**	(0.22) ***reuter***	(0.21) ***march***
	(0.20) **market**	(0.17) **billion**	(0.17) rates	(0.17) fed	(0.13) federal
	(0.52) **mln**	(0.48) stg	(0.33) **bank**	(0.27) **market**	(0.23) money
	(0.15) ***reuter***	(0.15) **pct**	(0.15) ***march***	(0.13) **today**	(0.13) england
	(0.91) blah	(0.15) pct	(0.13) **rate**	(0.12) fed	(0.11) bank
	(0.09) **trade**	(0.08) **billion**	(0.07) sets	(0.07) repurchase	(0.06) customer
	(0.59) **trade**	(0.28) japan	(0.19) ***reuter***	(0.18) ***march***	(0.16) **billion**
	(0.14) japanese	(0.13) ***year***	(0.12) washington	(0.12) countries	(0.11) told
ODP (5 clusters)	(0.88) **game**	(0.24) ***kid***	(0.13) ***top***	(0.13) ***teen***	(0.13) review
	(0.13) comput	(0.12) inform	(0.10) video	(0.08) includ	(0.06) **sport**
	(0.93) **lego**	(0.12) town	(0.09) ***kid***	(0.09) **toi**	(0.08) citi
	(0.08) **sport**	(0.08) **hobbi**	(0.08) ***top***	(0.08) ***teen***	(0.08) creation
	(0.92) math	(0.16) ***kid***	(0.11) school	(0.10) **game**	(0.10) time
	(0.09) ***teen***	(0.09) ***top***	(0.08) problem	(0.07) includ	(0.07) algebra
	(0.80) safeti	(0.42) ***kid***	(0.26) fire	(0.11) **game**	(0.10) prevent
	(0.10) tip	(0.09) safe	(0.09) ***teen***	(0.08) activ	(0.07) learn
	(0.81) **sport**	(0.35) hockei	(0.21) ***kid***	(0.15) **game**	(0.11) histori
	(0.11) inform	(0.11) ***top***	(0.10) ***teen***	(0.09) **hobbi**	(0.09) soccer
INSPEC (3 clusters)	(0.63) **network**	(0.49) **neural**	(0.18) **algorithm**	(0.17) ***model***	(0.15) **system**
	(0.14) ***method***	(0.13) **learn**	(0.13) **base**	(0.12) **train**	(0.10) **backpropag**
	(0.74) **control**	(0.49) **fuzzi**	(0.27) **system**	(0.10) **base**	(0.10) ***model***
	(0.08) **adapt**	(0.08) **design**	(0.08) **logic**	(0.07) ***method***	(0.07) **propos**
	(0.72) **cluster**	(0.29) **algorithm**	(0.25) **data**	(0.17) ***method***	(0.17) imag
	(0.17) **base**	(0.16) **fuzzi**	(0.12) ***model***	(0.09) **propos**	(0.09) **result**

Table 4.5 presents the top ten weighted terms of the H-FCM cluster centroids. The terms are dimensions of the centroid vectors and the weights are the values of those dimensions. This table contains examples of both kinds of high frequency terms: those with little discrimination power (shown in italic) and those representing the clusters topics. The term 'reuter' in both REUTERS collections, the terms 'kid' and 'teen' in the ODP

collection and the term 'method' in the INSPEC collection, are examples of terms that appear in all clusters centroids and consequently, are not good representatives of the clusters contents. These terms have been correctly discarded by the τ_{high}=0.20 filter as noise. But terms like 'game', 'lego' or 'sport' in the ODP case or 'neural', 'control' or 'fuzzi' in the INSPEC case, correspond to known topics in the document collections (see Table 3.2). Their elimination has led to the poor performances verified in the results. Therefore, using high specificity pre-processing filters presents no advantages for our particular application.

4.4.3 Considerations about the TF-IDF weighting scheme

In this section, we justify the poor clustering performances obtained with TF-IDF document vectors in section 4.3 in light of the pre-processing results presented above. The motivation behind this weighting scheme, which originated in the field of IR [59], is the fact that terms occurring frequently are usually poor in discriminating a document from the remainder of the documents in the collection. With this scheme, the frequency of each term (TF) is emphasised or de-emphasised according to its Inverse Document Frequency (IDF), or its specificity. From equations (2.3)and (4.62) it follows that,

$$x_{ij} = f_{ij} \cdot SP_j \tag{4.64}$$

Figures 4.32 to 4.35 show for the different document collections, respectively, by how much the 50 terms with the lowest specificity contribute to the total length of the document vectors when the TF and TF-IDF weighting schemes are used. The contribution of a given term to the length of each document is calculated and then averaged over all documents. The plots show the average contribution.

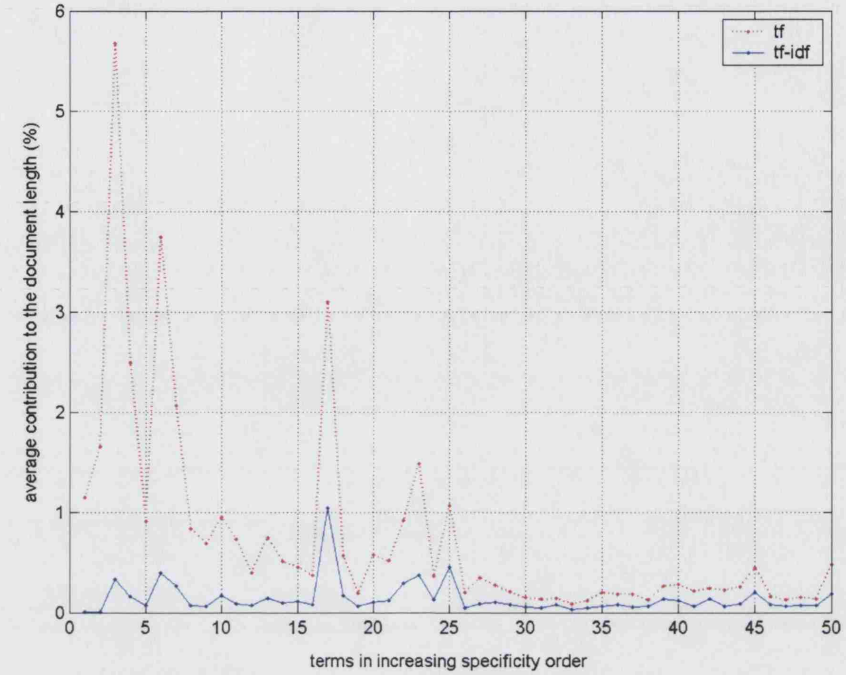


Figure 4.32: Contribution of the TF and TF-IDF weights of the 50 less specific terms to the length of the document vectors (averaged over all documents) for the REUTERS1 collection.

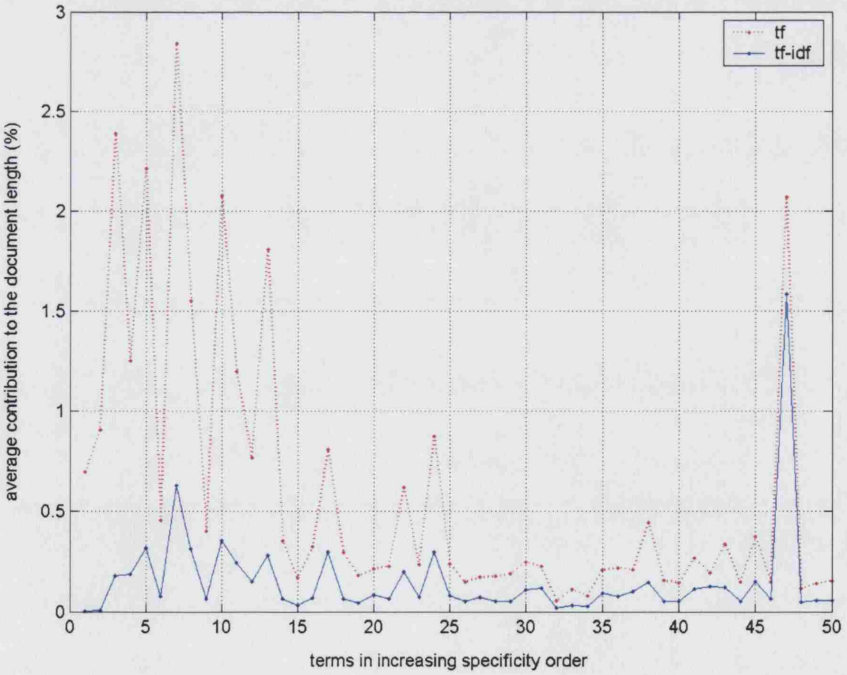


Figure 4.33: Contribution of the TF and TF-IDF weights of the 50 less specific terms to the length of the document vectors (averaged over all documents) for the REUTERS2 collection.

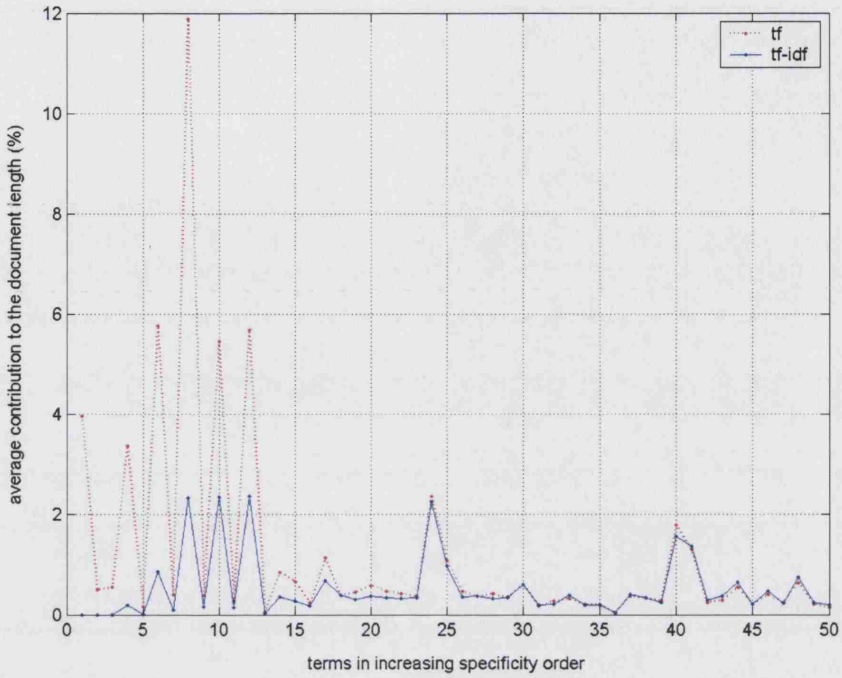


Figure 4.34: Contribution of the TF and TF-IDF weights of the 50 less specific terms to the length of the document vectors (averaged over all documents) for the ODP collection.

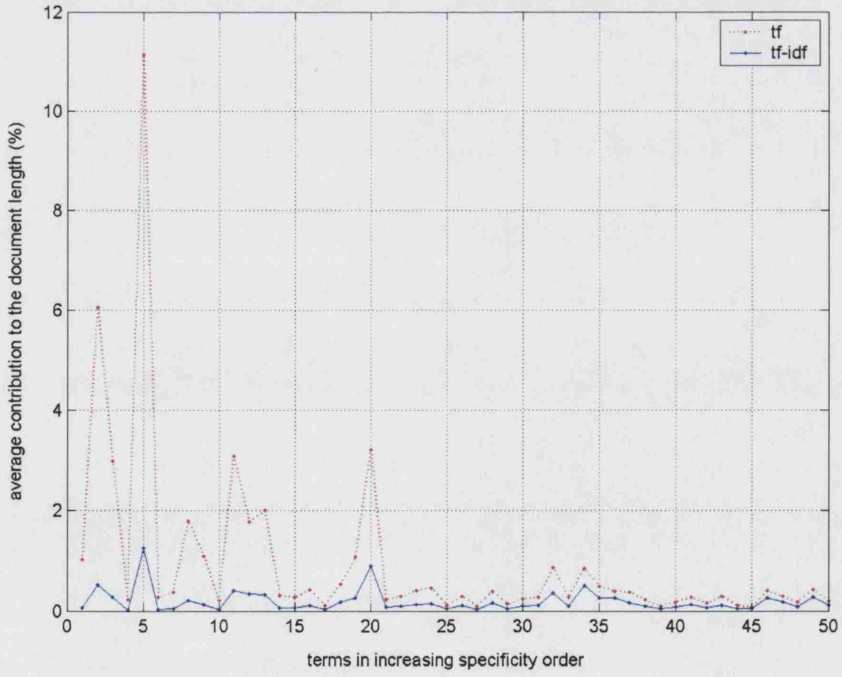


Figure 4.35: Contribution of the TF and TF-IDF weights of the 50 less specific terms to the length of the document vectors (averaged over all documents) for the INSPEC collection.

Table 4.6: The 10 less specific terms and the variation (Δ%) of their contribution to the length of the document vectors when TF-IDF weights are used instead of TF weights.

	REUTERS1									
Terms	reuter	march	**mln**	**dlrs**	year	**cts**	**net**	comp	any	corp
Δ%	-1.15	-1.66	**-5.35**	**-2.33**	-0.84	**-3.35**	**-1.80**	-0.77	-0.63	-0.79
	REUTERS2									
Terms	reuter	march	**pct**	market	**bank**	today	**mln**	**billion**	year	**trade**
Δ%	-0.70	-0.90	**-2.21**	-1.07	**-1.89**	-0.38	**-2.21**	**-1.24**	-0.34	**-1.72**
	ODP									
Terms	kid	teen	top	**sport**	hobbi	**game**	toi	**lego**	school	**safeti**
Δ%	-3.96	-0.51	-0.55	**-3.20**	-0.15	**-4.92**	-0.32	**-9.55**	-0.24	**-3.11**
	INSPEC									
Terms	base	**fuzzi**	**system**	paper	**control**	result	propos	**algorithm**	method	pesent
Δ%	-0.96	**-5.55**	**-2.72**	-0.21	**-9.89**	-0.25	-0.34	**-1.58**	-0.98	-0.18

To further analyse this effect, the 10 terms with the lowest specificity are presented in Table 4.6 and the reduction of their importance in the documents representation due to TF-IDF weighting is quantified. It can be observed that the most de-emphasised terms (shown in bold) are in fact good terms for clustering purposes. In the previous sub-section we have indeed verified that discarding these terms impacts significantly on the clustering performance. For the same reason, de-emphasising the weights of such good terms, while keeping all the terms in the collection, leads to poorer performances of the clustering algorithm. This justifies the poor results presented in sections 4.3.1 and 4.3.2 obtained with the TF-IDF data.

4.5 Performance of H-FCM with Jaccard and overlap similarity coefficients

In the previous chapter, we have developed the H-FCM algorithm for clustering unit length data vectors based on similarity measures (see section 3.5). From the analysis we have made in section 3.3 regarding the behaviour of the similarity coefficients, we have concluded that the cosine coefficient was the most suitable choice for document clustering

as it produced the highest pairwise similarity values between high-dimensional vectors when compared to other similarity measures. We have also observed that the overlap coefficient was only suitable for clustering low-dimensional data sets.

The results from the previous H-FCM experiments have all been obtained with the cosine coefficient. In this section, we analyse the performance of H-FCM with Jaccard and overlap coefficients and compare it with the cosine performance. Figures 4.36 to 4.39 present the results of this comparison, obtained for the REUTERS1, REUTERS2, ODP and INSPEC collections, respectively, without pre-processing the document vectors.

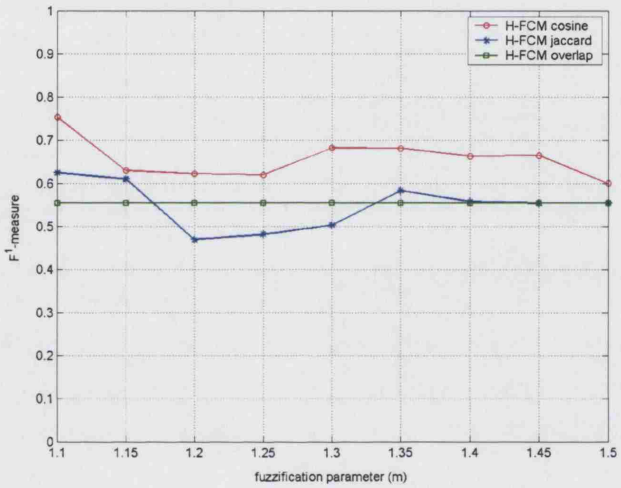


Figure 4.36: Average F^1-measure for the REUTERS1 document collection obtained with H-FCM using the cosine, Jaccard and overlap similarity coefficients.

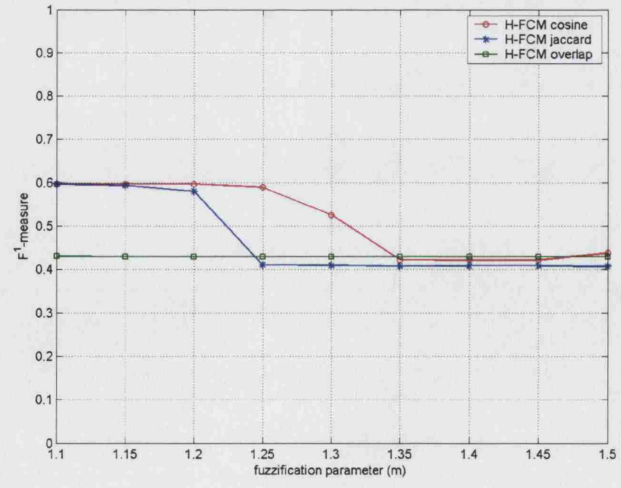


Figure 4.37: Average F^1-measure for the REUTERS2 document collection obtained with H-FCM using the cosine, Jaccard and overlap similarity coefficients.

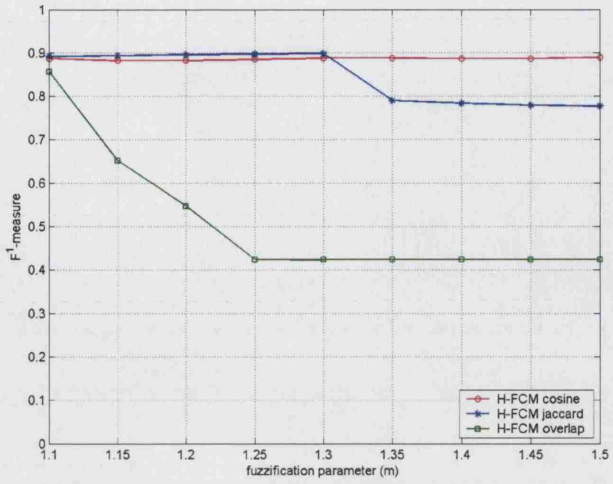


Figure 4.38: Average F^1-measure for the ODP document collection obtained with H-FCM using the cosine, Jaccard and overlap similarity coefficients.

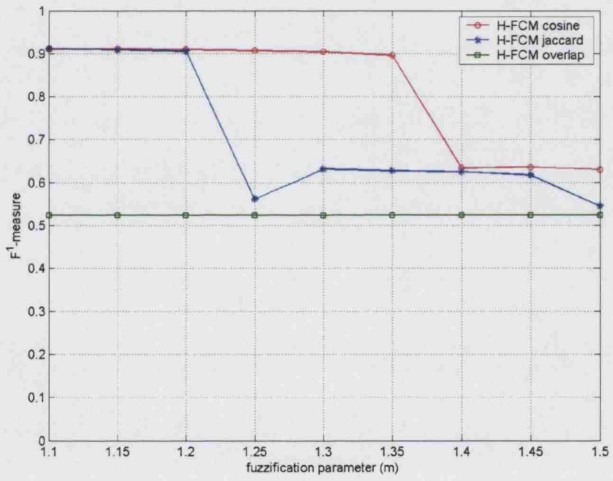


Figure 4.39: Average F^1-measure for the INSPEC document collection obtained with H-FCM using the cosine, Jaccard and overlap similarity coefficients.

The plots show the values of the average F^1-measure for increasing values of the fuzzification parameter m, obtained with the H-FCM algorithm with the cosine, Jaccard and overlap similarity coefficients. The experiments have only been carried out with TF data vectors due to the poor performance of the TF-IDF scheme.

It can be observed that for low values of m ($m \leq 1.20$ in the REUTERS2 and INSPEC cases and $m \leq 1.30$ in the ODP case) the external performance of H-FCM with the Jaccard coefficient is comparable to its performance with the cosine coefficient, except in the

REUTERS1 case where the Jaccard results are always worse. However, for higher m values the performance of H-FCM with the Jaccard coefficient is worse for all document collections. It can also be observed that when the overlap coefficient is used the H-FCM algorithm fails to find a clustering structure for m>1.20 in the ODP case and for all m values in the REUTERS1, REUTERS2 and INSPEC cases. We note that the values of the F^1-measure correspond to the case of maximum fuzziness of the partition matrix (see Table 4.2).

The differences in behaviour of the similarity coefficients in sparse high-dimensional spaces justify the current results. Such behaviour has been previously analysed in section 3.3.2. Regardless of the similarity coefficient, we have shown that when data vectors are less sparse, higher similarity patterns are achieved (see Figure 3.2). Furthermore, in section 4.3 we have shown that the higher the similarity between document vectors and cluster centroids, the higher the range of m values that can be selected to avoid maximum fuzziness of the resulting partition matrix. Hence, these considerations suggest that if we reduce the dimensionality of the document vectors with low specificity filters, the Jaccard and overlap coefficients may be used with higher m values.

The reasoning behind this assumption is that discarding terms reduces the sparsity of the document vectors and as a result, higher similarity values are obtained. This in turn may improve the performance of H-FCM with Jaccard and overlap coefficients for higher values of m. To verify this, we repeat the analysis of the low specificity filter impact on the performance of H-FCM with these similarity coefficients.

Figures 4.40 to 4.43 present the average F^1-measure obtained as a function of the number of indexing terms kept in each collection after applying the low specificity filter with the thresholds in Table 4.3. This experiment has been carried out with the Jaccard coefficient and it has been repeated by running the algorithm for several values of the fuzzification parameter m. For comparison reasons, the plots also show the results obtained with the cosine coefficient for the m=1.10 case.

The results demonstrate that, like in the cosine case, removing terms which are very specific does not decrease the performance of the algorithm for m values which, without pre-processing, had provided good clustering performances.

The previous assumption about the effects of reducing the sparsity of the document vectors can also be verified. It can be seen that the more terms are eliminated, the broader the range of m values that can be used for the same performance levels (see for *e.g.* m>1.30 in the ODP case). Even in the REUTERS1 case, pre-processing presents some benefits. It can

be seen that with just 159 indexing terms left, the H-FCM performance with Jaccard coefficient when $m \leq 1.20$ is similar to its performance with the cosine measure.

The same experiment has been carried out with the overlap coefficient. It has been verified that the low specificity filter did not improve the H-FCM performance with the REUTERS1, REUTERS2 and INSPEC collections. Hence, the plots containing these results are not presented here. However, the performance of the H-FCM algorithm has improved with the ODP collection, as it can be seen in Figure 4.44. The plots show that with the $\tau_{low}=0.40$ filter, the H-FCM performance with overlap coefficient is similar to its performance with the cosine measure for a broader range of m values ($m \leq 1.40$). Overall, these results suggest that the overlap coefficient is unsuitable for clustering sparse high-dimensional data sets.

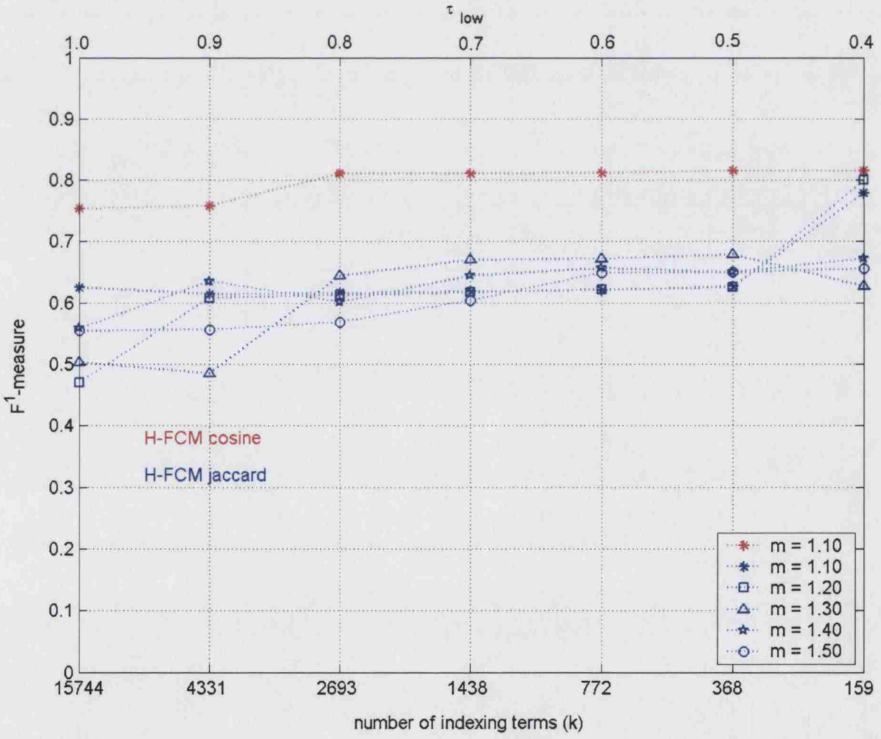


Figure 4.40: Impact of the low specificity filter on the average F^1-measure obtained with H-FCM using the Jaccard coefficient, for the REUTERS1 collection.

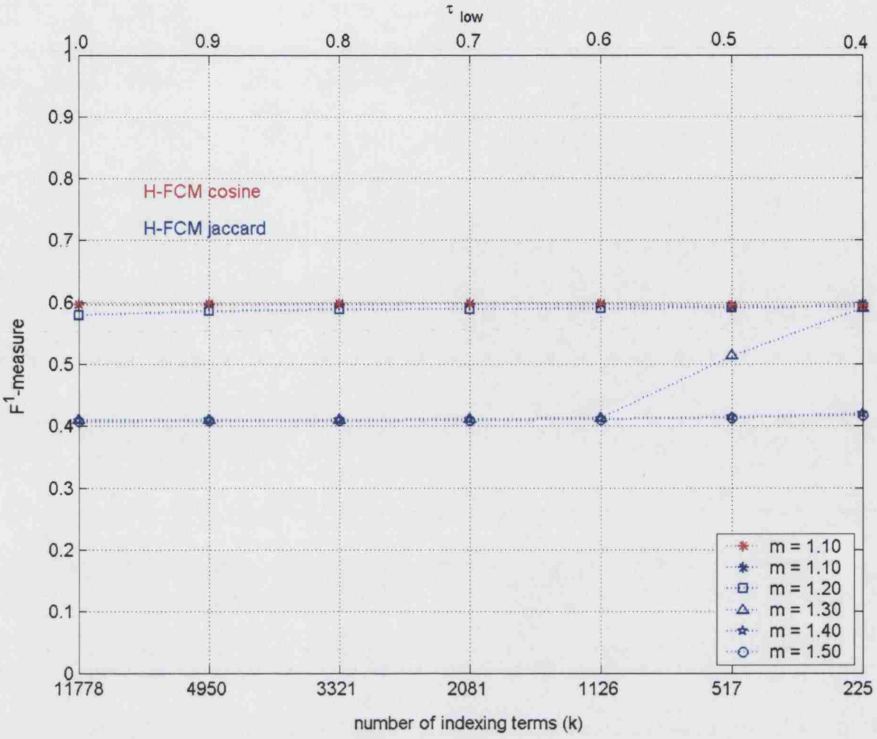


Figure 4.41: Impact of the low specificity filter on the average F^1-measure obtained with H-FCM using the Jaccard coefficient, for the REUTERS2 collection.

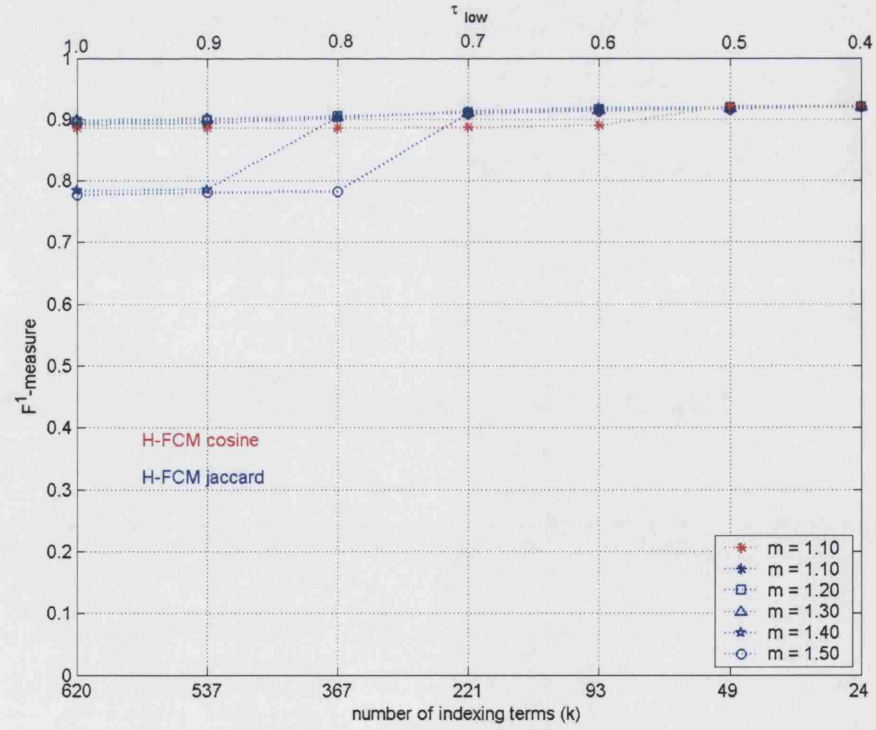


Figure 4.42: Impact of the low specificity filter on the average F^1-measure obtained with H-FCM using the Jaccard coefficient, for the ODP collection.

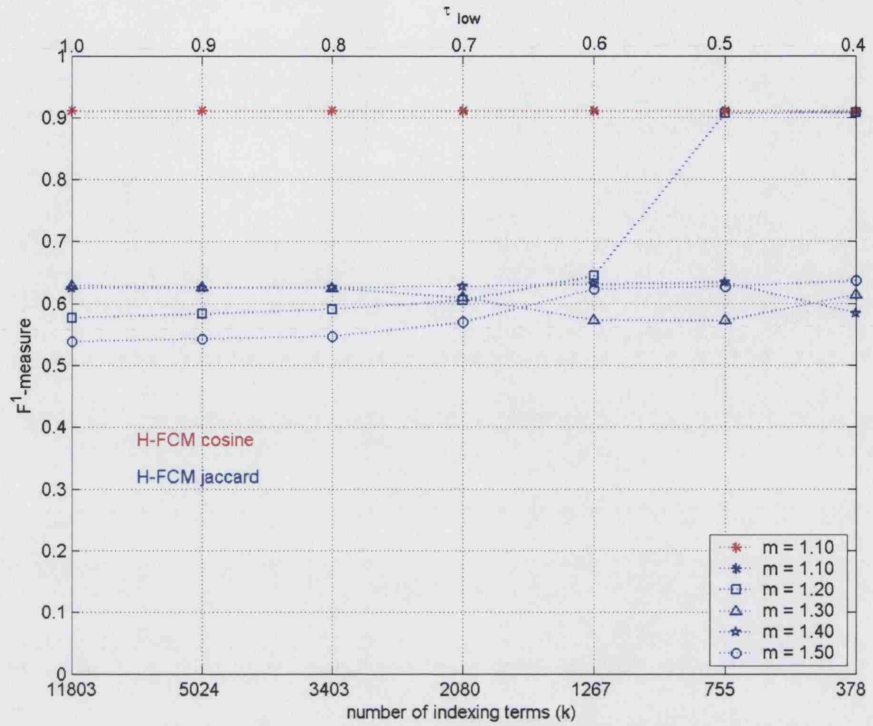


Figure 4.43: Impact of the low specificity filter on the average F^1-measure obtained with H-FCM using the Jaccard coefficient, for the INSPEC collection.

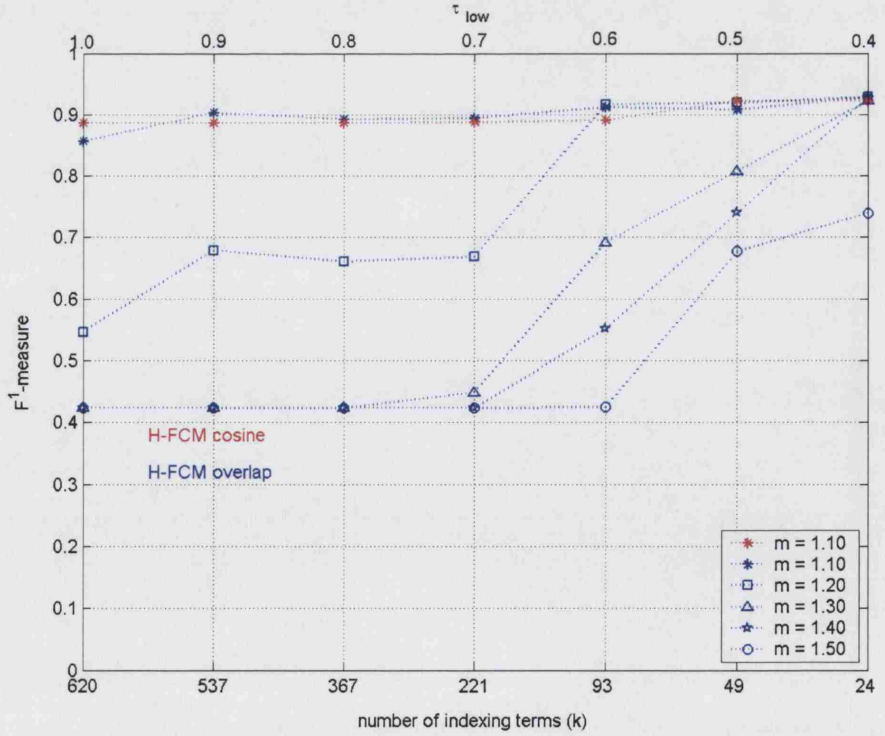


Figure 4.44: Impact of the low specificity filter on the average F^1-measure obtained with H-FCM using the overlap coefficient, for the ODP collection.

To conclude, the cosine coefficient is better than the Jaccard and overlap coefficients for clustering sparse high-dimensional document vectors with the H-FCM algorithm and it allows the use of a broader range of the fuzzification parameter. However, the Jaccard performance approaches the cosine performance when the dimensionality of the problem space is reduced through low specificity pre-processing filters.

4.6 Comparison between H-FCM and traditional hard clustering methods

In this section, we compare the performance of the H-FCM algorithm to that of traditional hard clustering methods in terms of the quality of the discovered document clusters. The main goal of these experiments is to establish whether there are any benefits in using the H-FCM algorithm instead of hard clustering algorithms including the *k*-Means algorithm [79] or agglomerative hierarchical methods [20, 27], which have long been applied for document clustering (see Chapter 2 for description of these methods).

Like with the H-FCM, the hard *k*-Means requires the definition of the final number of clusters *c*. This number has been selected to match the number of reference clusters in each collection (c_{REUTERS1}=3, c_{REUTERS2}=5, c_{ODP}=5 and c_{INSPEC}=3). In the case of the hierarchical methods, the cluster merging process has been stopped when the number of clusters in the hierarchy matched the reference number of clusters. The results with two linkage methods are analysed: Complete-Link (CL) and Group-Average (GA). We have also applied the Single-Link (SL) method but we have verified that this method performed very poorly with all collections, as it generated one very large cluster containing most of the documents while the remaining clusters were almost empty.

Here we analyse the performance of the algorithms for the test document collections after pre-processing the document vectors with the low specificity filter τ_{low}=0.40 and encoding each document with the TF weighting scheme. We note that this pre-processing filter has provided the best performance of the H-FCM for a broader range of values of the fuzzification parameter *m*. In Figures 4.45 to 4.48, it can be verified that the hard clustering methods are fairly insensitive to low specificity pre-processing. It can also be observed that pre-processing with τ_{low}=0.40 generally maintains or improves the algorithms performance when compared to the non pre-processing case.

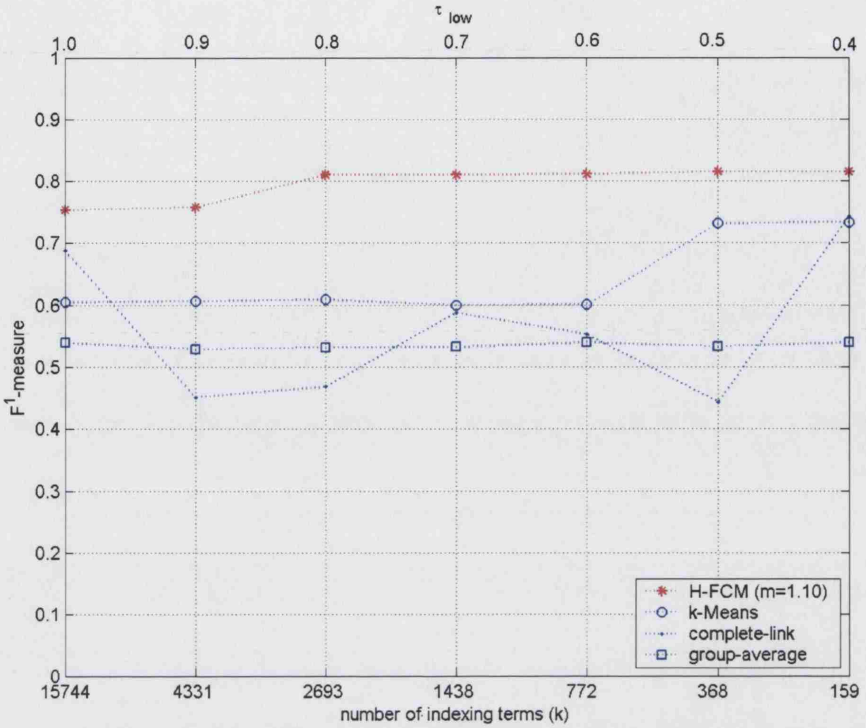


Figure 4.45: Impact of the low specificity filter on the average F^1-measure obtained with hard clustering methods, for the REUTERS1 collection.

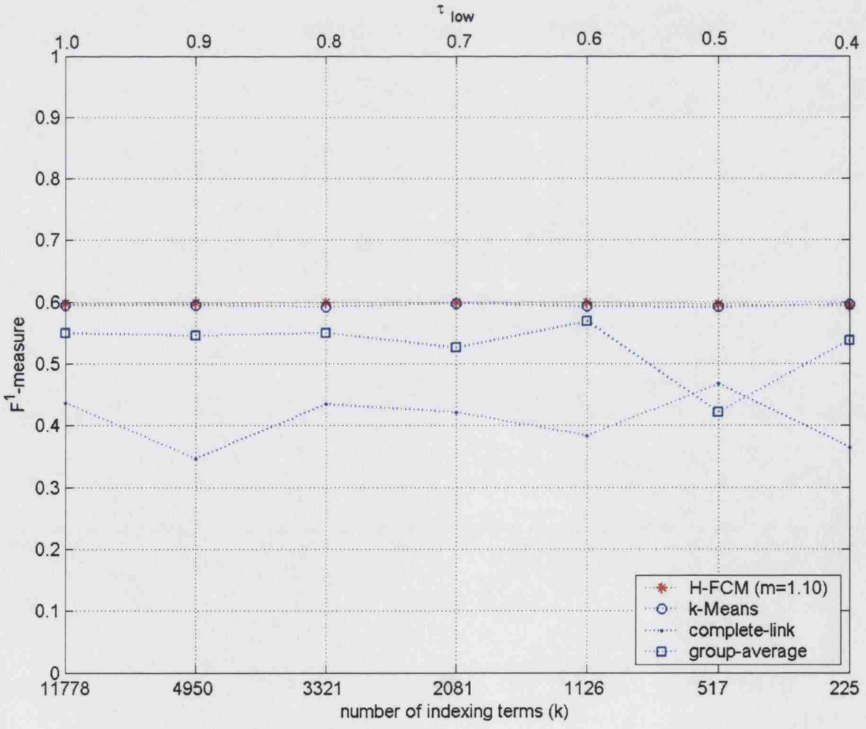


Figure 4.46: Impact of the low specificity filter on the average F^1-measure obtained with hard clustering methods, for the REUTERS2 collection.

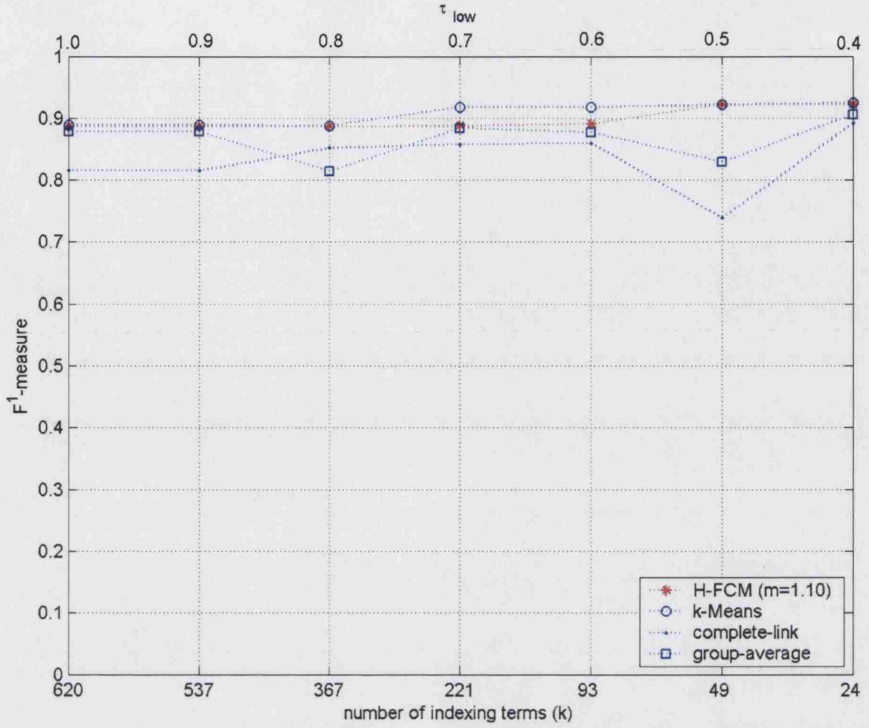


Figure 4.47: Impact of the low specificity filter on the average F^1-measure obtained with hard clustering methods, for the ODP collection.

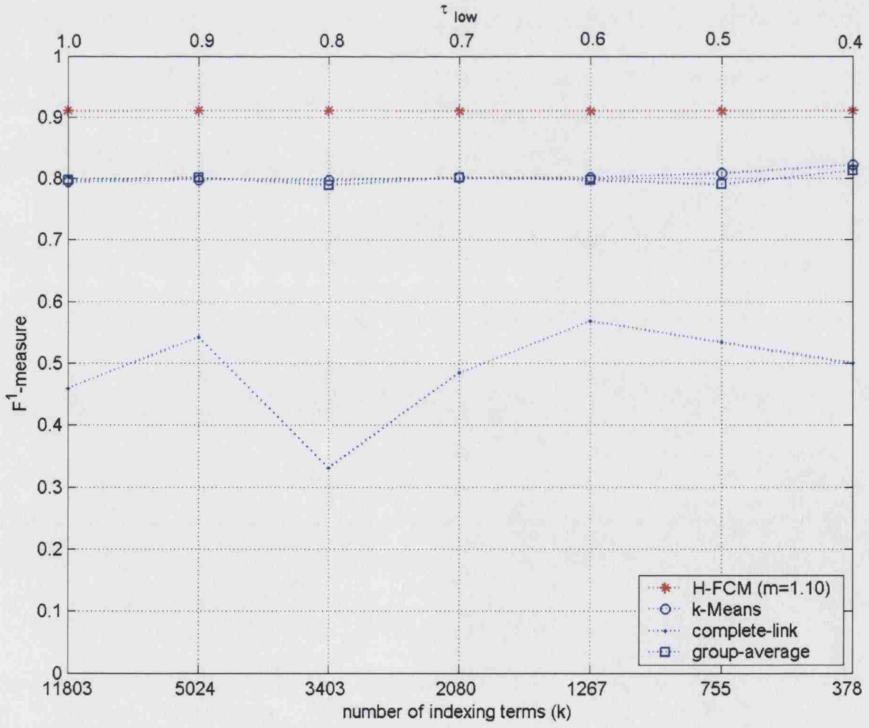


Figure 4.48: Impact of the low specificity filter on the average F^1-measure obtained with hard clustering methods, for the INSPEC collection.

Figures 4.49 to 4.52 compare the performance of the H-FCM with that of *k*-Means, CL and GA algorithms for REUTERS1, REUTERS2, ODP and INSPEC collections, respectively. The plots show the values of average clustering precision against average recall, obtained with each clustering algorithm, as well as the corresponding F^1-measure. As mentioned in section 4.2.2, fuzzy clusters obtained with H-FCM are made crisp before calculating precision and recall. In the experiments from previous sections we have hardened the fuzzy clusters based on the maximum membership criterion, thus attributing each document to a single cluster. For the present comparison it is relevant to consider the case when documents are assigned to several clusters simultaneously. Thus, we analyse the H-FCM performance (m=1.10) for several α-cuts of the partition matrix, *i.e.* documents with membership value in a given cluster higher than a threshold α are attributed to that cluster. The α-cut that maximises the F^1-measure and the F^1-measure value previously obtained with maximum membership hardening are also indicated in the plots.

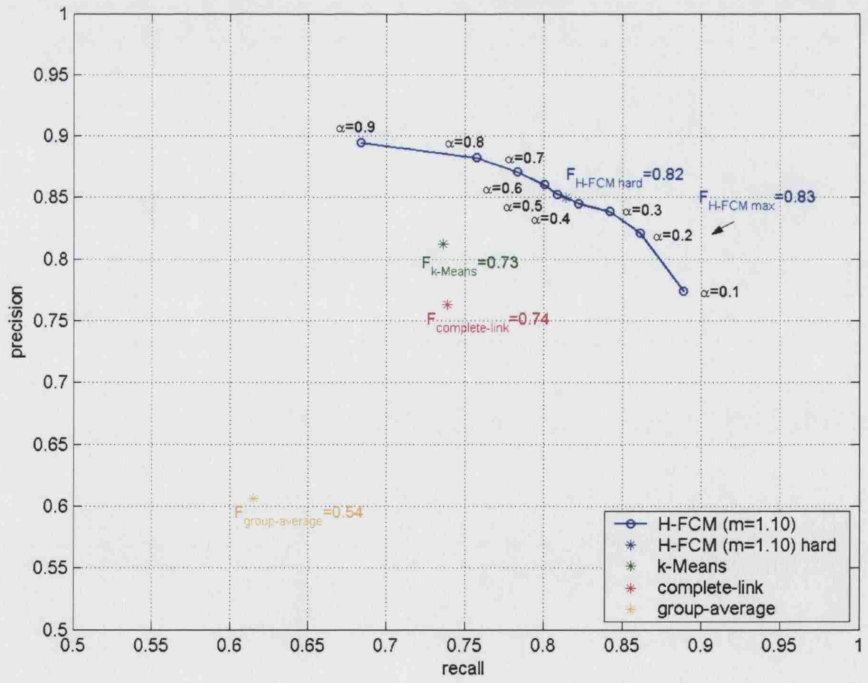


Figure 4.49: Average precision vs. recall obtained with H-FCM (m=1.10), *k*-Means, CL and GA methods, for the REUTERS1 collection.

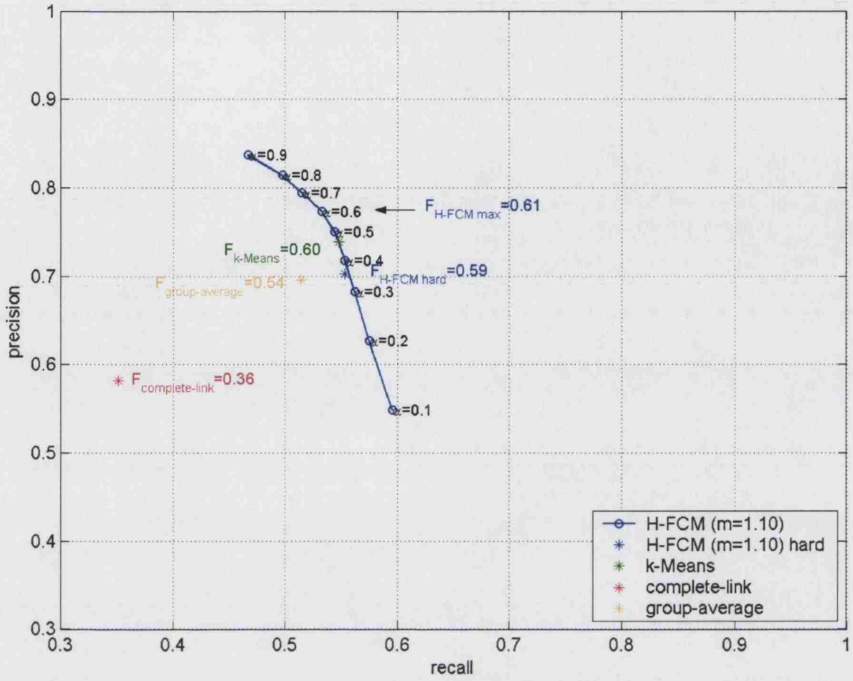


Figure 4.50: Average precision vs. recall obtained with H-FCM (m=1.10), k-Means, CL and GA methods, for the REUTERS2 collection.

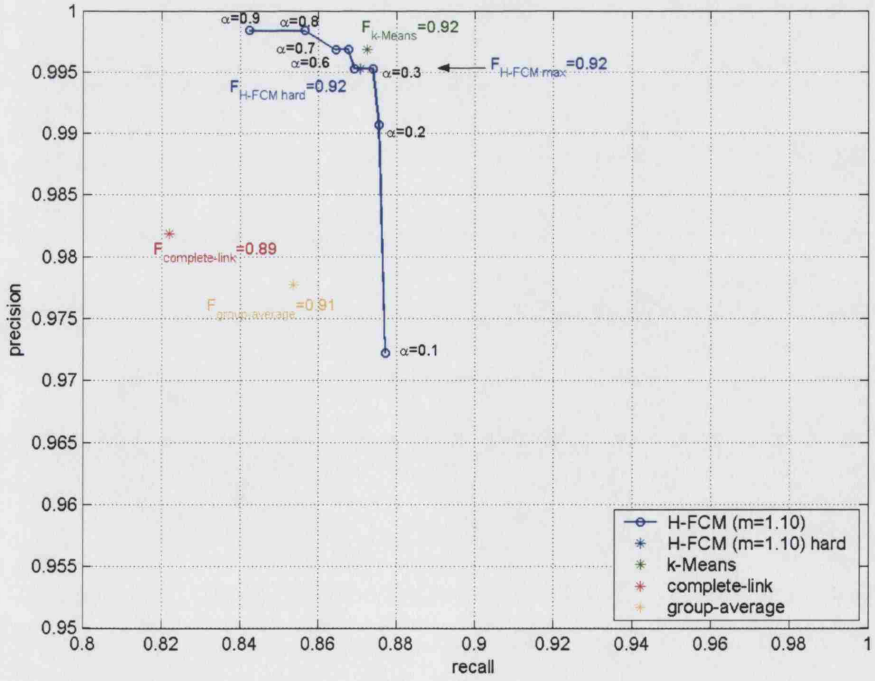


Figure 4.51: Average precision vs. recall obtained with H-FCM (m=1.10), k-Means, CL and GA methods, for the ODP collection.

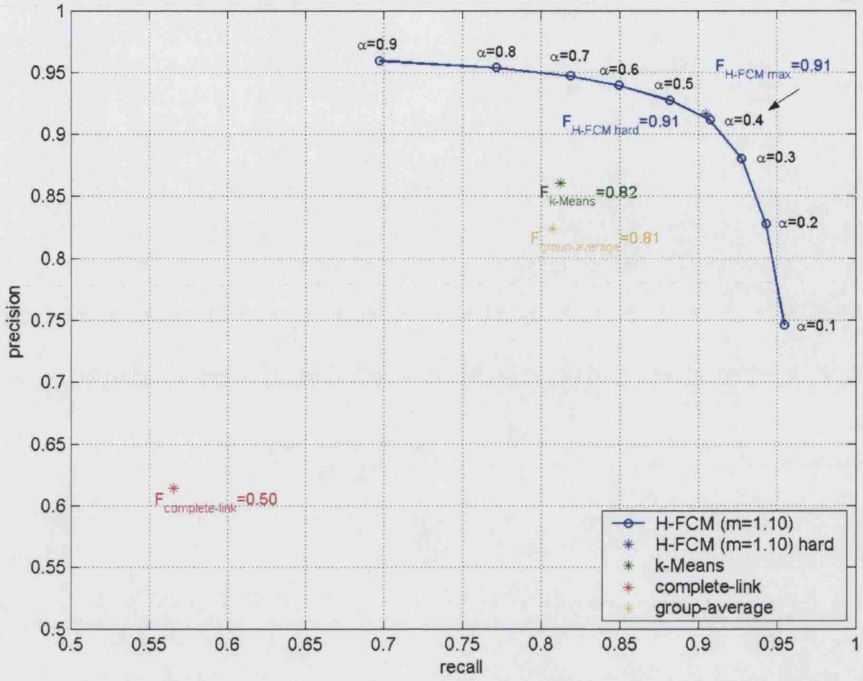


Figure 4.52: Average precision vs. recall obtained with H-FCM (m=1.10), k-Means, CL and GA methods, for the INSPEC collection.

It can be observed that with all document collections the H-FCM algorithm clearly outperforms the two agglomerative hierarchical methods. For the same level of clustering precision the H-FCM has always achieved higher recall than the CL and GA algorithms. Comparing its performance with the k-Means performance, it can be seen that for the α-cut that leads to the best compromise between precision and recall, the F^1-measure obtained with H-FCM is at least similar and in two of the cases much better (REUTERS1 and INSPEC collections) than in the k-Means case. In cases where the H-FCM performance is similar to the k-Means (REUTERS2 and ODP collections), the use of this algorithm may present advantages over the H-FCM because it is simpler and it does not require the selection of an m value. As shown before, such selection may impact strongly on the H-FCM outcome.

A great advantage of the H-FCM is that precision and recall can be controlled by setting different thresholds for α. It is obvious that lowering the threshold will lead to more documents being attributed simultaneously to more clusters, hence increasing recall and decreasing precision. This allows the discovery of associations between documents that contain loosely related topics, which is particularly important in the *e*-Learning context for flexible exploration of learning material.

Furthermore, the H-FCM algorithm provides the degree of membership of each document in the clusters. This enables to sort the documents according to their relevance to the set of topics conveyed by the clusters centroids. As mentioned previously, for low values of the fuzzification parameter the fuzzy clustering results tend to the hard case and consequently, the membership values of the documents in a given cluster tend to 1. We note that the results above have been obtained with m=1.10. However, in section 4.4.2 we have shown that with pre-processing the H-FCM performance for higher m values was similar to the m=1.10 case. Therefore, to make the sorting more effective higher m values can be used with H-FCM, as the documents membership within a cluster become more differentiated from each other.

4.7 Summary

In this chapter we have applied the new H-FCM algorithm for clustering document collections and we have quantified its performance through internal and external validity measures. The chapter started by comparing the H-FCM with the original FCM algorithm. We have demonstrated that the H-FCM performs much better than the FCM, with all test document collections. The new method consistently leads to a better compromise between clustering precision and recall. Furthermore, we have verified that unlike the FCM method, the H-FCM is able to find good clusters for a wider range of the fuzzification parameter m. Through these experiments, we have also empirically determined a suitable range for the m parameter, which is generally low for document clustering.

A comparison between TF and TF-IDF term weighting schemes has also been carried out. Most document clustering applications apply the TF-IDF scheme, however, our investigation has shown that the TF-IDF scheme leads to worse clustering performances than the TF scheme with either of the algorithms. The results obtained in this experiment are coherent with the results from the clustering tendency tests carried out in the previous chapter. We have investigated further these differences in performance and we have concluded that the TF-IDF scheme substantially de-emphasises the weights of terms representing the reference class topics. This leads to lower intra-class similarities and consequently, the clustering algorithm fails to discover the reference document clusters.

The impact of pre-processing the document vector representations through term frequency thresholding has also been examined. Term specificity filters have been applied

to all document collections. We have shown that discarding common terms has a negative impact on the clustering performance and we have concluded that such decrease in performance was due to the elimination of high-frequency terms that represented known topics in the document collections. On the other hand, we have shown that discarding very specific terms (*i.e.* that only appear in very few documents) leads to huge dimensionality reductions while maintaining or even improving the clustering performance, which is extremely beneficial in terms of computational costs.

Our experiments also included a comparison between cosine, Jaccard and overlap similarity coefficients. We have verified that the H-FCM algorithm generally performs better with the cosine measure and that a broader range of the fuzzification parameter can be used with this coefficient. We have also demonstrated that the different behaviour of the similarity coefficients in sparse high-dimensional spaces justifies the differences in the clustering results. Our investigation has shown that reducing the sparsity of the document vectors by filtering out terms results in an increased performance of the Jaccard coefficient, comparable to the cosine coefficient. However, similar increase in performance has not been verified with the overlap coefficient, which confirms that this similarity measure is unsuitable for clustering sparse high-dimensional data sets.

In the last clustering experiment we have compared the H-FCM algorithm with hard clustering methods. Our results have shown that the H-FCM algorithm clearly outperforms the CL and GA agglomerative hierarchical methods and that it generally performs better or similarly to the *k*-Means algorithm. We have also observed that a threshold on the H-FCM fuzzy cluster memberships could be set to control clustering precision and recall, which could be relevant for flexible exploration of *e*-Learning material.

In the next chapter, tools that integrate the H-FCM algorithm for clustering and browsing *e*-Learning material are developed and evaluated in an end-to-end *e*-Learning system.

Chapter 5
Knowledge Navigator for *e*-Learning material

5.1 Introduction

In the previous chapter, we have evaluated the performance of the H-FCM algorithm for discovering meaningful content relationships in document collections. In this chapter, we develop a prototype tool for representing and browsing the fuzzy knowledge space discovered by the document clustering process and we integrate this tool in an *e*-Learning environment.

In section 5.2, the CANDLE project, which has provided the context for the research work presented in this thesis, is described. In section 5.3, the representation of the fuzzy knowledge space is addressed with an overview of suitable knowledge representation technologies for Web based applications. One of such technologies is the Topic Maps standard, which we have chosen for our application. Section 5.4, provides the details of the use of this standard and also describes a tool we have developed for dynamically generating a topic map representing the fuzzy knowledge space. In section 5.5, we address the design and implementation of a search and navigation tool - the Knowledge Navigator - for flexible access to *e*-Learning material and we describe its functionality in detail. In section 5.6, the deployment of the Knowledge Navigator in the CANDLE *e*-Learning system is discussed and the results of user trials are presented and analysed. Finally, the main contributions of this chapter are summarised in section 5.7.

5.2 The CANDLE project

CANDLE (Collaborative And Networked Distributed Learning Environment) [3] is a cross European *e*-Learning project recently finished under the IST (Information Society Technologies) program of the European Commission [130]. CANDLE emerged to promote the collaboration among content authors of different educational and corporate institutions working in the area of telematics through the sharing and reuse of learning material, enabling rapid development and deployment of new open courseware [4]. The principal objectives of the project were, on the educational side, to increase the flexibility of the learning process to accommodate various pedagogical models and to enable flexible usages of learning material, and on the technical side, to develop a methodology and a set of guidelines for the creation of reusable learning material. CANDLE has developed an *e*-Learning system with tools for authoring, maintaining, evolving, administering and brokering of network-based multimedia courses and for online access and navigation of learning material.

Often the authors of learning material have in mind specific ways of a learner using that material (*e.g.* following a set of lecture notes and then doing a test, or using relevant material in problem-based learning tasks). The way the learner is expected to use the material constitutes an explicit or implicit pedagogical model. The CANDLE project has developed a methodology for modularisation and tagging of learning material, which enables the reuse of material to support a variety of pedagogical models. Using this approach, the target granularity of the learning material is established, and then the material is tagged with appropriate XML metadata.

The idea of adding structured metadata to Web-based content is the lead idea behind the Semantic Web initiative[6]. This initiative has recently emerged with the aim of specifying open and interoperable technologies for sharing and processing Web data using automated tools [15]. There are a number of standardisation efforts that have defined interoperable metadata models for a wide range of applications, including *e*-Learning. One of the first metadata initiatives was the Dublin Core (DC), which defined a set of 15 basic metadata elements for describing Web resources, known as the Dublin Core Metadata Element Set (DCMES) [131]. This element set is very generic in nature and has clear semantics, which

[6] Semantic Web activity: http://www.w3.org/2001/sw/

enables simple, yet effective, high-level descriptions of wide range of Web resources. However, the DC model was found too general to cope with the particular demands of *e*-Learning applications. Hence, several educational metadata initiatives have emerged with the purpose of defining appropriate metadata models.

The first metadata initiatives for *e*-Learning were initiated by the IMS (Instructional Management System) project [10], by groups within the NIST (National Institute for Standards and Technology) [132] and also by the IEEE Learning Technology Standards Committee (LTSC) [1]. The NIST effort then merged with the IMS effort and later IMS also began collaborating with the European project ARIADNE [11], which was also actively working towards a suitable educational metadata model. This collaboration led to a joint proposal and specification to the IEEE, which formed the basis for the current IEEE Standard for Learning Object Metadata (LOM) [12].

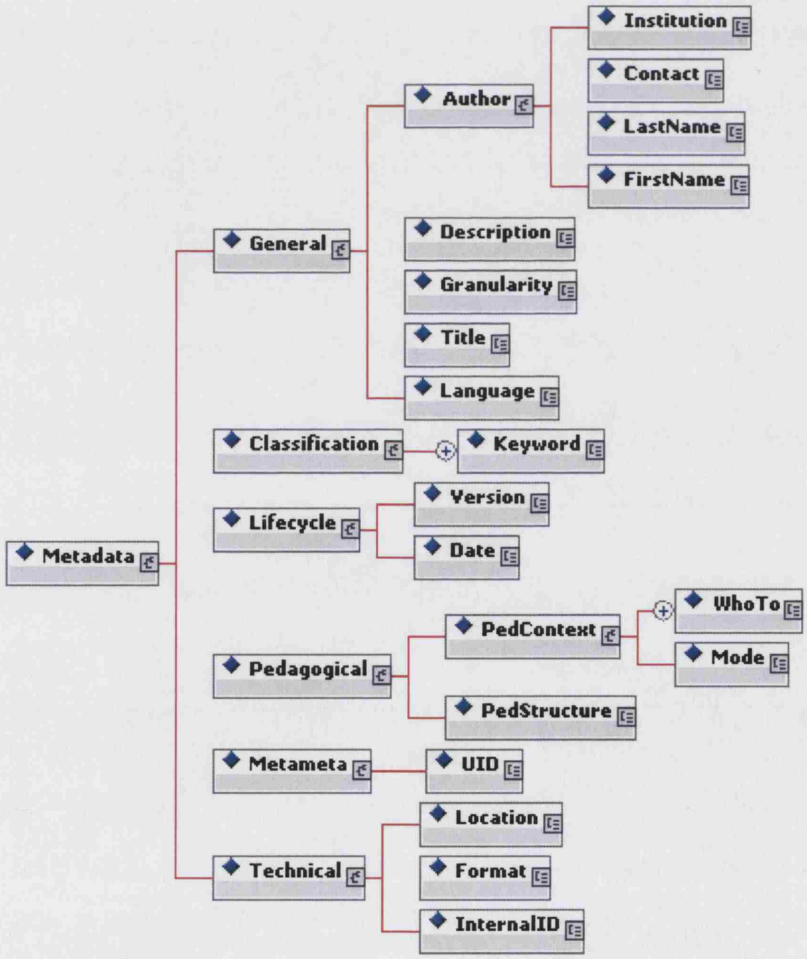


Figure 5.1: The reduced CANDLE metadata schema.

The CANDLE metadata set is an extension of the LOM specification. Its metadata elements capture a number of attributes which describe learning objects, such as, general information (*e.g.* title, author, description), lifecycle information (*e.g.* version), technical information, etc. In particular, the CANDLE metadata model includes details about the envisaged pedagogical approach, as well as classification information, where a number of taxonomic keywords describe the content of the learning material. The reduced CANDLE metadata schema, containing only the mandatory fields, is shown in Figure 5.1.

The classification information as well as some of the generic metadata fields, such as, title and description, enable processes of knowledge discovery to associate related learning material according to its relevance to a particular learning context. The fuzzy clustering process explored in this thesis provides a dynamic knowledge representation framework, which can be used for assisting and enhancing the navigation of the learner in exploratory interactions with learning material.

5.3 Representation of the fuzzy knowledge space

The knowledge-based navigation of *e*-Learning material using the fuzzy knowledge space obtained through the document clustering process is illustrated in Figure 5.2. In this section we address ways of formally representing the content relationships discovered by the fuzzy clustering algorithm to be used in the Knowledge Navigator.

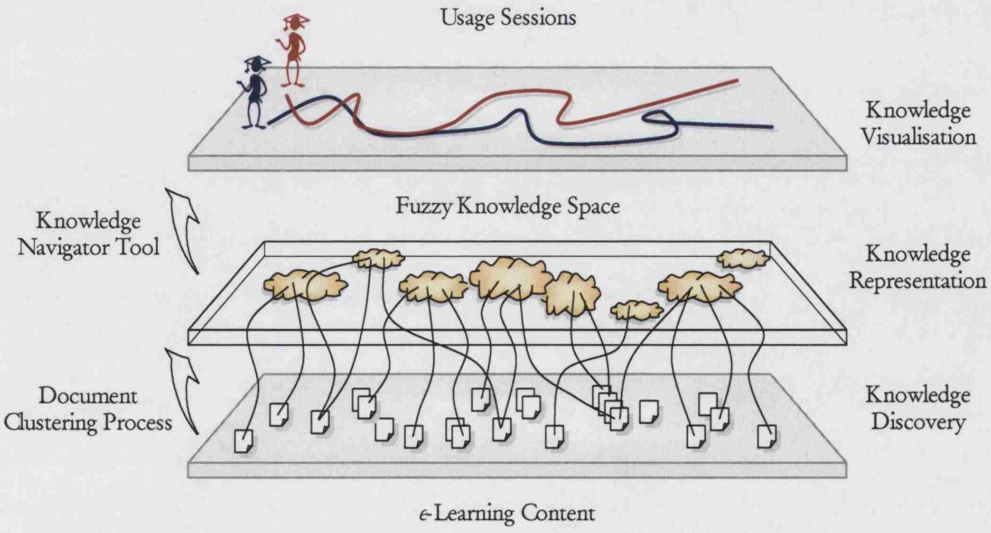


Figure 5.2: Navigation of *e*-Learning material using the fuzzy knowledge space.

5.3.1 Technologies for knowledge representation

Semantic networks [133] have long been used for knowledge representation. These networks are basically labelled directed graphs where the nodes represent concepts and the links between connected nodes, which are directed and labelled, represent the relationship between the nodes. The link labels express the semantics of the concepts relationships. Frames [134] are another example of traditional knowledge representation systems. Each frame represents a concept (or an object) and it stores an identifier and a set of attributes about the concept, each of which is kept in a different frame slot.

Although fairly simple to implement, these traditional knowledge representation approaches present some limitations regarding interoperability. The concepts, relationships and attributes are usually defined for a specific application without clear semantics, which is acceptable in centralised and self-contained systems where everyone shares exactly the same understanding of the concepts. However, these approaches are unsuitable for distributed information systems like those accessible through the World Wide Web.

The representation of the knowledge discovered through the fuzzy clustering process can use similar tools to the ones applied for knowledge representation on the Semantic Web. The Semantic Web architecture provides semantic interoperability through the use of metadata for resource description and it includes automated reasoning capabilities for knowledge sharing and reuse through the definition of ontologies, which describe concepts and relationships between concepts [135].

The W3C's (World Wide Web Consortium) RDF (Resource Description Framework) standard [101] and the XML standard [13] have both been proposed for knowledge representation on the Semantic Web [16]. These two standards present complementary features. RDF has the flexibility to represent any sort of data relationships, but it does not provide mechanisms to transport data between systems. On the other hand, XML is very suitable for interchanging data between systems and so this standard is used to encode and transport RDF.

The RDF data model is quite simple: it consists of *entities* and *statements*. An RDF entity is either a resource, which is represented by URI (Unique Resource Identifier), or a literal, which is simply a string. An RDF statement is a binary relationship between two entities based on three components: subject, predicate and object. A graph representation of RDF statements and an example statement are shown in Figure 1.2.

Figure 5.3: Graph representation of a) RDF statements and b) an example RDF statement.

Recently, the Topic Maps (TM) ISO standard [136] has also been proposed for knowledge representation on the Web and it provides and alternative to RDF. A TM can be defined as a collection of *topics*, which can represent anything, linked together by *associations*. TM and RDF exhibit a number of similarities as they both aim at describing information resources and making it easier to find relevant information. However, the RDF and TM data models present conceptual differences which have raised the issue of compatibility between both standards in the Semantic Web context. This problem has been addressed by Lacher and Decker [137] which have proposed a framework for mapping TM information into RDF information. In fact, Moore [138] has demonstrated that both standards are expressive enough for TM to be modelled in RDF and RDF to be modelled as TM.

An advantage of the TM data model over the RDF data model is the ease with which *n*-ary relationships are handled. In RDF, a relationship involving multiple entities has to be defined through multiple RDF statements since the RDF data model only defines binary associations between entities. Furthermore, the direct and inverse associations between two entities has to be explicitly defined by two distinct RDF statements. On the other hand, the TM data model handles bi-directional *n*-ary relationships in a single TM association. Hence, processing and interpreting *n*-ary relationships between entities/topics in the TM case is much more straightforward than in the RDF case. For our application, we have therefore decided to use the TM standard which is described next.

5.3.2 Overview of the Topic Map data model

Topic Maps allow to model knowledge structures contained in information resources with the purpose of enhancing navigation and retrieval on the Web. This standard has been specified under two different syntaxes: the ISO/IEC 13250 Document Type Definition (DTD) [136], which is based on SGML (Standard Generalized Markup Language), and the

XTM 1.0 DTD [139], which has been designed and optimised for the Web and is expressed in XML.

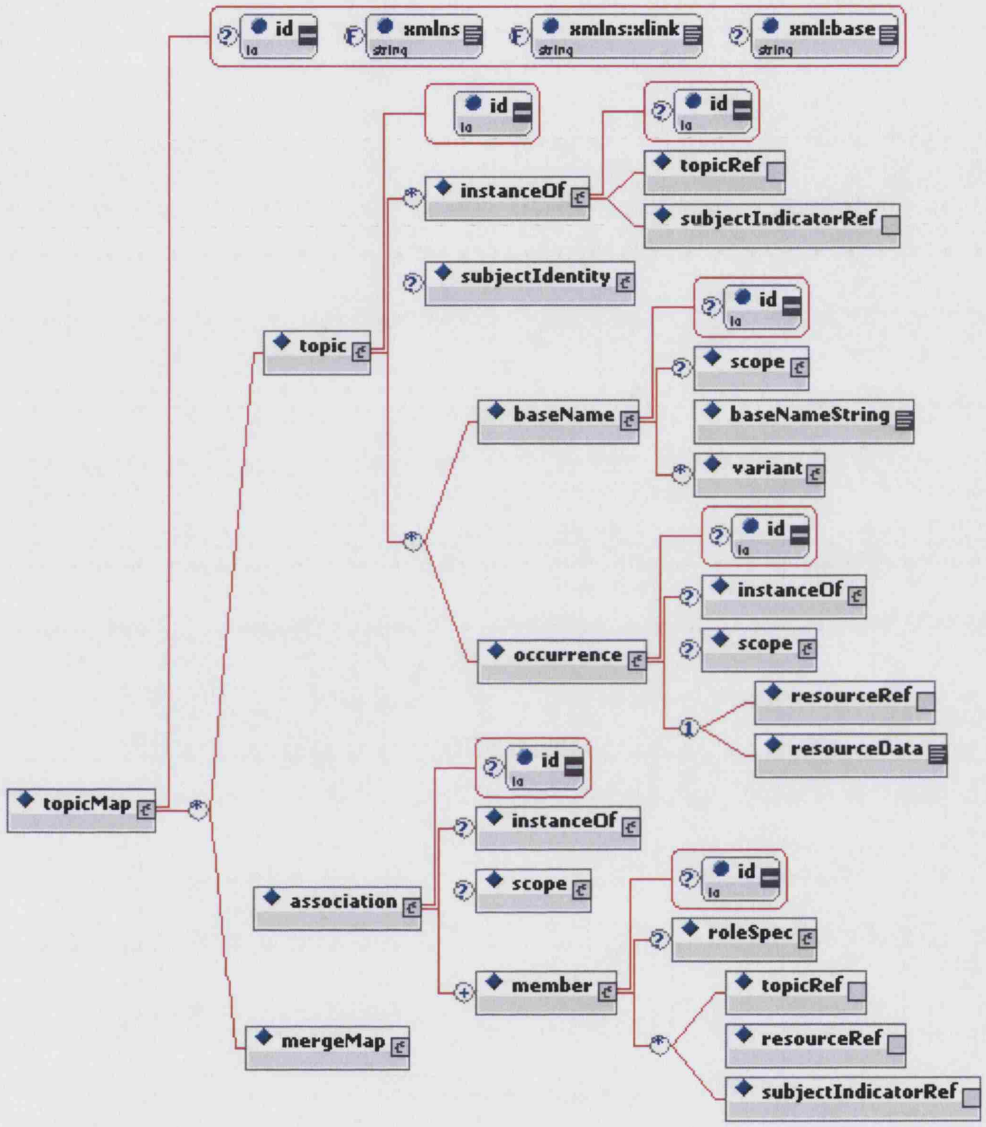


Figure 5.4: Top-level nodes of the Topic Map XTM 1.0 DTD.

Figure 5.4 presents a graphical representation of the top-level nodes of the XTM 1.0 DTD. The basic components of the TM model are: *topics*, *occurrences* and *associations*. Topics are the central elements of the TM model and they can represent any kind of subject (*e.g.* a person, a concept, an idea). Topics can be semantically described by defining explicit names for them (see `baseName` node). The TM standard allows to assign multiple names to the same topic for use in different contexts (*i.e.* through the definition of `variants`). For example, the topic name can be defined in several languages or in different terminology for different communities. Furthermore, topics can be categorized according to their type since

the TM model allows to define that any given topic is an instance of zero or more topic types (see `instanceOf` node). For example, a topic named "UCL E&EE" could be defined as an instance of a topic named "Engineering Department". The relation between topics and their types is a typical class-instance relationship. According to the standard specification, topic types are themselves defined as topics.

Another main component of the TM model is topic occurrences (see `occurrence` node). Topic occurrences represent a set of links to one or more information resources that are considered somehow relevant to the topic. Occurrences are usually external to the TM document and they are addressable by URIs (according to the XTM 1.0 specification). This facility provides a separation between topics and their actual occurrences into two distinct layers, making the TM standard extremely flexible for representing knowledge.

Topic associations, which is the third basic component of the TM model, can represent any kind of relationship between two or more topics. Topics linked by an association are formally identified as members of that association and each of them plays a role in the association (see `member` node). Both associations and member roles can be instances of pre-defined types. According to the standard, association types and role types are also defined as topics.

Since topics may exist in different contexts the TM model presents the facility to scope the characteristics of any given topic (*i.e.* topic name, occurrences and association roles) depending on the particular context. Another feature of this standard is the ability to merge several topic maps without copying or modifying them, which makes the TM technology very scalable.

5.4 Modelling the fuzzy knowledge space with Topic Maps

In this section the representation of the fuzzy knowledge space discovered by the H-FCM algorithm in the document clustering process is addressed. We propose the automatic generation of a clustering topic map based on the XTM 1.0 standard specification. Firstly, it is necessary to identify which entities should be modelled as topics. Hence, we analyse the outcome of the clustering process: the H-FCM algorithm generates c clusters, each of which is represented by a centroid, and a partition matrix containing the

memberships of each document in each cluster. Both centroids and documents are *k*-dimensional vectors where each dimension corresponds to a term.

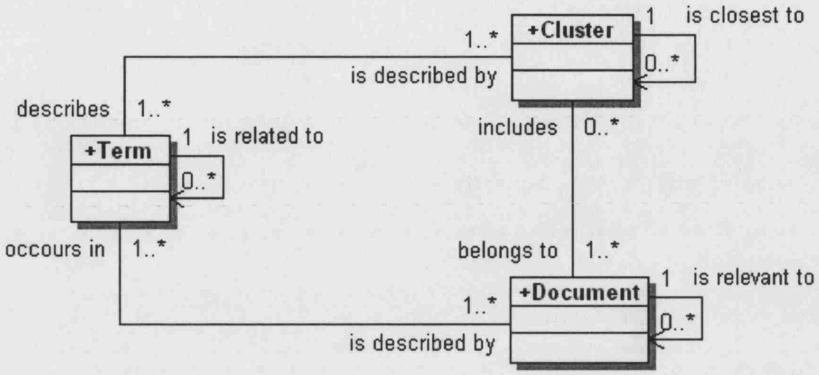


Figure 5.5: UML class diagram for the fuzzy clustering Topic Map template.

From this description of the clustering outcome three topic types can be intuitively identified: cluster, document and term. In Figure 5.5, a UML class diagram representing these knowledge "elements" and the relationships between them is presented.

In the next sub-section we explain how the various relationships between the three topic types have been extracted from the H-FCM results. In sub-section 5.4.2, we define a fuzzy clustering TM template containing the necessary set of association types and role types for representing those relationships. The dynamic generation of a clustering TM is then addressed in sub-section 5.4.3.

5.4.1 Relationships in the fuzzy knowledge space

As mentioned before, the H-FCM algorithm computes the centroids of *c* clusters and the memberships of each document in each cluster in the form of two data matrices. It is thus necessary to define and extract the various relationships shown in Figure 5.5 based on this information.

The definition of cluster-document relationships is straightforward. An association between a given document and a given cluster is defined if the document has been assigned to that cluster, *i.e.* if its membership value is above some threshold α. The membership value is taken as a quantitative measure of the association between the two.

As we have shown in section 4.6, the selected α-cut for the fuzzy clusters impacts on the clustering precision and clustering recall. For a lower threshold value, more documents

are assigned to multiple clusters, which results in lower clustering precision, but in higher clustering recall. Higher recall is important to reveal less obvious associations between documents that contain related topics. This is particularly important in the *e*-Learning context, for enabling flexible exploration of learning material. Hence, the TM includes all cluster-document relationships for the 0.1 α-cut.

The definition of cluster-term and term-document associations follows directly from the term weights in the cluster centroid vectors and document vectors, respectively. Term weights indicate the relative importance of each term in describing the concepts conveyed by clusters or by documents. Terms with the highest weights are the most expressive ones. Hence, cluster-term and term-document associations can be defined based on the terms with the highest weights. Two options have been considered: i) specifying a fixed number of terms t and selecting only the top t terms from each cluster centroid and from each document vector, or ii) specifying the minimum contribution of the term weight to the vectors length and selecting all the terms which satisfy that condition (*e.g.* contribution of at least 1% to the vectors length). The second option is more appropriate because it avoids the selection of insignificant terms and hence, it has been followed for defining cluster-term and term-document associations. Each term weight is taken as a quantitative measure of the association between the term and the cluster/document.

Cluster-cluster relationships are obtained by computing the similarity between each pair of cluster centroids. The most similar cluster to any given cluster is identified and an association between the two clusters is defined. The association is quantified by the degree of similarity between the cluster centroids.

The centroid vectors capture the concepts present in the clustered documents and hence, they provide not only direct but also latent associations between terms. Therefore, a term-term relationship is defined if both terms are present in a given cluster centroid with weights that contribute significantly to the length of that vector (*e.g.* contribution of at least 1% to the vector length, as above).

Finally, document-document relationships are defined as fuzzy relations derived from the cluster membership information contained in the partition matrix. We have decided to obtain the degree of relevance of any given document to some other document using a max-min composite fuzzy relation [140]. Firstly, the minimum of the documents membership values in every cluster is calculated and then, the maximum over all clusters is taken as the degree of association between both documents.

5.4.2 Fuzzy clustering Topic Map template

As mentioned in section 5.3.2, topics, associations and the roles of the members of an association can be instances of pre-defined types. According to the TM standard specification, topic types, association types and role types are explicitly defined as topics. For our application we have defined a set of topic types, association types and role types in a clustering TM template for modelling the fuzzy clustering structure obtained with the H-FCM. Cluster, document and term instances, as well as associations between them, are created based on the pre-defined types within that template. Appendix B contains the complete XML template for the clustering TM.

In Figure 5.6, we present an example topic type declaration. This example contains a topic identified as `tm-cluster` and named `Document group`, that can be instantiated to represent topics of the cluster type. The declarations of document and term types are similar to this one.

```
<topic id="tm-cluster">
   <baseName>
      <baseNameString>Document group</baseNameString>
   </baseName>
</topic>
```

Figure 5.6: XML declaration of the topic type from which clusters are instantiated.

The document type refers to some representation of a given information resource and not the resource itself. In the case of *e*-Learning material, a document would be the metadata record of some learning resource (*e.g.* lecture notes, a diagram, etc.). Thus, an instance of this type is linked, or has an occurrence, in a single information resource. An example of a document from the ODP test collection is given in Figure 5.7.

```
<topic id="http://www.reddykids.com">
   <instanceOf>
      <topicRef xlink:hRef="#tm-document"/>
   </instanceOf>
   <baseName>
      <baseNameString>Reddy's Safety Zone</baseNameString>
   </baseName>
   <occurrence>
      <instanceOf>
         <topicRef xlink:hRef="#resource-description"/>
      </instanceOf>
      <resourceData>
         Educational safety games for kids to help them be
         safe while bike riding, playing sports, and walking
         to school.
      </resourceData>
   </occurrence>
   <occurrence>
      <instanceOf>
         <topicRef xlink:hRef="#resource-hierarchy"/>
      </instanceOf>
      <resourceData>
         Top/Kids_and_Teens/Health/Safety
      </resourceData>
   </occurrence>
   <occurrence>
      <resourceRef xlink:hRef="http://www.reddykids.com"/>
   </occurrence>
</topic>
```

Figure 5.7: XML declaration of a topic of the document type.

In Figure 5.8, we present an example of an association type declaration. This example contains a topic identified as `cluster-document` and named `Cluster-Document relation`, that can be instantiated to represent the membership of a given document (*i.e.* an instance of the topic `tm-document`) in a given cluster (*i.e.* an instance of the topic `tm-cluster`). The declarations of the other types of relationships shown in Figure 5.5 are similar to this one. The role of each member of an association can be expressed either textually or by instantiating pre-defined role types. Like other pre-defined types, role types are also defined as topics and thus, they are also part of the TM template.

In Figure 5.9, we present an instance of the `cluster-document` association type in which the members contain references to three role types: `tm-cluster`, `tm-document` and `tm-membership`.

```
<topic id="cluster-document">
   <baseName>
       <baseNameString>
           Cluster-Document relation
       </baseNameString>
   </baseName>
   <baseName>
       <scope><topicRef xlink:href="#tm-cluster"/></scope>
       <baseNameString>
           Documents in this group
       </baseNameString>
   </baseName>
   <baseName>
       <scope><topicRef xlink:href="#tm-document"/></scope>
       <baseNameString>
           Document group memberships
       </baseNameString>
   </baseName>
</topic>
```

Figure 5.8: XML declaration of the association type from which cluster-document associations are instantiated.

```
<association id="fc_1_d_116">
   <instanceOf>
       <topicRef xlink:href="#cluster-document"/>
   </instanceOf>
   <member>
       <roleSpec>
           <topicRef xlink:href="#tm-cluster"/>
       </roleSpec>
       <topicRef xlink:href="#fuzzyCluster1"/>
   </member>
   <member>
       <roleSpec>
           <topicRef xlink:href="#tm-document"/>
       </roleSpec>
       <topicRef xlink:href="#http://www.reddykids.com"/>
   </member>
   <member>
       <roleSpec>
           <topicRef xlink:href="#tm-membership"/>
       </roleSpec>
       <topicRef xlink:href="#fc_1_d_116_membership"/>
   </member>
</association>
<topic id="fc_1_d_116_membership">
   <baseName>
       <baseNameString> 0.76 </baseNameString>
   </baseName>
</topic>
```

Figure 5.9: XML declaration of an association of the cluster-document type.

This association quantifies the membership of the document from Figure 5.7 in a cluster identified as `fuzzyCluster1`. The third member of the association refers to a topic containing the membership value itself.

5.4.3 Dynamic Topic Map generation

In this section we describe a tool that has been developed and implemented in Java[7] for the dynamic generation of the clustering TM. This tool parses the fuzzy clustering results and converts them according to the TM model into an XML file representing the fuzzy knowledge space.

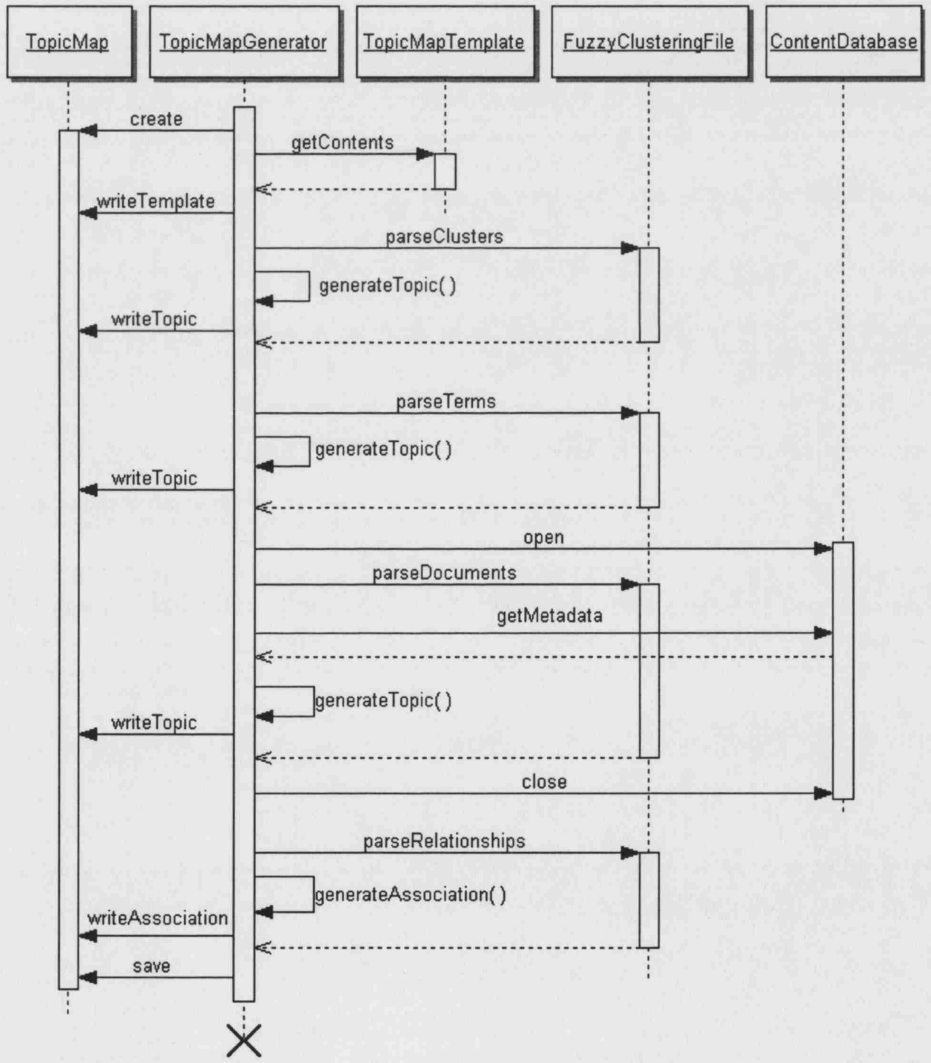


Figure 5.10: UML sequence diagram of the Topic Map generation process.

[7] Java technology homepage: http://java.sun.com/

In Figure 5.10, we present the UML sequence diagram of the TM generation process, which is implemented by the `TopicMapGenerator` class. Firstly, the clustering `TopicMap` file is created and the contents of the `TopicMapTemplate` file are merged into the former. The complete `TopicMapTemplate` file is shown in Appendix B.

Then, the information contained in the `FuzzyClusteringFile` is parsed. This file contains the results of the document clustering process (*i.e.* cluster centroids and fuzzy partition matrix) and it includes information about clusters, terms and documents, as well as the relationships between them (see section 5.4.1). The appropriate topic instances and association instances are generated and written to the `TopicMap` file. When generating topics of the `tm-document` type, the program accesses a database where the metadata about the actual resource is stored to retrieve fields like the document title and description.

5.5 Design and implementation of the prototype Knowledge Navigator

In this section, the Knowledge Navigator tool that has been designed for dynamic Web-based exploration of the fuzzy knowledge space is described. The aim of this tool is to provide means for interpreting the XML clustering topic map and to display all the topics and associations within it in a simple, meaningful and intuitive way. As our ultimate objective is to evaluate whether the fuzzy clustering process succeeds in finding useful content relationships in *e*-Learning material, we have designed this prototype tool focusing mainly on functionality issues rather than on its user interface. Next, we present technical considerations regarding the design of the tool and in sub-section 5.5.2, we provide detailed information regarding its implementation and functionality. The user interface is presented in sub-section 5.5.3.

5.5.1 Design considerations

The use of the TM technology provides a scalable and platform-independent representation of the fuzzy knowledge space, that is suitable for Web-based applications such as *e*-Learning. However, for the topic map to serve its purpose of enhancing the discovery of information resources there needs to be a meaningful way of displaying the knowledge structure to the user.

Traditional visualisation techniques like graphs and trees can be implemented for the navigation of TM [141]. Although such techniques are quite appropriate for capturing the global structure of the topic map, they do not provide many options for presenting detailed information about specific topics or associations. Furthermore, their usefulness becomes somehow limited in the presence of a large number of topics and associations.

Hyper-linked Web pages provide a simple interface for navigating the relationships between topics and for displaying specific information about a topic when the user requests it. Since the topic map is a structured XML document, we have considered several options available for displaying XML content. These are shown in Figure 5.11.

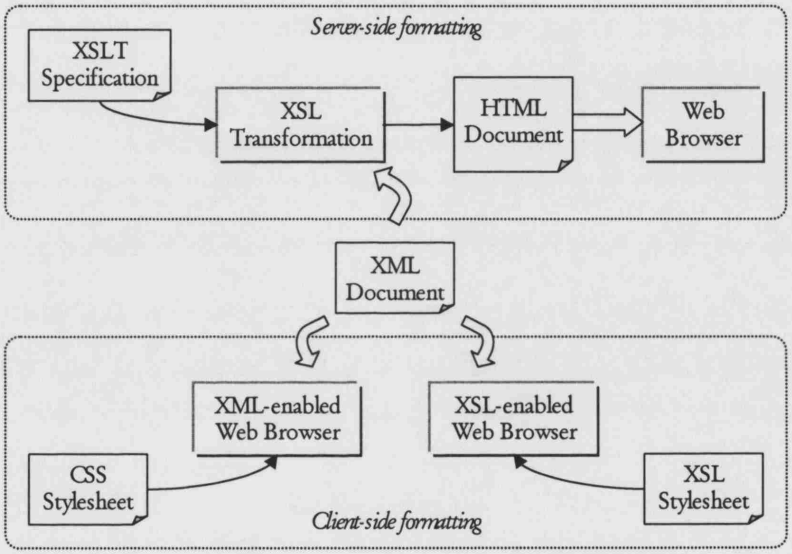


Figure 5.11: Displaying the content of XML documents.

Client-side formatting, *i.e.* on the Web browser, can be performed using stylesheet languages like Cascading Style Sheets (CSS) [142] and eXtensible Stylesheet Language (XSL) [143]. CSS and XSL stylesheets enable to attach style to structured documents, thus allowing the separation between presentation style and content. The XSL standard has been specifically developed for XML. The limitation of the client-side approach is that not every browser offers stylesheets support. Furthermore, customised views of the topic map require more complex transformations of the XML content. XSL Transformations (XSLT) [144] can be used for that purpose. However, like with XSL, not every browser supports XSLT.

Server-side formatting of XML documents basically consists in converting the XML content into HTML format and sending the HTML content to the client. XSLT, which

provide rules for transforming XML documents, are usually processed on the server-side. This approach allows to develop dynamic Web pages that can be customised to the individual user (for *e.g.* in terms of content layout, links displayed, etc.). Thus, this approach is suitable for our application.

We have decided to develop a simple user interface for the exploration of the fuzzy knowledge space using XSLT to transform the information contained in the XML topic map into an HTML frame set. We have selected the Java Servlet technology[8], which emerged mainly for generating dynamic Web content on the server-side, to implement the Knowledge Navigator prototype.

5.5.2 Implementation and functionality

The implementation of the Knowledge Navigator tool involved two main tasks: the specification of a set of XSLT, to define the necessary rules for enabling customised views of the topic map nodes, and the development of a Java package containing all the necessary classes for running the appropriate XSL transformations in response to user requests (*i.e.* HTTP requests).

In section 5.4, we have described the three main topic types for the clustering TM: cluster, document and term. For each of these types we have defined an XSL template to render the associations of a given topic instance and to transform that information into HTML. Three XSL scripts have been developed: `transclust.xsl`, `transres.xsl` and `transterm.xsl` (for clusters, documents and terms, respectively). To make these scripts applicable to any topic instance, we have created variables for storing the topic id. In Figure 5.12, we show two variables defined in the `transclust.xsl` file. When the user requests to view information about a given cluster from a given TM, the names `dummy-topic` and `dummy-tmap` are automatically replaced with the actual topic ID and TM name before applying the XSLT.

```
<xsl:variable name="cluster">dummy-topic</xsl:variable>
<xsl:variable name="tmap">dummy-tmap</xsl:variable>
```

Figure 5.12: Definition of variables in the XSL template.

[8] Java Servlet technology homepage: http://java.sun.com/products/servlet/

```
<xsl:template match="association">
<!-- Test if association is of the #cluster-document type -->
<xsl:if
   test="instanceOf/topicRef[@xlink:href='#cluster-document']">
   <!-- Get the first member of the association, which is the
        topic containing the cluster membership value -->
   <xsl:variable name="membership"
   select="substring-after(member[1]/topicRef/@xlink:href,'#')"/>
   <!-- Get the second member of the association, which is a
        cluster to which the document belongs -->
   <xsl:variable name="cluster"
   select="substring-after(member[2]/topicRef/@xlink:href,'#')"/>
   <!-- HTML formatting -->
   <font face="Arial,Helvetica" color="#0000BB" size="-1"><b>
   <!-- Get the membership value itself -->
   <xsl:apply-templates select="key('topic',$membership)"/></b> -
   <!-- CREATE AN HYPER-LINK -->
   <a>
   <!-- Link to request from the topic map detailed information
        about the cluster in this association -->
   <xsl:attribute name="HREF">
   ControllerServlet?tmap=dummy-map&amp;xsl=transclust&amp;tid=
   <xsl:value-of select="$cluster"/>
   </xsl:attribute>
   <xsl:attribute name="title">
       Click to see this group.
   </xsl:attribute>
   <!-- Display the basename of the cluster which servers as the
        hyper-link text -->
   <xsl:apply-templates select="key('topic',$cluster)">
       <xsl:with-param name="scope">#tm-cluster</xsl:with-param>
   </xsl:apply-templates>
   </a>
   <!-- END OF HYPER-LINK -->
   </font>
   <!-- End of HTML formatting -->
</xsl:if>
<!-- Test if association is of the #term-document type -->
<xsl:if
test="instanceOf/topicRef[@xlink:href='#term-document']">
   ...
</xsl:if>
<!-- Test if association is of the #document-document type -->
<xsl:if
test="instanceOf/topicRef[@xlink:href='#document-document']">
   ...
</xsl:if>
</xsl:template>
```

Figure 5.13: XSL template for rendering associations of topics of the document type.

In Figure 5.13, we show part of the XSL script that has been developed to transform topics of the document type (*i.e.* `transres.xsl`). Specifically, we show how associations of the `cluster-document` type are extracted from the TM and formatted as HTML hyper-links. The hyper-link invokes an HTTP request to a servlet (`ControllerServlet`), which in turn interrogates the TM and retrieves detailed information about the current cluster in the association. The TM name (`tmap`), XSL stylesheet (`xsl`) and cluster topic ID (`tid`) are passed as arguments to the servlet. The dynamic generation of a hyper-link for each association between topics provides the basic mechanism for the navigation of the knowledge space.

Having implemented the XSLT templates, the next phase of the development of the Knowledge Navigator tool consisted in implementing the necessary Java classes for server-side processing of the XSL transformations and also for retrieving content to the user at any time during the browsing process.

Figure 5.14 shows the sequence diagram of the knowledge navigation process. The user accesses the Knowledge Navigator tool through a Web browser which displays an HTML page containing information about a given node of the clustering TM. Then, the user might want to know which documents are related to each other or which concepts are covered by a specific document or he might want to find relevant documents by following term relationships. In any case, the user clicks on the appropriate hyper-link to obtain the associations of the selected topic from the clustering TM. This sends an HTTP request to the `ControllerServlet` containing information about the topic ID, the TM name and also the XSL template to use. Then, the servlet invokes a method of the `XSLModifier` class to open the XSL template and replace the variable names `dummy-topic` and `dummy-tmap` with the values received from the client. A temporary XSL file is created and sent to `ControllerServlet`, which then proceeds with the XSL transformation. The XSLT is implemented by the `XSLTransformer` class, which uses the temporary XSL file to transform the clustering TM and returns the result of the XSLT back to the servlet. Finally, the servlet sends the HTML data to the client.

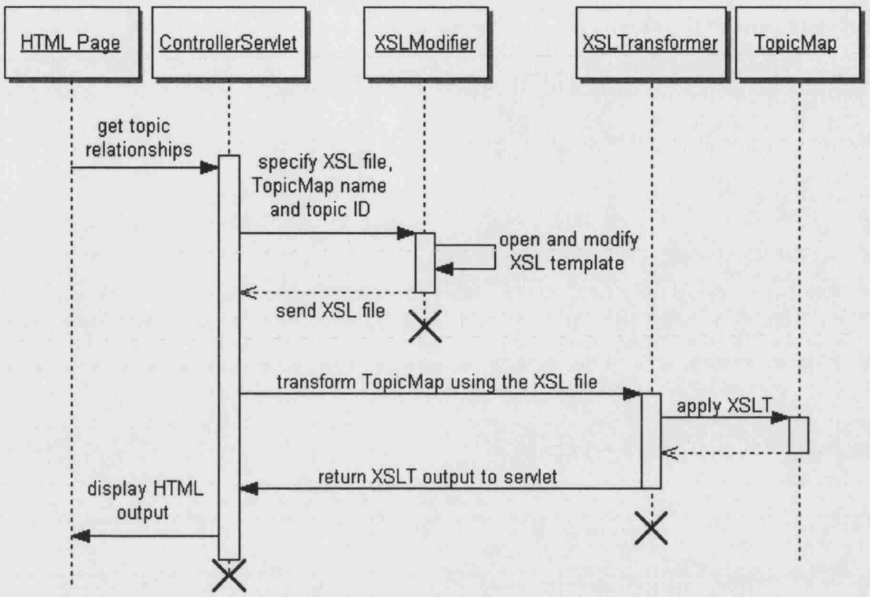


Figure 5.14: UML sequence diagram of the knowledge space navigation process.

Another functionality of the Knowledge Navigator tool is to enable the user to access the actual information resources that are linked to documents defined in the TM. In the case of *e*-Learning, information resources (*i.e.* learning content) are usually stored in a database along with their metadata descriptions. The tool offers the possibility to display the metadata record before downloading the resource itself.

Figure 5.15 shows the sequence diagram of the content retrieval process. Once the user has found a relevant document through the browsing process, a link to its metadata record can be followed for obtaining more detailed information. This sends an HTTP request to the `MetadataServlet` containing the document ID. Then, the servlet accesses the database to retrieve the complete set of metadata fields, which are formatted in XML. Besides the three XSL templates for navigating the TM, another XSL template has been developed for formatting the XML metadata as an HTML page (`metadata.xsl`). The servlet makes a request to `XSLTransformer` to perform the XSLT using this template and then sends the HTML output to the client. A hyper-link is provided in the metadata HTML page for downloading the resource. The link invokes an HTTP request to another servlet, the `ContentServlet`, which accesses the content database and retrieves the resource to the user.

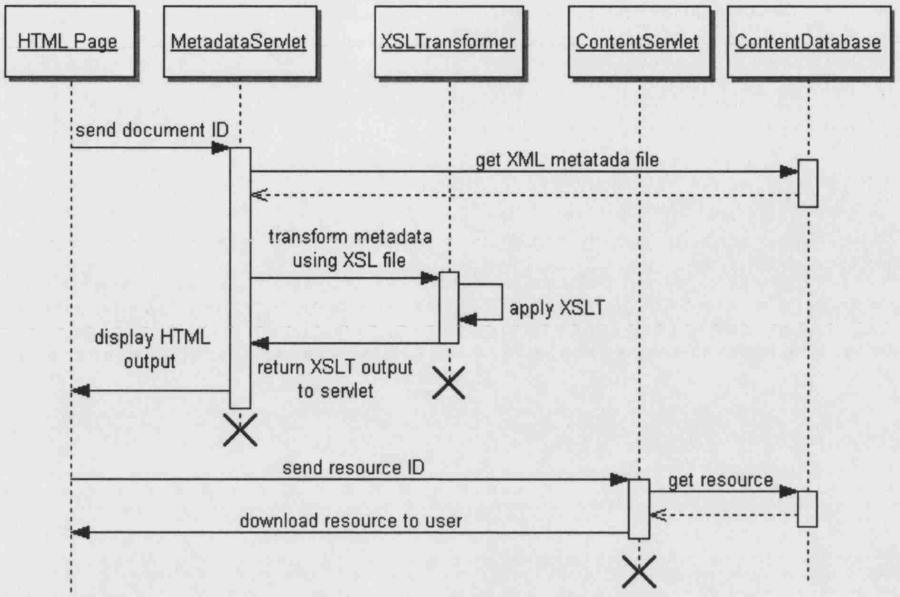


Figure 5.15: UML sequence diagram of the content retrieval process.

Finally, to provide an entry point to the knowledge space we have implemented a simple keyword search facility. The process for such functionality is shown in Figure 5.16. The user enters a keyword of interest that is sent to the `QueryServlet`, which is running on the Web server. The servlet accesses the XML content database and queries it to retrieve the IDs of documents that match the keyword. Then, the servlet formats the results as an HTML page consisting of hyper-links to nodes of the clustering TM. These links can then be used to start navigating the fuzzy knowledge space as shown in Figure 5.14.

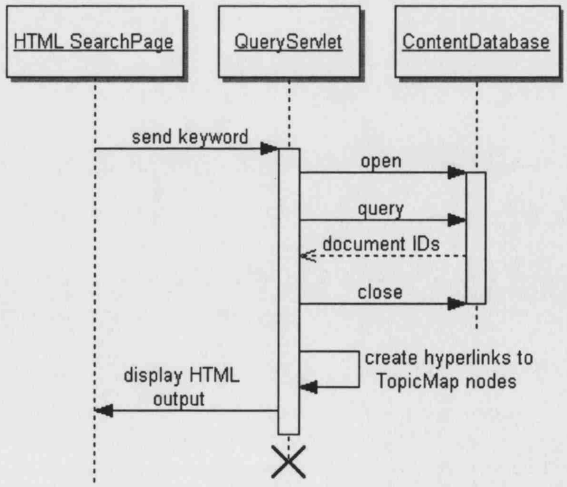


Figure 5.16: UML sequence diagram of the keyword search process.

5.5.3 User interface

In this section we present the Knowledge Navigator's user interface, which has been created for the *e*-Learning trials carried out in the context of the CANDLE project. The user interface basically consists of dynamic HTML pages generated through the processes described previously. As the main goal of this prototype tool was to test the usefulness of the relationships discovered by the fuzzy clustering algorithm, we have designed a simple HTML layout for displaying those relationships. In the following figures, the main screens of the tool are presented.

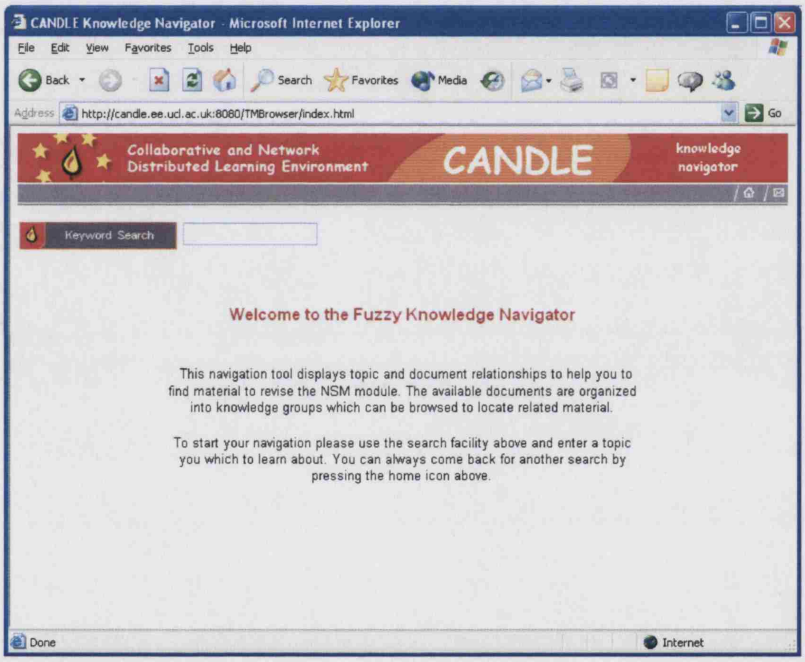


Figure 5.17: Knowledge Navigator welcome screen.

The welcome screen is a static HTML page that provides the user with some basic information on how the tool should be used. The user is encouraged to enter keywords describing the topics on which he or she wishes to obtain more information.

Figure 5.18 presents the results of the search for learning material matching the keyword "security". The HTML page has been dynamically generated by the `QueryServlet` following the process shown in Figure 5.16. The documents listed in this page present the user several alternative points to enter the fuzzy knowledge space. Any document link can be selected for further exploration.

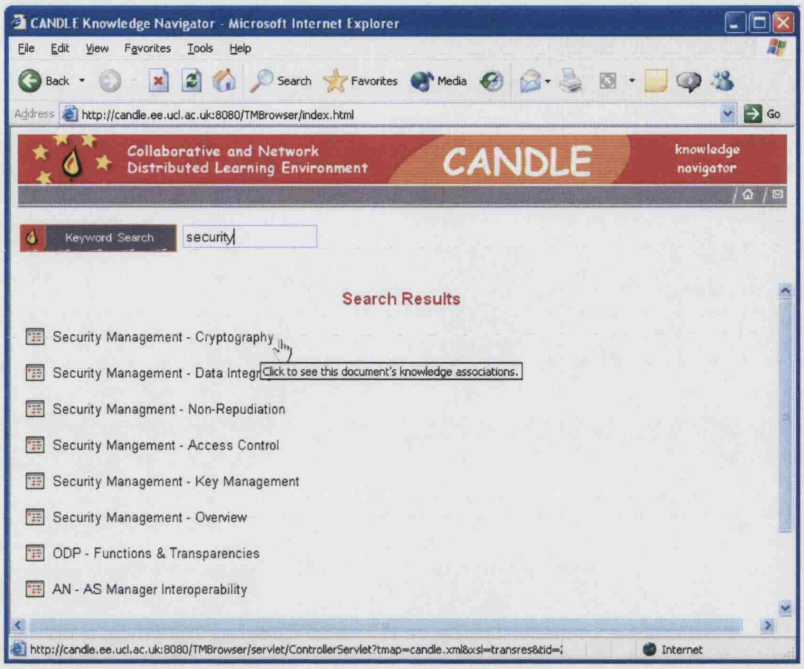


Figure 5.18: Keyword search results.

For instance, the page on the screen below has been displayed after clicking on the "Security Management – Cryptography" link. It has been dynamically generated by the `ControllerServlet` following the process shown in Figure 5.14.

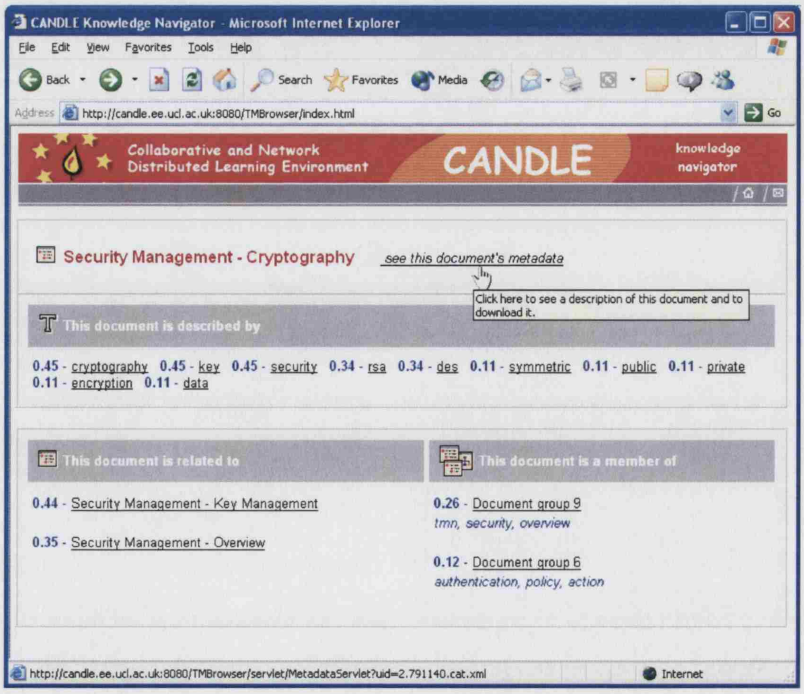


Figure 5.19: View of the document relationships.

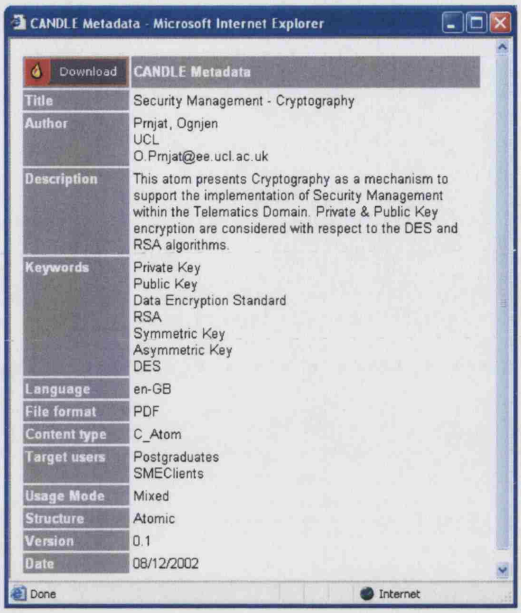


Figure 5.20: Document metadata.

By clicking on the link next to the document title (on the top frame), the window shown in Figure 5.20 opens, containing the metadata of the CANDLE document. In this page, the user can see more detail regarding the document structure and content. If desired, the content can also be downloaded. The metadata page has been dynamically generated by the `MetadataServlet` following the process shown in Figure 5.15.

Below the title frame in the document view page (Figure 5.19), there is a list of terms that describe the document contents. The knowledge relationships of these terms can be seen if the links are followed by the user. On the left hand side of the bottom frame, there are links to documents which are related to the selected document and on the right hand side there are links to the clusters in which the document has membership. These links assist the user to navigate the knowledge space should he or she need to search for other related material.

Figure 5.21 shows a screen containing the relationships of a particular document cluster. Like with the document view page, this page has also been dynamically generated by the `ControllerServlet`. On the top frame, the terms describing the concepts captured by the current cluster are listed in order of their relevance. On the left hand side of the bottom frame, the documents belonging to this cluster are listed and sorted according to their degree of membership. On the right hand side, links to other document clusters are provided with indication of their proximity. This allows the user to move easily to other related concepts of the knowledge space.

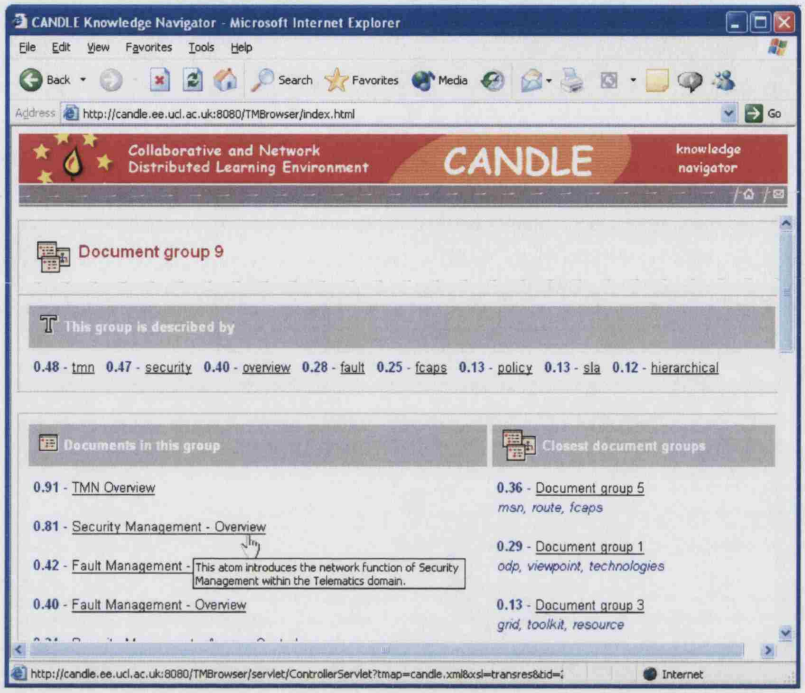


Figure 5.21: View of a fuzzy cluster.

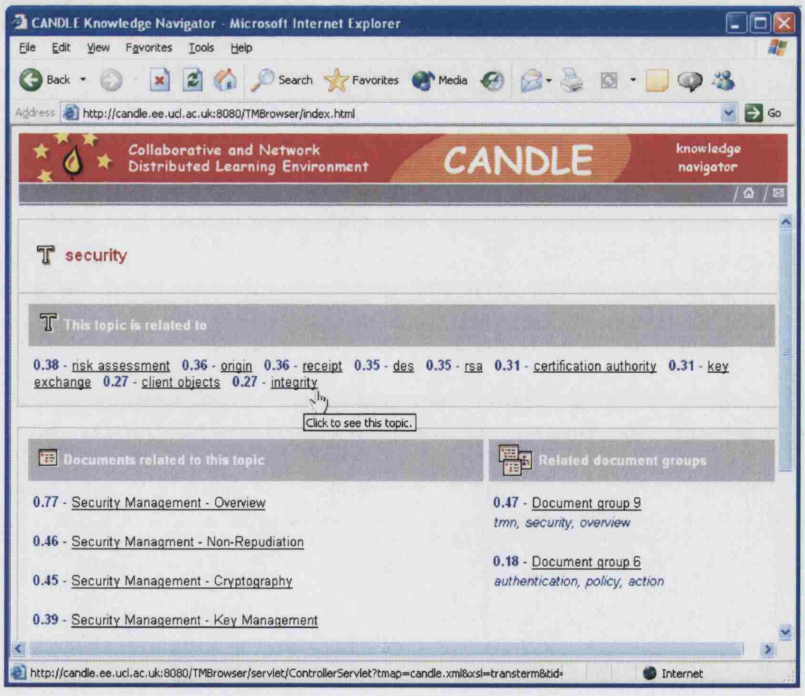


Figure 5.22: View of the relationships of a term.

Finally, a screen containing the relationships of the term "security" is shown in Figure 5.22. Under this view, the chosen term becomes the focal point and therefore the semantics of the frames changes accordingly. The top frame now lists terms that are closely related to the current term. On the left hand side of the bottom frame, there are links to documents that are directly related to this term, while on the right hand side there are links to clusters for which the current term forms a significant descriptive component.

5.6 User trials

The Knowledge Navigator tool has been evaluated in a real *e*-Learning environment in the context of the CANDLE project. In this section, the user trials are described. Firstly, we describe the learning material that was available for the trials and we explain how this material has been clustered. Then, we address the evaluation objectives, the methodology that has been followed and we present the results from the trials.

5.6.1 Clustering the CANDLE learning material

The learning material available for the user trials consisted on lecture notes from a course taught on the University College London MSc in Telecommunications programme entitled "Network and Services Management". This material was engineered into a number of reusable learning objects by the CANDLE project and stored in the content repository. Each object consisted of an XML document containing metadata about a self-contained part of the lecture notes, which included a reference to the actual location of the content. The XML documents were formatted according to the CANDLE metadata schema presented in Figure 5.1.

Clustering the CANDLE material followed the same process described earlier. The XML documents resemble those from the ODP test document collection. Like in the ODP case, the clustering process was based on the metadata information alone. For the current application, documents were indexed based on the following metadata fields: title, description and keywords. These are the fields which bear information about the actual content of the learning object.

The documents were pre-processed to discard very specific terms (using the low specificity filter as described in section 4.4.2) and were numerically encoded using the TF weighting scheme. The H-FCM algorithm was then applied to this set of documents and

the relationships between clusters, documents and terms were derived from the fuzzy clustering results as explained in section 5.4.1. Finally, a topic map for the CANDLE learning material was dynamically generated using the `TopicMapGenerator` tool.

5.6.2 Evaluation setup and results

The main objective of the trials was to evaluate the functionality and usefulness of the Knowledge Navigator tool for helping students in their exploration of online learning material. The CANDLE project also carried out an evaluation of this prototype tool but focusing more on its usability and user interface.

The study involved a group of 25 students attending the MSc in Telecommunications course on "Network Services and Management" at University College London. These students were given the opportunity to revise for their final exam, over a period of one month, using the tool. All the students attended an introductory trial session, where they had the chance to try the functionality of the Knowledge Navigator tool.

The method used to evaluate the tool consisted on custom-designed questionnaires that were handed out at the end of the introductory session, to be filled-in and returned four weeks later. The participation in the trial was not mandatory and only five students returned the questionnaires, giving a response rate of 20%. The raw data and descriptive statistics of the questionnaires are given in Appendix C.

The results obtained from analysing the available responses to the questionnaires are summarised as follows. The students estimated that they had spent on average 6% of their revision time using the Knowledge Navigator tool.

The students also reported that the tool was more useful for finding documents related to a specific topic or for identifying related documents rather than for finding very specific documents (average usefulness scores were 4, 4.25 and 3.4, respectively, on a scale of 1=not useful to 5=very useful). Regarding the relevance of the links, the students found them more or less relevant to what they were looking for (average score was 3.4, 1=not relevant, 5=very relevant).

The keyword search facility was the most used feature for finding documents (average usage score was 4.8, 1=rarely, 5=very often). Browsing the fuzzy knowledge space by term associations and document associations were also used quite often (average scores were 3.5 and 3.6, respectively). The students browsed by document group infrequently compared to the previous browsing facilities (average usage score was 2). Regarding the

usefulness of the document weights some users thought this feature was considerably useful whereas others thought it was not so useful (average score was 2.6, 1=not useful, 5=very useful).

During their navigation of the knowledge space, most students quite often came across other relevant material they had not initially been looking for (average frequency score was 3.4, 1=rarely, 5=very often). They also found that the metadata fields sometimes helped them to distinguish which documents were worth opening or not (average score was 3).

Overall, the students particularly favoured the grouping and display of relevant documents, and the quick and easy search and location of documents. In fact, 3 out of 5 students reported that displaying relevant documents was the best feature of the tool in helping their revision. They also found the correlation of particular terms was a good feature which helped them to link up and classify topics they were learning. The students were also highly in favour of including relevant complementary material in the knowledge space, including that of other universities, which they thought would motivate them to use the tool more often.

5.7 Summary

In this chapter, we have presented the Knowledge Navigator system that has been developed for flexible browsing of *e*-Learning material. The development of such system involved the choice of technologies and the implementation of a tool for representing the knowledge space, which has been discovered by the H-FCM algorithm. It also involved the development of tools for Web-based navigation of the knowledge space relationships.

Initially, we have introduced the CANDLE project, which has provided the context for the research work presented in this thesis. Then, we have reviewed suitable knowledge representation technologies for Web-based applications and we have proposed the use of the Topic Maps standard for modelling in a platform-independent way the set of document and concept relationships that resulted from the fuzzy clustering process. We have developed an XML template for the fuzzy clustering topic map and we have also described a tool for the dynamic generation of an instance of such topic map, which we have implemented in Java.

Subsequently, we have described the implementation of mechanisms and tools for Web-based navigation of the clustering topic map. We have decided to develop a simple user interface for browsing *e*-Learning material in the form of an HTML frame set. We have also decided to use of Java Servlet technology and XSLT to interpret the XML clustering topic map and to dynamically generate the HTML pages. The various navigation facilities have been also detailed. These include browsing *e*-Learning material by related topics, related documents or by document group.

Finally, we have described the deployment of the Knowledge Navigator in a real *e*-Learning system and we have carried out user trials in the context of the CANDLE project. These trials have revealed that grouping and displaying relevant material was the best feature of the tool. Furthermore, students found topic-based navigation quite useful to link up topics of a particular subject. However, the results of these user trials lack statistical significance due to the low response rate of students.

The experiments carried out here involved a relatively small repository of *e*-Learning material. However, in *e*-Learning systems larger repositories are likely to be encountered. In this case, two fundamental issues arise. First, a suitable number of clusters for appropriately representing the knowledge space needs to be defined. Second, the navigation of a large number of clusters with the flat clustering structure produced by the H-FCM is not very intuitive. In the next chapter, we extend the H-FCM algorithm to the Hierarchical Hyper-spherical Fuzzy c-Means (H^2-FCM) algorithm in order to address these problems.

Chapter 6
Hierarchical Hyper-spherical Fuzzy c-Means

6.1 Introduction

As described in the previous chapter, the H-FCM algorithm has been integrated in a Web-based navigation tool that enables flexible exploration of *e*-Learning material. User trials have revealed that grouping and displaying relevant material was the best feature of the tool. Moreover, students found that topic-based navigation was also quite useful for associating topics linked to a particular subject. Such navigation feature has been implemented based on the latent relationships between terms co-occurring in the cluster centroids. However, there are two main issues associated with the clustering process that need to be considered. On the one hand, a suitable number of clusters for appropriately representing the knowledge space needs to be defined. On the other hand, the limitations of exploring a very large number of clusters with a flat clustering structure need to be addressed. In this chapter we address these two issues by proposing a novel hierarchical fuzzy clustering algorithm – the Hierarchical Hyper-spherical Fuzzy c-Means (H^2-FCM).

In section 6.2, we introduce the new clustering algorithm which builds upon the H-FCM clustering outcomes (namely the cluster centroids) and an asymmetric similarity measure to generate a topic hierarchy. In section 6.3, the performance of the H^2-FCM algorithm is evaluated and in section 6.4, the topic hierarchy generated by the algorithm is analysed. Finally, the main contributions of this chapter are summarised in section 6.5.

6.2 Hierarchical H-FCM algorithm

In section 3.6.2, the impact of the initial number of clusters c on the intra- and inter-cluster similarity distributions produced by the H-FCM algorithm has been analysed. We have concluded that finding the optimum number of clusters in the high-dimensional document space is not essential and that the choice of c should rather reflect the desired granularity of the document clusters. However, browsing large document repositories with a flat clustering structure becomes impractical. Topic hierarchies are more user-friendly and allow for a better understanding of the concept relationships in the knowledge space.

Assuming that the topics represented by each document cluster are more specific when c is higher and more generic when c is lower, we propose using the H-FCM algorithm to obtain an over-specified number of clusters and subsequently creating a hierarchical organisation of those clusters based on parent-child type relationships between cluster centroid vectors. We have named this approach as Hierarchical Hyper-spherical Fuzzy c-Means algorithm (H^2-FCM).

This development involves both the definition of a suitable measure to determine parent-child relationships between cluster centroids and the definition of a heuristic for hierarchically linking the clusters based on that measure. The following sub-sections address these two issues.

6.2.1 Asymmetric similarity measure

The H-FCM algorithm groups documents and computes cluster centroids based on symmetric similarity measures such as the cosine, Jaccard or overlap coefficients. As we saw in Chapter 2, document relationships are indeed traditionally assumed to be symmetric. This is mainly due to the fact that most applications, including document clustering, are simply concerned with whether documents are related or not.

However, Tversky [68] has challenged the symmetry assumption arguing similarities based on human judgment are often quite asymmetric. According to Tversky's model of similarity, a document vector containing many indexing terms should be judged less similar to a sparser document vector than the opposite. Recent examples that apply asymmetric similarity measures to determine inclusion relations between text documents can be found in [70, 145]. For our purpose of building a cluster hierarchy with the H-FCM cluster

centroid vectors, a parent-child relationship between those vectors has necessarily to be asymmetric.

By definition a parent-child relationship embraces the concept of inheritance: the child inherits all the attributes from its parents, while adding some new attributes. Hence, a child vector should contain all the terms from its parent vector plus additional terms with non-zero weight. In our application, cluster centroids are high-dimensional vectors of unit length containing k term weights. Each centroid vector is a function of all the documents that belong to its cluster and hence, it is likely that many dimensions of the centroid vectors contain low term weights. Consequently, parent-child relationships that conform to the definition above are not expected to occur between cluster centroids. However, following Tversky's model of similarity the notion of inheritance can be relaxed considering that a child cluster should be less similar to its parent than the opposite. Thus, we propose using an asymmetric similarity measure to obtain parent-child relationships between centroids. An example of such measures that has been applied in the IR field [73] is given in equation (6.65). This measure will be used in our experiments. However, other asymmetric similarity measures could be defined and applied to link cluster centroids hierarchically.

$$S_{asymmetric}(\boldsymbol{v}_\alpha, \boldsymbol{v}_\beta) = \frac{\sum_{j=1}^{k} \min(v_{\alpha j}, v_{\beta j})}{\sum_{j=1}^{k} v_{\alpha j}} \tag{6.65}$$

Analysing the equation above, if $\boldsymbol{v}_\alpha$ is a child of $\boldsymbol{v}_\beta$ then $S(\boldsymbol{v}_\alpha, \boldsymbol{v}_\beta) < S(\boldsymbol{v}_\beta, \boldsymbol{v}_\alpha)$. Otherwise, if $\boldsymbol{v}_\beta$ is a child of $\boldsymbol{v}_\alpha$ then $S(\boldsymbol{v}_\alpha, \boldsymbol{v}_\beta) > S(\boldsymbol{v}_\beta, \boldsymbol{v}_\alpha)$.

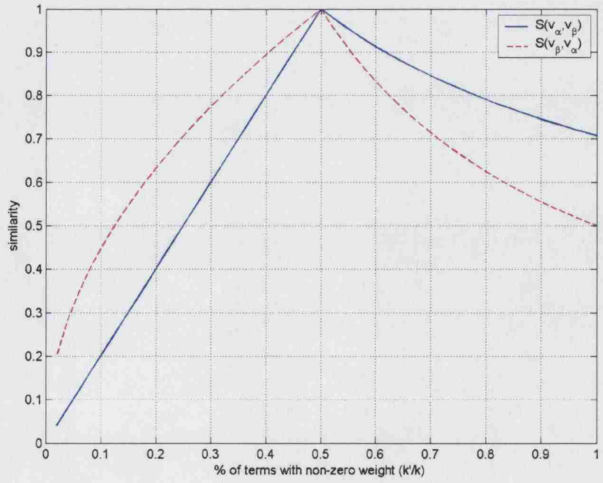


Figure 6.1: Behaviour of the asymmetric similarity measure for sparse unit-length vectors $\boldsymbol{v}_\beta$ containing k' uniform term weights, considering $\boldsymbol{v}_\alpha$ with a sparsity level of 50%.

In Figure 6.1, we give an example to support this, for the case where the inheritance condition is satisfied. We have considered k-dimensional unit length vectors with varying sparsity levels. The k' non-zero entries of each vector v_β consist of uniform weights, *i.e.*, $v_{\beta j} = 1/\sqrt{k'}$, $\forall_j$. A reference vector v_α with a sparsity level of 50% has been taken and the similarity to every other vector has been calculated using the function in equation (6.65).

From the plots it can be observed that if v_β has fewer terms (*i.e.* more zero entries) than v_α, then $S(v_\alpha, v_\beta) < S(v_\beta, v_\alpha)$, and vice-versa. Thus, in the first case v_β is a parent of v_α and in the second case v_β is a child of v_α.

6.2.2 Description of the H²-FCM algorithm

The new H²-FCM algorithm consists of three main stages. It starts by invoking the H-FCM algorithm for obtaining the desired number of document clusters. Then, it takes each pair of clusters and calculates their asymmetric similarity based on the cluster centroids and finally, it links the clusters hierarchically using a top-to-bottom approach to obtain a topic hierarchy. The pseudo-code description that summarises the H²-FCM algorithm is given in Figure 6.2.

In the first stage (steps 1 to 4), the selected number of H-FCM clusters should be sufficiently high to reflect the desired granularity for the topic hierarchy. To avoid the inclusion in the hierarchy of clusters with very few documents, a threshold t_{ND} for the minimum number of documents is defined. This user controlled parameter serves to eliminate K clusters that for a given α-cut contain less than t_{ND} documents. The H-FCM algorithm is re-applied for $c=c-K$, until there are no more clusters to be eliminated. A high α-cut can be used to ensure that at least t_{ND} documents have high cluster membership.

The second stage of the algorithm (step 5) involves $c\times(c-1)$ similarity calculations using the measure in equation (6.65), one for each pair of distinct cluster centroid vectors. A $(c\times c)$ matrix S is generated where the element $s_{\alpha\beta}$ quantifies the parent-child relationship between centroid v_α and centroid v_β. The diagonal entries of S are set to zero.

In the third stage of the algorithm (steps 8 to 12) we take an heuristic approach for generating the hierarchy itself. The development of such heuristic has involved two main considerations regarding which criteria to use for: i) selecting a candidate cluster to be added to the hierarchy and for ii) finding a parent cluster in the hierarchy for the current candidate.

1. **Apply** the H-FCM algorithm to a $N \times k$ data matrix X for an **over-specified** number of clusters c and **specify** the minimum number of documents t_{ND} per cluster for the desired α-cut and obtain the partition matrix U and cluster centroids V
2. **While** (there are $K>0$ clusters with less than t_{ND} documents)
3. **Re-apply** the H-FCM algorithm for $c = c-K$ clusters
4. **Endwhile**
5. **Compute** the asymmetric similarity between each pair of cluster centroids using (6.65)
6. **Specify** the parent-child similarity threshold t_{PCS}
7. **Define** V_H and V_F as the set of cluster centroids assigned and not assigned to the hierarchy, respectively. Initially, $V_H = \emptyset$ and $|V_F| = c$
8. **While** ($V_F \neq \emptyset$)
9. **Select** a candidate vector $v_\alpha \in V_F$ such that
   $\exists v_\beta \in V_F : S(v_\alpha, v_\beta) = \max[S(v_i, v_\varphi)], \forall v_i, v_\varphi \in V_F$
   If there is more than one candidate, temporarily set
   $S(v_\alpha, v_\beta) = 0$ and repeat selection process as before
10. **If** $V_H = \emptyset$ **make** v_α a root cluster
11. **Else find** the set of vectors $V_P \subseteq V_H$ such that
    $S(v_\alpha, v_\gamma) \geq t_{PCS}, \forall v_\gamma \in V_P$, and **make** v_α a child of v_γ
    If $V_P = \emptyset$ make v_α a root cluster.
11. **Remove** v_α from V_F and **add** it to V_H
12. **Endwhile**
13. **Foreach** root cluster v_ρ get the number of documents N_ρ in its hierarchy branch
14. Starting with the lowest N_ρ,
    If ($N_\rho < t_{NDH}$) remove v_ρ from the root of the hierarchy and **find** a parent cluster $v_\gamma \in V_H$ such that
    $S(v_\rho, v_\gamma) = \max[S(v_\rho, v_\varphi)], \forall v_\varphi \in V_H$, and make v_ρ a child of v_γ
15. **Return** cluster hierarchy, partition matrix U and cluster centroids V

Figure 6.2: Summary description of the H²-FCM algorithm.

We have defined a candidate selection criterion based on the maximum asymmetric similarity among the set of clusters that have not yet been assigned to the hierarchy (step 9). The candidate cluster is thus considered to be the best parent to one of the remaining non-assigned clusters. Such selection process ensures the right ordering for a descending cluster insertion (*i.e.* top to bottom) into the hierarchy.

Once the candidate cluster has been selected, it is assigned to the hierarchy either under one or more of the existing clusters or at the root of the hierarchy. For the insertion of the candidate cluster into the hierarchy, we have defined a parent selection criterion based on a user defined threshold t_{PCS} for the asymmetric similarity. If the similarity

between the candidate and a potential parent in the hierarchy equals or falls above the threshold t_{PCS}, then the candidate becomes a child of that cluster (step 11). In case no suitable parent is found (*i.e.* asymmetric similarity below t_{PCS}), the candidate cluster starts a new hierarchy tree. This linking process ensures that a cluster representing distinct topics from those already represented in the hierarchy is able to start its own topic tree. The higher the threshold value, the higher the number of clusters that will appear at the root and the lower the number of hierarchy levels. Thus, the t_{PCS} parameter enables to control either the number of hierarchy roots or the depth of the hierarchy.

Finally, the algorithm allows to remove small clusters from the root of the hierarchy (step 14). A parameter t_{NDH} is defined such that if a given hierarchy branch (*i.e.* the tree under a given hierarchy root) has less than t_{NDH} documents assigned to it, then that branch is eliminated and its documents reassigned to other branches. This enables to ignore clusters which are not representative of the main topics of a given document collection.

6.2.3 Related work

The development of the H²-FCM algorithm was the result of our research into scalability issues. As mentioned in Chapter 2, traditional hierarchical clustering methods are computationally unattractive as they present time and memory complexity of at least $O(N^2)$, where N is the number of documents. In section 3.4, we have reviewed some hierarchical fuzzy clustering methods which also present high computational cost [111, 112, 113]. Here, we present a new clustering method which generates fuzzy clusters using a linear time algorithm $O(N)$, the H-FCM, while at the same time builds a topic hierarchy with low computational cost.

The idea of over-specifying the number of clusters has been previously explored in [146, 147, 148] with the goal of finding the optimum number of clusters. The methods presented in [146, 147] are fuzzy relational methods that start with a high number of clusters which are successively merged based on the clusters cardinality (*i.e.* the number of elements assigned to the clusters). The method proposed in [148] combines the FCM clustering algorithm with a hierarchical ascending method to build a proximity graph with the initial set of clusters. This graph is then cut based on a compactness criterion and clusters that remain linked are merged. The reason why we over-specify the number of clusters in our algorithm is not to attempt to find the optimum number of clusters but

rather to represent the desired granularity of the topics covered by a given set of documents.

The dynamic generation of topic hierarchies by means of clustering methods that explore asymmetric relations has been previously addressed [33, 69]. In [33], a hierarchical algorithm that uses the concept of inheritance has been proposed for clustering Web-search results. The algorithm works with binary document vectors, and selects some of those vectors to represent hierarchy classes based on the number of terms they contain. Documents are then allocated to those classes using a minimum distance criterion. In [69], a related clustering algorithm has been proposed for summarising document collections. Instead of working with N document vectors of k dimensions this algorithm works with k binary concept vectors of N dimensions, *i.e.* an inverted data matrix is used. The algorithm selects some of those concepts to represent, or lead, hierarchy clusters based on the number of documents they appear in. Concepts are then assigned to an existing cluster if their relation to the cluster leader is higher than a pre-defined threshold. Sub-clusters are created by eliminating the lead concept and recursively applying the same algorithm.

The heuristic used in our algorithm to build the cluster hierarchy is also related to the ones above. However, instead of using raw binary document or concept vectors, we use cluster centroid vectors containing weighted terms that represent the topics covered by the clustered documents. Furthermore, the H^2-FCM algorithm generates a fuzzy partition and although the previous methods also allow documents to be assigned to multiple clusters, they do not quantify the degree of membership. With the H^2-FCM algorithm, documents can be sorted according to their relevance to the topics represented by a given cluster.

6.3 Evaluation of the H^2-FCM algorithm

In this section, the performance of the H^2-FCM algorithm is investigated in terms of the document clustering quality and of the algorithm computational cost. For the current experiments we use the same test document collections as before, which have been pre-processed with the τ_{low}=0.40 specificity filter and encoded with the TF scheme. As we observed in Chapter 4, such filter leads to significant dimensionality reduction of the document vectors, without having any negative impact on the performance of the H-FCM algorithm.

The main goal of these experiments is to establish whether having more clusters does indeed mean more granularity regarding the topics represented by each of them and whether the hierarchical linking heuristic produces meaningful associations between the clusters. In section 6.3.1, the impact of the initial number of clusters on the external quality of the hierarchy clusters is investigated. In section 6.3.2, the effects of the t_{PCS} threshold on the cluster hierarchy characteristics is examined and in section 6.3.3 the time complexity of the cluster linking heuristic is analysed.

6.3.1 Impact of the initial number of clusters

In Chapter 4, the quality of the H-FCM clustering results has been evaluated for a number of clusters c that corresponded to the number of reference classes in each document collection c_{REF} (as described in Table 3.2). Clustering precision and recall have been calculated by comparing the documents from each H-FCM cluster with those from the reference classes. In the present experiments, the H^2-FCM algorithm generates a number of smaller sized clusters, *i.e.* $c>c_{REF}$, and links them hierarchically. Thus, for assessing the performance of the algorithm not only do we have to analyse the quality of each individual cluster but also to determine whether sub-clusters of the same reference class are linked in the hierarchy.

As explained in section 6.2.2, the user defined parameter t_{PCS} controls the depth and the number of roots of the cluster hierarchy. When a limit is set for the hierarchy depth or for the number of hierarchy roots, the value of this threshold can be adaptively found. For example, the algorithm can start with a low value for t_{PCS} and then successively increase this threshold until the desired hierarchy depth or the desired number of root clusters is obtained. For performance evaluation purposes, we follow this approach to obtain as many roots as the number of reference classes in each document collection. Clustering precision and clustering recall are then calculated by comparing the contents of all the clusters from a given hierarchy branch with the contents of the corresponding reference class. The documents membership in a given branch is taken as their maximum membership in any of the branch clusters. A good H^2-FCM performance thus indicates that both the quality of each individual cluster is good and the sub-clusters of the same reference class are part of the same hierarchy branch.

Figures 6.3 through 6.6 present the average clustering precision and recall of the H²-FCM algorithm as a function of the number of clusters, obtained for the REUTERS1, REUTERS2, ODP and INSPEC collections, respectively. The average clustering precision and recall have been calculated as explained in section 4.3.2.

For these experiments, the fuzzification parameter m has been set to 1.10 and the cosine coefficient has been used in the H-FCM algorithm. From the experiments described in Chapter 4, we have concluded that the H-FCM performance was generally better with this m value and when the cosine similarity measure was used. For the first stage of the H²-FCM algorithm (see steps 1 to 4 in Figure 3.9), t_{ND} has been set to 2 and an α-cut of 0.5 has been used. On the one hand, this allows small clusters to be formed, but on the other hand, such clusters have to be meaningful (since a degree of membership of 0.5 indicates that the similarity between the documents and the cluster centroid is relatively high).

From the plots it can be observed that the performance of the algorithm generally does not degrade as the number of clusters in the hierarchy increases. The average clustering precision and recall do not vary significantly for any of the test document collections. The quality of each individual cluster can be analysed indirectly based on these results. We can conclude that as c increases, documents from the same reference class remain grouped together, but these documents are now divided into a higher number of smaller clusters. Furthermore, we can also conclude that the hierarchical linking procedure succeeds at placing in the same hierarchy branch clusters corresponding to the same reference topic.

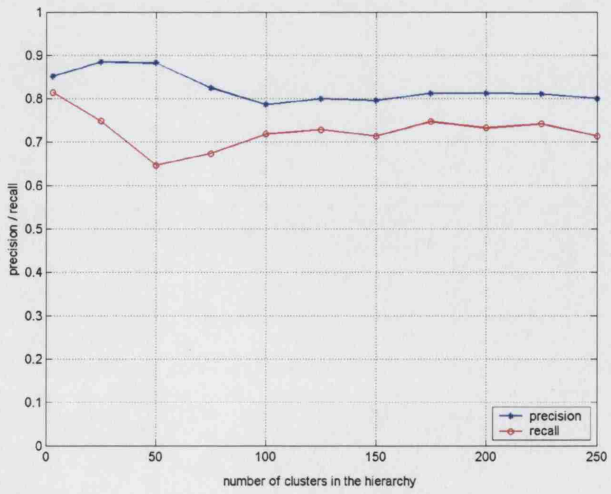


Figure 6.3: Average precision and recall of the H²-FCM (m=1.10) for the REUTERS1 collection.

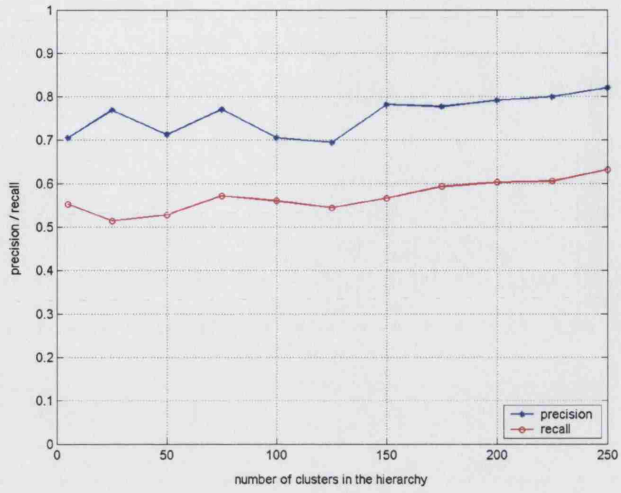


Figure 6.4: Average precision and recall of the H²-FCM (m=1.10) for the REUTERS2 collection.

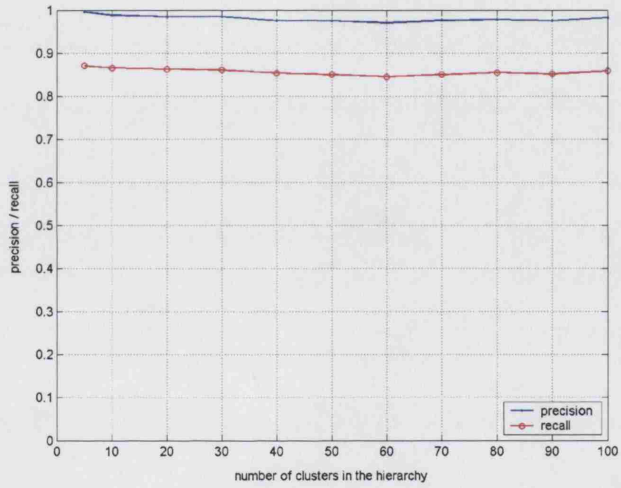


Figure 6.5: Average precision and recall of the H²-FCM (m=1.10) for the ODP collection.

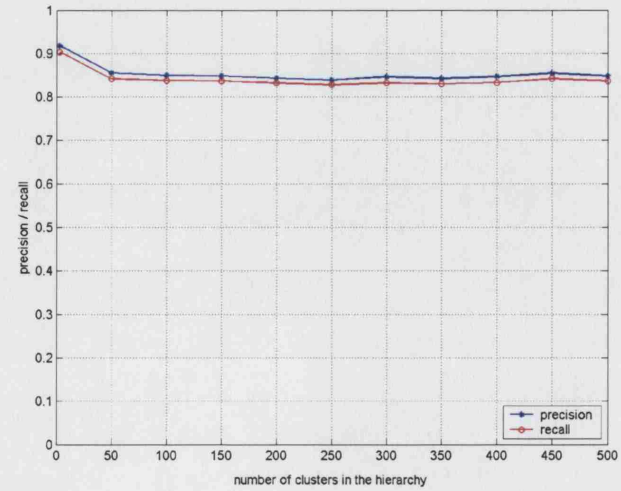


Figure 6.6: Average precision and recall of the H²-FCM (m=1.10) for the INSPEC collection.

6.3.2 Impact of the asymmetric similarity threshold t_{PCS}

The cluster hierarchy obtained with the H²-FCM algorithm depends on the value of the user-defined asymmetric similarity threshold t_{PCS}. In the previous experiments the value of this parameter has been adaptively found in order to obtain as many hierarchy roots as the number of reference classes in each document collection. The algorithm started with a low value for t_{PCS}, which was successively increased, by a fixed increment Δt, until the number of root clusters was equal to the number of reference classes. Firstly, a coarse resolution was used to search for the right value (Δt=0.1). Then, if the number of root clusters exceeded the number of reference classes, t_{PCS} was decreased to the previous value ($t_{PCS}=t_{PCS} - \Delta t$) and a higher resolution was used to search in the right sub-interval ($\Delta t=\Delta t/2$).

In Table 6.1 we present an example of the impact of this parameter on the number of hierarchy roots and on the depth of the hierarchy. This data has been obtained with the ODP collection for c=40. As expected, an increase of t_{PCS} leads to an increase of the number of roots and to a decrease of the number of hierarchy levels. The same behaviour is observed for other values of c and for all document collections

Table 6.1: Characteristics of the cluster hierarchy for various levels of the asymmetric similarity threshold t_{PCS} (ODP collection and c=40)

t_{PCS}	0.21	0.24	0.28	0.31	0.42	0.44	0.49	0.59	0.60	0.61	0.65	0.67	0.68
roots	1	2	4	5	6	7	8	9	10	11	12	14	15
depth	6	5	5	4	4	4	4	3	3	3	3	3	3

t_{PCS}	0.72	0.74	0.75	0.76	0.78	0.79	0.80	0.81	0.82	0.83	0.84	0.85	0.86
roots	17	20	21	22	24	25	27	29	31	32	33	34	37
depth	3	3	3	3	3	3	3	3	2	2	2	2	2

In some applications it may be desirable to control the number of root clusters or the number of levels in the hierarchy. In particular, for efficient browsing of large document collections (*e.g.* in *e*-Learning applications) the number of root clusters presented to the user should not be very large. Otherwise, the navigation of the hierarchical structure can become impractical.

6.3.3 Time complexity of the cluster linking heuristic

In this section, the order of the H²-FCM time complexity is analysed. The complexity of this algorithm depends both on the H-FCM complexity and on the complexity of the cluster linking heuristic.

As mentioned before, the H-FCM computation time varies linearly with the number of documents N and with the number of dimensions k. For fixed N and k, the computation of the H-FCM cluster centroids runs in at most $O(c^2)$, essentially due to the computational cost associated with the computation of the partition matrix (see step 7 in Figure 3.9). Thus, the overall time complexity of the H-FCM algorithm is $O(Nc^2k)$. However, such cost can be reduced to $O(Nck)$ following the suggestion in [149], whereby the update of the partition matrix and cluster centroids is combined into a single update of the cluster centroids at each iteration of the algorithm.

For a fixed number of clusters c, the time complexity of the H²-FCM algorithm is dictated by the time complexity of the H-FCM algorithm, because the time required for generating the cluster hierarchy is only a function of the number of clusters. Thus, we now analyse the order of the H²-FCM time complexity as the number of clusters varies.

The computation of the asymmetric similarity between every pair of cluster centroids in step 5 of the H²-FCM algorithm is $O(c^2)$. The generation of the cluster hierarchy (steps 8 to 12) involves c cluster insertions into the hierarchy and each of those insertions requires the selection of a parent cluster based on a comparison of the asymmetric similarity against the t_{PCS} threshold. Thus, the linking heuristic runs in at most $O(c^2)$.

To support this analysis, Table 6.2 contains the simulation results of the H²-FCM computation time for increasing values of c, obtained with the test document collections. The computation time t associated with algorithmic steps 8 to 12 has been divided by c, c^2 and c^3, to determine if the complexity is of linear, quadratic or cubic order. The data presented in the table indicates quadratic complexity, since t/c^2 is approximately constant for increasing values of c.

To conclude, the overall time complexity of the H²-FCM is of the same order as the H-FCM algorithm, *i.e.* $O(Nc^2k)$. Thus, the new algorithm scales well with the number of documents.

Table 6.2: Analysis of the order of the time complexity of the H²-FCM cluster linking heuristic.

	REUTERS1									
c	25	50	75	100	125	150	175	200	225	250
t/c	3.18×10^{-2}	7.51×10^{-2}	1.37×10^{-1}	2.34×10^{-1}	3.49×10^{-1}	5.50×10^{-1}	7.46×10^{-1}	1.02×10^{0}	1.36×10^{0}	1.83×10^{0}
t/c^2	5.08×10^{-5}	3.00×10^{-5}	2.44×10^{-5}	2.34×10^{-5}	2.23×10^{-5}	2.45×10^{-5}	2.43×10^{-5}	2.54×10^{-5}	2.69×10^{-5}	2.93×10^{-5}
t/c^3	2.03×10^{-6}	6.01×10^{-7}	3.25×10^{-7}	2.34×10^{-7}	1.79×10^{-7}	1.63×10^{-7}	1.39×10^{-7}	1.27×10^{-7}	1.19×10^{-7}	1.17×10^{-7}
	REUTERS2									
c	25	50	75	100	125	150	175	200	225	250
t/c	3.18×10^{-2}	7.56×10^{-2}	1.38×10^{-1}	2.25×10^{-1}	3.65×10^{-1}	5.18×10^{-1}	7.41×10^{-1}	1.02×10^{0}	1.38×10^{0}	1.80×10^{0}
t/c^2	5.09×10^{-5}	3.02×10^{-5}	2.45×10^{-5}	2.25×10^{-5}	2.33×10^{-5}	2.30×10^{-5}	2.42×10^{-5}	2.56×10^{-5}	2.73×10^{-5}	2.87×10^{-5}
t/c^3	2.04×10^{-6}	6.05×10^{-7}	3.26×10^{-7}	2.25×10^{-7}	1.87×10^{-7}	1.53×10^{-7}	1.38×10^{-7}	1.28×10^{-7}	1.21×10^{-7}	1.15×10^{-7}
	ODP									
c	10	20	30	40	50	60	70	80	90	100
t/c	1.16×10^{-2}	2.55×10^{-2}	4.04×10^{-2}	5.68×10^{-2}	7.69×10^{-2}	1.00×10^{-1}	1.25×10^{-1}	1.57×10^{-1}	1.91×10^{-1}	2.30×10^{-1}
t/c^2	1.16×10^{-4}	6.38×10^{-5}	4.48×10^{-5}	3.55×10^{-5}	3.08×10^{-5}	2.78×10^{-5}	2.56×10^{-5}	2.45×10^{-5}	2.36×10^{-5}	2.30×10^{-5}
t/c^3	1.16×10^{-5}	3.19×10^{-6}	1.49×10^{-6}	8.87×10^{-7}	6.15×10^{-7}	4.64×10^{-7}	3.65×10^{-7}	3.06×10^{-7}	2.62×10^{-7}	2.30×10^{-7}
	INSPEC									
c	50	100	150	200	250	300	350	400	450	500
t/c	7.83×10^{-2}	2.32×10^{-1}	5.19×10^{-1}	1.01×10^{0}	1.75×10^{0}	2.91×10^{0}	4.20×10^{0}	6.42×10^{0}	8.88×10^{0}	$1.20\times10^{+1}$
t/c^2	3.13×10^{-5}	2.32×10^{-5}	2.31×10^{-5}	2.53×10^{-5}	2.79×10^{-5}	3.23×10^{-5}	3.43×10^{-5}	4.01×10^{-5}	4.39×10^{-5}	4.78×10^{-5}
t/c^3	6.26×10^{-7}	2.32×10^{-7}	1.54×10^{-7}	1.27×10^{-7}	1.12×10^{-7}	1.08×10^{-7}	9.80×10^{-8}	1.00×10^{-7}	9.75×10^{-7}	9.56×10^{-7}

6.4 Topic hierarchy

The set of terms that compose the centroid of a given cluster represent the topics that tie documents together in that cluster. The hierarchy generated by the H²-FCM algorithm acknowledges that some topics may be more specific than others, hence cluster centroids are organised hierarchically. This form of knowledge representation provides an additional layer for efficient browsing of large document repositories.

In the previous chapter we have described how relationships between documents and between terms could be extracted from the H-FCM clustering outcomes, *i.e.* cluster centroids and fuzzy partition matrix (see section 5.3.1). We have also shown how these relationships could be used for locating related documents during an *e*-Learning browsing session. The same browsing features still apply to the H²-FCM outcomes. The hierarchical

layer provided by this algorithm enables to extract additional relationships between terms and between clusters (*i.e.* inclusion relations) as well as to present the whole clustering structure to the user for browsing the cluster contents more effectively (*e.g.* through an interactive graph or a folder tree) and for contextualising the user's navigation path in the whole knowledge space.

An example of the H²-FCM clustering outcomes is presented in Figure 6.7. The picture shows a graph representation of the cluster hierarchy obtained for the ODP test document collection (c=40) with t_{PCS}=0.31 (which led to 5 root clusters and 4 hierarchy levels). To simplify the graph visualisation, we have hardened the fuzzy clusters using the maximum membership criterion, *i.e.* we only show documents in the cluster where their membership is maximum (the light blue nodes in the graph are documents).

The topics covered by each hierarchy branch are summarised by the centroids of the root clusters. The top term weights that contribute to 80% of the length of each centroid vector are shown in the graph. It can be observed that these terms correspond to the topics of the reference classes of the ODP collection (see Table 3.2).

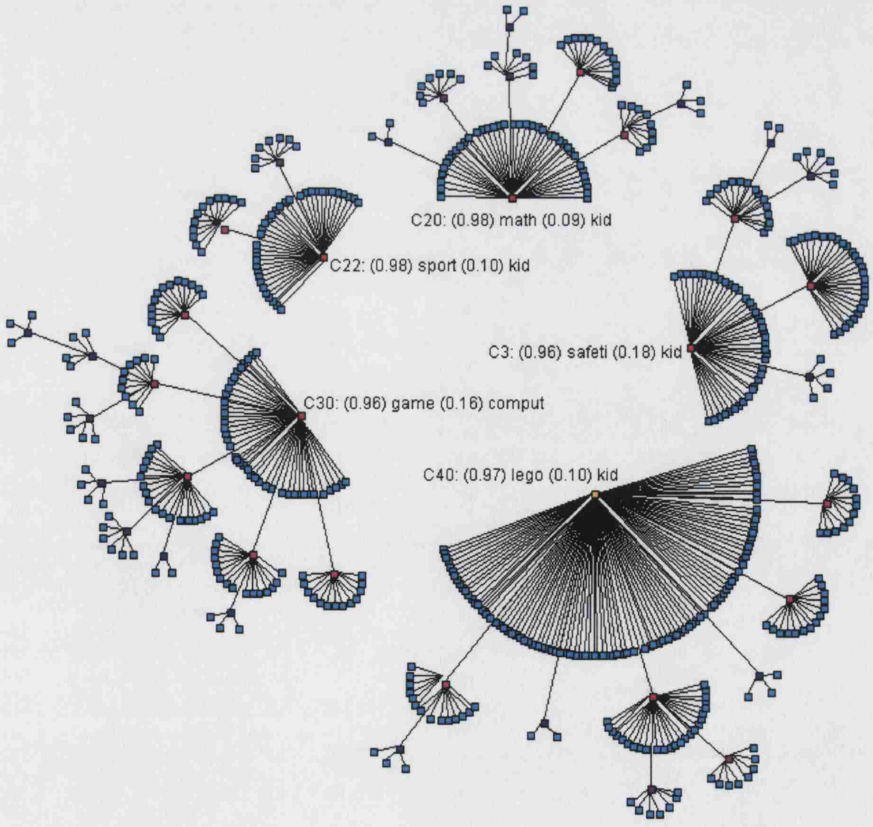


Figure 6.7: Graph visualisation of the ODP cluster hierarchy (c=40).

The complete topic hierarchy is presented in Figure 6.8. Again, the picture only displays the top terms of the centroid vectors, whose weights contribute to 80% of the centroids length. It can be observed that clusters which are more specific, *i.e.* that are located deeper in the hierarchy, are described by a higher number of terms. From this observation we conclude that the asymmetric similarity measure applied in the algorithm indeed allows to identify sub-topics at each level of the hierarchy.

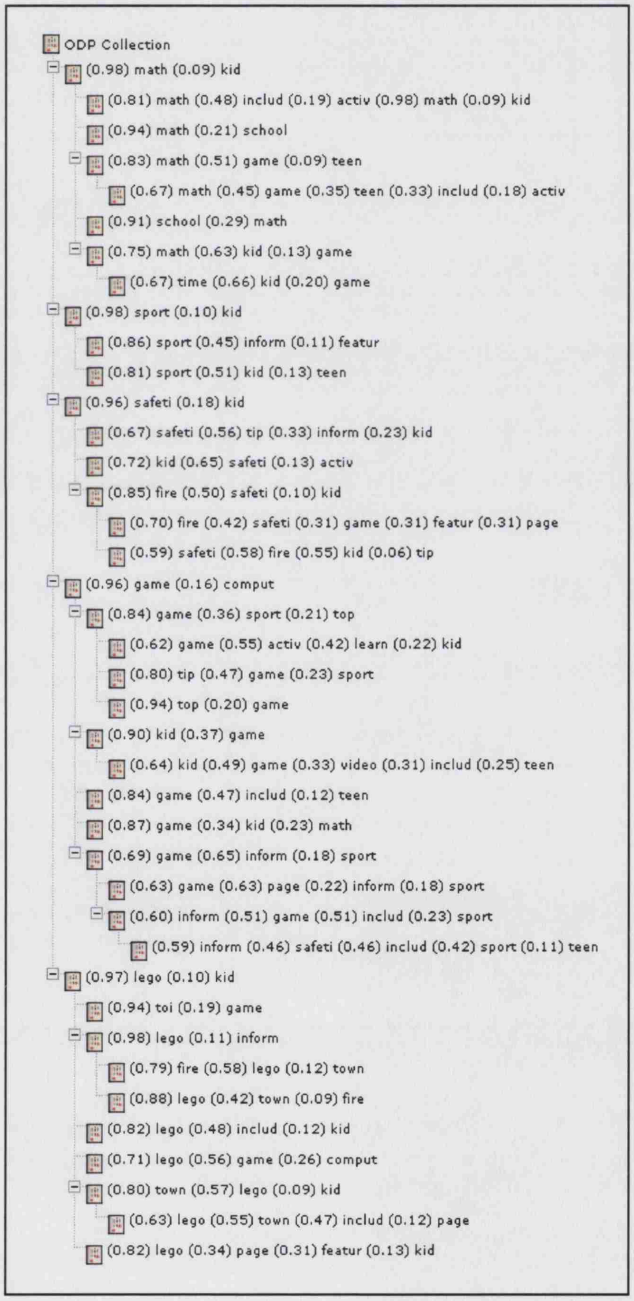


Figure 6.8: Topic hierarchy formed by the cluster centroids for the ODP collection (c=40).

6.5 Summary

In this chapter we have proposed a novel hierarchical fuzzy clustering method – the H²-FCM algorithm – for dynamically organising documents into a topic hierarchy. The new algorithm applies the H-FCM algorithm to obtain a fuzzy partition of the document set and it links the resulting cluster centroids hierarchically based on the notion of asymmetric similarity.

We have evaluated the quality of the H²-FCM clustering results through an analysis of clustering precision and recall. We have shown that as the number of clusters increases, the H-FCM (which is applied in the first stage of the H²-FCM algorithm) splits related documents into a higher number of smaller and more specific clusters. We have also shown that the resulting cluster hierarchy successfully links clusters of the same generic topic. These results support the analysis carried out in section 3.6, where it was argued that instead of attempting to find an optimum number of clusters, the choice of c should reflect the desired granularity of the document clusters.

We have also analysed the computational cost of the H²-FCM algorithm and we have verified that the order of its time complexity is quadratic with the number of clusters and linear with the number of documents and with the number of dimensions. Thus, the new algorithm scales well with the number of documents and hence, it is efficient for clustering and organising large document repositories into a topic hierarchy.

Finally, we have shown how topic hierarchies are obtained based on the information conveyed by the cluster centroids. Specifically, the weighted terms that compose each centroid vector represent the topics associated with the respective document cluster. By organising the cluster centroids hierarchically, a meaningful topic hierarchy is obtained. We have also explained the usefulness of this form of knowledge representation for exploring e-Learning material. In particular, a visualisation of the complete topic hierarchy has been suggested to assist the user to contextualise his exploration of the knowledge space.

Chapter 7
Concluding Remarks

The motivation for the research work presented in this thesis has emerged in the *e*-Learning context, where there is the need for flexible learning environments. Among other flexibility aspects, *e*-Learning systems should enable flexible interactions with the learning material. In particular, both exploratory and research oriented interactions should be facilitated. This, in turn, requires an organisation of the available learning material into an abstract knowledge space to assist the users in their exploration.

This thesis is concerned with an analysis of fuzzy document clustering for dynamic knowledge representation, with particular application in the *e*-Learning context. The research presented in this thesis demonstrated that with fuzzy clustering techniques meaningful document associations can be discovered and that this form of knowledge representation can be used for flexible browsing of *e*-Learning material and for exploring topics of a given knowledge domain.

The investigation started with a characterisation of the problem space to decide on a similarity measure and on a fuzzy clustering algorithm for document clustering. In Chapter 3, the properties of heterogeneous document collections were analysed and it was verified that document vector representations are typically high-dimensional and very sparse. Several similarity coefficients (inner product, cosine, Dice, Jaccard and overlap coefficients) were considered for measuring inter-document relationships. An analysis of similarity distributions showed that documents are typically very dissimilar to each other, mainly due to their sparsity. However, a close examination of both intra- and inter-class similarity distributions revealed that there is a clear separation between the two distributions, thereby allowing the discovery of a clustering structure. It was verified that with the cosine

similarity coefficient the separation between intra- and inter-class similarity distributions was maximised, which indicated that this was a good similarity measure for document clustering. It was also verified that better separation was achieved when document vectors were normalised to unit length and when documents were encoded with the TF term weighting scheme rather than with the TF-IDF scheme.

The selection of a fuzzy clustering algorithm was also considered in Chapter 3. We decided to use the well-known FCM algorithm as the basis for document clustering, for its simplicity and linear time complexity, and also for being the fuzzy counterpart of a traditional document clustering method, the *k*-Means. The limitations of using the Euclidean distance as a measure of inter-document relationship were identified and the replacement of this metric by similarity coefficients in the FCM algorithm was proposed. Novel mathematical expressions for computing the cluster centroids were developed accordingly. Three similarity coefficients were considered: cosine, Jaccard and overlap coefficients. The Hyper-spherical Fuzzy c-Means (H-FCM) algorithm was the result of these developments. The characteristics of the new algorithm in the high-dimensional sparse space were also considered. In particular, two issues were discussed: the handling of outliers and the selection of the number of clusters. Given the properties of the problem space, it was concluded that outliers have little impact on the final location of the cluster centroids and that finding an optimum number of clusters is not a key issue. It was shown that regardless of the number of clusters, a clustering structure emerges. It was thus concluded that the number of clusters should reflect the required granularity of the document clusters.

The performance of the H-FCM algorithm was evaluated through the research experiments presented in Chapter 4. The main goal of the investigation was: i) to determine whether the H-FCM algorithm was able to discover valid document clusters and meaningful associations between related documents and ii) to compare the new algorithm with the FCM and also with commonly used hard clustering algorithms. The experiments were carried out with four test document collections: two subsets of the Reuters-21578 text categorisation collection (REUTERS1 and REUTERS2), a subset of the ODP metadata files and a set of scientific abstracts from the INSPEC database. An assessment of the clustering quality was possible because these collections were pre-classified into a known set of reference classes. Both internal and external performance measures were used to evaluate the H-FCM algorithm. In particular, Partition Entropy and Xie-Beni index were applied to

assess the intrinsic properties of the document clusters, namely the fuzziness, compactness and separation of the clusters. Clustering precision, clustering recall and *F*-measure were applied to assess the actual quality of the document clusters by considering the reference classes of each collection.

The experiments demonstrated that the H-FCM with the cosine coefficient clearly outperforms the FCM. The new algorithm consistently produced the best compromise between clustering precision and clustering recall, which means it generated document clusters of higher quality. It was also verified that the clustering results were dependant on the user-defined fuzzification parameter m and that the H-FCM was able to find good clusters for higher m values than the FCM. A suitable range for the m parameter was determined empirically: it was observed that $m \leq 1.25$ led to good clustering performances with all collections.

The performance of the H-FCM algorithm was also analysed in two cases: when the document vectors were encoded with i) TF and ii) TF-IDF term weighting schemes. The clustering results confirmed that the H-FCM performance was much better with the TF scheme. The differences in performance were investigated and it was concluded that the poor performance of TF-IDF was because this scheme considerably de-emphasised term weights corresponding to the main topics of the reference document clusters. The TF-IDF scheme in commonly used IR systems, where the aim is to retrieve specific documents in response to the user's queries. Since TF-IDF is good at discriminating documents from each other, this scheme suits the IR purpose. Many document clustering applications have indeed used the TF-IDF scheme, essentially due to its performance in IR. However, in this thesis it was shown that for document clustering purposes this is not the best weighting scheme.

An experiment to assess the impact of pre-processing the document vectors on the H-FCM performance was also carried out. The results showed that discarding common terms led to a decrease of the H-FCM clustering performance due to the elimination of high-frequency terms that represented known topics in the document collections. The results also showed that discarding very specific terms (*i.e.* that only appeared in very few documents) had no negative impact on the H-FCM clustering performance, while reducing significantly the dimensionality of the problem space.

The performance of the H-FCM algorithm with the Jaccard and overlap similarity coefficients was also investigated. A comparative analysis between cosine, Jaccard and overlap coefficients showed that the clustering quality was generally better when the cosine

coefficient was used in the algorithm. It was demonstrated that the differences in clustering performance were due to differences in behaviour of the similarity coefficients in sparse high-dimensional spaces, where the cosine measure presents higher similarity patterns. The investigation confirmed that when the sparsity of the document vectors was reduced by filtering out very specific terms, the H-FCM performance with the Jaccard coefficient increased and was comparable to the cosine case. The investigation also confirmed that the overlap coefficient is unsuitable for clustering sparse high-dimensional data sets.

A comparison between H-FCM and hard-clustering methods commonly used for document clustering was the subject of the last performance experiment. Three clustering algorithms were considered: the *k*-Means and two agglomerative hierarchical methods, CL and GA. The H-FCM clustering results were analysed for several α-cuts of the fuzzy partition matrix. It was verified that there was always a threshold α for which H-FCM produced the highest quality document clusters, thus outperforming the hard clustering methods. It was also demonstrated that the H-FCM clustering precision and clustering recall could be adaptively controlled by the threshold α: a higher value favours precision whereas a lower value favours recall. In the *e*-Learning context, higher precision is important for users whose aim is to access learning material on specific topics, whereas higher recall is important for enabling the exploration of less obvious relationships between learning material on loosely related topics. Hence, with fuzzy clustering a range of pedagogical approaches for *e*-Learning can be supported.

The H-FCM document clustering process was integrated in a prototype tool, the Knowledge Navigator, for flexible browsing of *e*-Learning material. The implementation of this tool and its deployment in a real *e*-Learning environment was the subject of Chapter 5. The details on how a knowledge space representation can be derived from the H-FCM fuzzy clustering outcome were given. It was explained how the associations between related documents and related topics can be extracted. In particular, the relevance of a given document to other documents was derived from the fuzzy membership values and topic associations were derived from the latent relationships between terms co-occurring in the cluster centroids. The Topic Maps standard was used for modelling those associations in XML, thus providing a formal representation for the knowledge space that is suitable for Web-based applications. Java tools were implemented for the dynamic generation of the clustering topic map and for interpreting and browsing the knowledge space associations defined therein. The mechanisms for Web-based navigation of the clustering topic map

were implemented based on Java Servlet and XSLT technologies. These mechanisms are at the core of the Knowledge Navigator tool. The user interface of this prototype consists of a simple HTML frame set that allows visualising the knowledge space relationships as hyper-links. The HTML pages are generated dynamically by the servlets in response to the user's path. The Knowledge Navigator was deployed and evaluated in a real *e*-Learning system where user trials were carried out in the context of the CANDLE project. The trials revealed that having links to relevant material was indeed a good feature of the tool and that topic-based navigation was quite useful.

The development of a scalable hierarchical fuzzy clustering algorithm for flexible exploration of the knowledge space was the subject of Chapter 6. The new algorithm, the Hierarchical Hyper-spherical Fuzzy c-Means (H^2-FCM), builds upon the H-FCM and an asymmetric similarity measure to generate a cluster hierarchy. This development explored the characteristics of the H-FCM algorithm in the sparse high-dimensional document space, where the number of clusters can be defined according to the required granularity of the topics represented by each cluster. The new algorithm applies the H-FCM to obtain a fuzzy partition of the document set into a sufficiently large number of clusters and it takes an heuristic approach based on the asymmetric similarity between cluster centroids to link hierarchically the resulting clusters. The performance of the H^2-FCM algorithm was evaluated considering: i) the quality of each individual cluster and ii) the efficiency of the linking procedure, as the number of clusters increased. The same test document collections were used in the investigation. It was shown that clustering precision and clustering recall were essentially constant for any number of clusters and that the algorithm successfully linked clusters of the same reference topic. The computational costs of the H^2-FCM algorithm were also analysed and it was verified that the order of its time complexity is quadratic with the number of clusters and linear with the number of documents and with the number of dimensions, which makes it scalable for large document sets. It was also shown how the information captured by the cluster centroids could be used to generate a topic hierarchy.

7.1 Recommendations for future research

Based on the work presented in this thesis, there are a number of areas that can be suggested for future research. Topics worthy of further investigation can be divided into issues related to the *e*-Learning application and issues related to the clustering algorithms, their performance and applications. On the *e*-Learning application side, the following topics are worthy of further investigation:

- **Adapting the Knowledge Navigator to the learner's needs**: the navigation of *e*-Learning material by exploring the fuzzy knowledge space relationships has been demonstrated with a prototype tool, the Knowledge Navigator. Further work can address the development of a link adaptation mechanism that would use the fuzzy knowledge representation. Such mechanism would consider the relationships in the fuzzy knowledge space and the learning context (*i.e.* the learners's knowledge level, learning objectives, pedagogical model, etc.) to determine which set of links would be the most relevant to a particular learner.
- **Improving the Knowledge Navigator interface**: the prototype Knowledge Navigator tool provides a simple user interface to visualise and browse the knowledge space relationships defined in the topic map, which are represented as HTML hyper-links. However, several aspects of this tool could be improved. Currently, the entry point to the knowledge space is provided by a very basic keyword search mechanism and once the user starts browsing the knowledge space, there is no way of knowing the complete set of topics that are represented in the topic map. A visualisation of the whole knowledge space could be included: i) to allow the user to initiate his exploration of the knowledge space through the visual selection of a particular topic and ii) to contextualise the user's navigation path. Such visualisation could for instance consist of a representation of the complete cluster hierarchy generated by the H^2-FCM algorithm as an interactive graph. Another feature that could be included in the Knowledge Navigator tool would be allowing the user to control the number of relevant document links that are displayed (favouring either precision or recall), *i.e.* instead of having a fixed α-cut for the fuzzy clusters, the user could have a sliding bar to adaptively set the membership threshold.

- **User trials**: the user study carried out in the context of the CANDLE project was mainly focused on the usefulness of the having related topics and related documents associated for flexible exploration of *e*-Learning material. It would be interesting to carry out more extensive user studies including: i) an analysis of the impact the user interface on the perceived usefulness of the knowledge space representation and ii) an evaluation of the usefulness of the H^2-FCM algorithm for knowledge discovery in the *e*-Learning context and in other applications.

On the clustering algorithms side, the following topics are worthy of further investigation:

- **Context-based clustering**: in some applications one might want to bias the H-FCM/H^2-FCM clustering output towards a particular set of topics of interest. For example, in the *e*-Learning context an user could define a set of keywords to represent his interests and a customised view of the knowledge space could be provided accordingly. Instead of having an adaptation mechanism that considers the complete knowledge space to determine link relevance, the knowledge space relationships could be emphasised or de-emphasised by the clustering algorithm itself. Following a similar approach to that presented in [105], a context variable could be included in the clustering process to provide the bias towards some dimensions of the k-dimensional document space. Such variable is expected to impact on the final location of the cluster centroids and thus, on the knowledge space representation. Future work can investigate an extension of the H-FCM algorithm to provide context-based clustering based on such context variable.
- **Using syntactic and statistical information to index documents**: in this thesis the bag-of-words approach has been followed for indexing documents, where relevant single words have been extracted from the text of the documents and used as indexing terms. Other approaches for indexing documents include the use of syntactic and statistical information, where phrases or groups of words are selected as indexing features. An investigation into the performance of the H-FCM algorithm with the more elaborate indexing approaches should be carried out to determine whether from a clustering perspective there are any advantages of representing documents with multi-word features instead of single words. Those indexing approaches could be also investigated from an *e*-Learning perspective to determine whether a better knowledge representation is obtained.

- **Applying the H-FCM/H²-FCM to index and to summarise documents:** our research experiments have been carried out with test document collections presenting heterogeneous properties, such as the average document length. In particular, our results have revealed that the H-FCM performed very well with the ODP collection, that presented an average document length of 15.14 words. Considering that longer documents can be segmented into sub-parts containing few words (*e.g.* sections, paragraphs or sentences), and that each of those segments can be represented as a k-dimensional vector of indexing terms following the same process as before, the H-FCM algorithm can be applied to cluster the segments of a given text document. It is likely that the weighted terms of the cluster centroid vectors will reveal the main topics covered by the document. Those topics could then be used to index the document as well as to extract key sentences from the document for summarising its contents. Key sentences could be identified based on a sentence weighting scheme that would determine which sentences contained information related to the main document topics [150]. Whether or not the H-FCM is able to discover the topics of a given document through segment clustering should be subject of further investigation. Furthermore, as this indexing approach could be applied in the document clustering process, the impact on the document clustering performance when each individual document is indexed in this way should also be considered in future work.
- **Incremental clustering:** the H-FCM algorithm has been applied to cluster the whole set of documents from a given collection. For dynamic repositories it should not be necessary to re-cluster the entire set of documents every time a new document is included. For example, a possible approach is as follows: the similarity of the new document to every cluster centroid could be calculated and the degree of membership could be obtained using the H-FCM equations. The new document could then be inserted in the clusters where the membership value exceeded a given threshold α. Further research can investigate whether this is in fact a feasible approach to scale the algorithm to large and dynamic document sets.

7.2 Summary

Overall, this thesis has evaluated the performance of fuzzy clustering for dynamic knowledge representation, with particular application in *e*-Learning systems, through the discovery of topic relationships in text documents. New fuzzy clustering algorithms have been developed and their performance has been evaluated through a range of document clustering experiments. It was shown that fuzzy clustering succeeds in finding meaningful document associations based on the documents contents. This thesis has also demonstrated that the proposed knowledge representation approach enables exploratory interactions with learning material for flexible *e*-Learning. Although the research work presented in this thesis was motivated by the *e*-Learning context, the methods and tools developed have wider applicability.

Appendix A
Lagrange multipliers

In this appendix the method of the Lagrange multipliers is described. Such method is used to solve constrained optimisation problems, usually specified in terms of equality and inequality constraints [151]. Here we focus only on equality constraints like those found in the H-FCM optimisation problem (see section 3.5).

Given an objective function $f(x)$ of n variables to be maximised or minimised,

$$f(x) = f(x_1, x_2, ..., x_n) \tag{A.1}$$

subject to k constraints $g_i(x)$ for which is known,

$$g_1(x) = C_1, \quad g_2(x) = C_2, \quad \ldots, \quad g_k(x) = C_k \tag{A.2}$$

where C_i are constants, a new function $L(x)$ is defined by introducing k new parameters λ_i, that are called Lagrange multipliers,

$$L(x) = f(x) + \sum_{i=1}^{k} \lambda_i \cdot \left(g_i(x) - C_i\right) \tag{A.3}$$

The optimisation of (A.1) subject to the constraints in equation (A.2) is performed by differentiating $L(x)$ with respect to the unknown variables and parameters, which consists in solving the following $n+k$ equations,

$$\frac{\partial}{\partial x_j} L(x) = \frac{\partial}{\partial x_j}\left(f(x) + \sum_{i=1}^{k} \lambda_i \cdot \left(g_i(x) - C_i\right)\right) = 0, \qquad 1 \le j \le n \tag{A.4}$$

$$\frac{\partial}{\partial \lambda_i} L(x) = 0 \Leftrightarrow g_i(x) = C_i, \qquad 1 \le i \le k \tag{A.5}$$

This method of the Lagrange multipliers basically converts constrained optimisation problems into unconstrained ones, by searching for maxima or minima of the objective function only among points that satisfy the existing constraints.

Appendix B
Fuzzy clustering Topic Map template

This appendix contains the fuzzy clustering XML Topic Map template which has been defined as follows.

```
<!-- TOPIC TYPES -->

<!-- "CLUSTER" TYPE DECLARATION -->

<topic id="tm-cluster">
   <baseName>
       <baseNameString>Document group</baseNameString>
   </baseName>
</topic>

<!-- "TERM" TYPE DECLARATION -->

<topic id="tm-term">
   <baseName>
       <baseNameString>Term</baseNameString>
   </baseName>
</topic>

<!-- "DOCUMENT" TYPE DECLARATION -->

<topic id="tm-document">
   <baseName>
       <baseNameString>Document</baseNameString>
   </baseName>
</topic>

<!-- END OF TOPIC TYPES -->
```

```
<!-- OCCURRENCE TYPES -->

<topic id="resource-title">
   <baseName>
      <baseNameString>Title</baseNameString>
   </baseName>
</topic>

<topic id="resource-url">
   <baseName>
      <baseNameString>Url</baseNameString>
   </baseName>
</topic>

<topic id="resource-description">
   <baseName>
      <baseNameString>Description</baseNameString>
   </baseName>
</topic>

<topic id="resource-text">
   <baseName>
      <baseNameString>Full text</baseNameString>
   </baseName>
</topic>

<topic id="resource-hierarchy">
   <baseName>
      <baseNameString>Hierarchy</baseNameString>
   </baseName>
</topic>

<!-- END OF OCCURRENCE TYPES -->

<!-- TOPIC ROLE TYPES -->

<topic id="tm-cluster1">
   <baseName>
      <baseNameString>Document group</baseNameString>
   </baseName>
</topic>

<topic id="tm-cluster2">
   <baseName>
      <baseNameString>Document group</baseNameString>
   </baseName>
</topic>

<topic id="tm-weight">
   <baseName>
      <baseNameString>Weight</baseNameString>
   </baseName>
</topic>

<topic id="tm-membership">
```

```
    <baseName>
        <baseNameString>Membership</baseNameString>
    </baseName>
</topic>

<topic id="tm-fuzzy-relation">
    <baseName>
        <baseNameString>Fuzzy relation</baseNameString>
    </baseName>
</topic>

<!-- END OF TOPIC ROLE TYPES -->

<!-- ASSOCIATION TYPES -->

<topic id="cluster-term">
    <baseName>
        <baseNameString>Cluster-Term relation</baseNameString>
    </baseName>
    <baseName>
        <scope>
            <topicRef xlink:href="#tm-cluster"/>
        </scope>
        <baseNameString>Related document groups</baseNameString>
    </baseName>
    <baseName>
        <scope>
            <topicRef xlink:href="#tm-term"/>
        </scope>
        <baseNameString>Descriptive topics</baseNameString>
    </baseName>
</topic>

<topic id="cluster-document">
    <baseName>
        <baseNameString>Cluster-Document relation</baseNameString>
    </baseName>
    <baseName>
        <scope>
            <topicRef xlink:href="#tm-cluster"/>
        </scope>
        <baseNameString>Documents in this group</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-document"/></scope>
        <baseNameString>Document group memberships</baseNameString>
    </baseName>
</topic>

<topic id="term-document">
    <baseName>
        <baseNameString>Term-Document relation</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-term"/></scope>
```

```
        <baseNameString>Documents containing this term</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-document"/></scope>
        <baseNameString>Related terms</baseNameString>
    </baseName>
</topic>

<topic id="term-term">
    <baseName>
        <baseNameString>Term-Term relation</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-term"/></scope>
        <baseNameString>Related terms</baseNameString>
    </baseName>
</topic>

<topic id="cluster-cluster">
    <baseName>
        <baseNameString>Cluster-Cluster relation</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-cluster1"/></scope>
        <baseNameString>The closest group to this is</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-cluster2"/></scope>
        <baseNameString>This group is the closest of</baseNameString>
    </baseName>
</topic>

<topic id="document-document">
    <baseName>
        <baseNameString>Document-Document relation</baseNameString>
    </baseName>
    <baseName>
        <scope><topicRef xlink:href="#tm-document"/></scope>
        <baseNameString>Related documents</baseNameString>
    </baseName>
</topic>

<!-- END OF ASSOCIATION TYPES -->
```

Appendix C

Questionnaires

This appendix contains the questionnaires used for the evaluation of the Knowledge Navigator tool.

Questionnaire 1

Username: nsm____
Password: _____

Note: *This questionnaire can be done anonymously and it will ONLY be used to evaluate the Knowledge Navigator tool. It will not affect your assignment marking in any way. You can find above the username and password that will allow you to login in the system.*

1. How many hours did you spend doing revision of the whole NSM course? _____

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q1	50	45	70	50	40	51

2. How many hours did you use the Knowledge Navigator tool for revision? _____

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q2	2	5	2	2	4	3

Note: *In the following questions you will be asked to rate some aspects of the tool on a scale of 1 to 5. The specific meaning of the scale will be detailed for each question.*

3. Please rate on the scale provided how useful you found the navigation tool for:

	(1: not useful; 5: very useful)	Not used
a) finding a specific document,	1 2 3 4 5	☐
b) finding documents related to a specific topic,	1 2 3 4 5	☐
c) finding documents related to a specific document.	1 2 3 4 5	☐

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q3a)	3	4	5	1	4	3.4
Q3b)	2	5	5	4	4	4
Q3c)	-	4	5	4	4	4.25

4. In general, did you find the links in the navigation tool relevant to find what you were looking for? (1: not relevant; 5: very relevant)
1 2 3 4 5

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q4	3	4	2	4	4	3.4

5. Please indicate which features you used to find your documents and rate each one on the scale provided:

	(1: rarely; 5: very often)	Not used
a) browsing by document group,	1 2 3 4 5	☐
b) browsing by topics,	1 2 3 4 5	☐
c) browsing by related documents,	1 2 3 4 5	☐
d) keyword search.	1 2 3 4 5	☐

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q5a)	-	1	3	-	2	2
Q5b)	4	3	3	-	4	3.5
Q5c)	3	4	4	5	2	3.6
Q5d)	4	5	5	5	5	4.8

6. Did you find the document weights/sorting useful?
(1: not useful; 5: very useful)
1 2 3 4 5

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q6	4	3	2	3	1	2.6

7. During your navigation, did you come across other relevant document(s) you were not initially looking for?
(1: rarely; 5: very often)
1 2 3 4 5

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q7	4	4	1	3	5	3.4

8. By looking at the metadata were you able to distinguish which document was worth opening and which was not?

(1: never; 5: always)
1 2 3 4 5

Student	# nsm45	# nsm31	# nsm23	# nsm12	# nsm46	Average
Q8	3	3	3	4	2	3

9. Based on your usage of the tool, what was the best feature for supporting your revision?

- grouping of documents (1 user)
- display of relevant documents (3 users)
- quick search (1 user)
- online access (1 user)
- the idea of the tool (1 user)
- helped to find definition of terms without the need to browse through each page in the printout (1 user)
- the classification of all the topics (1 user)

10. And what feature of this tool would you change to enhance your revision ?

- for revision PDF files were not found useful (1 user)
- make it easier to a find specific document (1 user)
- make it easier to use, since it is not clear at first glance (1 user)
- search engine (1 user)

11. Would you like to use this kind of tool to access content from other courses and institutions and how useful would it be? Yes/No (*Please circle your answer*)

- one user thought that the tool should provide external material and not main documents of the course so he/she was not in favour
- another user said it would depend on the type of data, which should preferably be more structured
- others said they were in favour because:
  - the tool was helpful in locating documents of interest efficiently
  - the tool could be much more useful with more data to search for
  - the tool would be useful as a complementary tool for revision

12. Any other comments?

General comments about the tool stressed that the content available was not an incentive to use the tool because it was the same as the printed handouts. They support the inclusion of external and complementary material.

Questionnaire 2 (CANDLE)

1. What learning materials have you used for the revision? And (roughly) how much of the time? (as a % of total time)

	Student					Average % of time
	#1	#2	#3	#4	#5	
Course materials provided – online	5	10	100	5	5	25
Course materials provided – printed	30	30	100	50	65	55
Recommended textbooks	20	5	0	5	10	8
Other textbooks	30	10	0	10	0	10

Other online materials – *please specify*	Student				
	#1	#2	#3	#4	#5
IETF	✓	✓		✓	
ETSI		✓	✓		✓
Google	✓	✓			
IEC		✓		✓	
BT		✓			
Cisco web tutorials				✓	
IEE	✓				
Dr. Lionel Sacks website		✓			
Network Magazine		✓			
RECs		✓			
SMARTDRAW site		✓			

2. How frequently did you make use of the navigation tool?

(not at all) 1 2 3 4 5 (all the time)

Student	#1	#2	#3	#4	#5	Average score
Score	2	1	2	-	2	1.75

3. If you have used other online materials, what search and navigation tools did you use to:

a) find materials?

Student				
#1	#2	#3	#4	#5
Google, Yahoo	Portals, search engines, specific sites	Google, Yahoo	Search engines	Google, Internet Explorer

Descriptive stats	Response
100%	Search engines
60%	Google
40%	Yahoo
20%	Portals

b) navigate / search materials?

Student				
#1	#2	#3	#4	#5
Acrobat Reader	Internet Explorer (PDF, HTML)	Google, Internet explorer		Search in PDF reader

Descriptive stats	Response
60%	Acrobat Reader
40%	Internet Explorer

4. Thinking about your general experience with computer-based navigation tools, are there any features of the navigation tool which you think are:

a) particularly good?

#1 Metadata/document weight.
#2 Related documents list.
#3 Good idea.
#4 The correlation of particular terms.
#5 Keyword search and having related documents saves a lot of time.

b) particularly bad?

#1 It's too messy to have to save every file, but cannot just right click and open (although for the latter one, files are saved in temp, but it is more reasonable).

#2 -
#3 Not very straightforward to use, and there were not enough things to search for. I think it should be tested with a much greater pool of data (*e.g.* whole UCL website).
#4 There is no search engine.
#5 The purpose of document grouping was not very useful.

5. What was the *most useful* aspect of using the tool as part of your revision studies?

#1 To find an outline that we can't search for by a flipping-through method.
#2 Listing of related documents, keyword search.
#3 It contained all the notes, so I didn't have to look them up.
#4 It helped me to group some topics of network management together and classify them in my mind.
#5 See Q4.

6. What was the *least useful* aspect of using the tool as part of your revision studies?

#1 Documents provided are exact the same in the handout. So there's no point to use CANDLE when it is quicker to use the hardcopy version.
#2 Not available outside EE lab.
#3 Didn't find it easier than looking up my notes!
#4 Losing time to search for more specialised subjects which I didn't even know if they included.
#5 See Q4.

7. Any other comments?

#1 CANDLE will be more practical if the notes online are different, useful than the normal noted. For example if CANDLE can provide SNMP, MIB, CORBA in details. This would encourage students to use it, rather than letting them searching for what they already have.
#2 -
#3 I think it wasn't easier because the notes weren't that big. It would be much more helpful if there was more data to search on.
#4 -
#5 Overall it saved time in searches for definitions but it did not help me to understand the subject.

References

[1] IEEE Learning Technology Standards Committee (LTSC): http://ltsc.ieee.org/

[2] D. Zang, "Powering E-Learning In the New Millennium: An Overview of e-Learning and Enabling Technology", Information Systems Frontiers, vol. 5, no. 2, pp. 207-218, 2003.

[3] J. Batlogg, R. Braek, C. Fowler, I. Kermarreck, L. Sacks, J. Wetterling and L. Gutierrez, "CANDLE: an European e-education project to improve the teaching on the Internet," In: *Proceedings of the* EUNICE *Summer School* 2000, Enschede, the Netherlands, September 2000.

[4] A. Pras, "Sharing telematics courses - the CANDLE project," In: *Proceedings of the* IFIP WATM & EUNICE 2001 *Summer School*, pp. 205-212, Paris, France, September 2001.

[5] L. Sacks, A. Earle, O. Prnjat, W. Jarrett and M. Mendes, "Supporting variable pedagogical models in network based learning environments," In: *Proceedings of the* IEE 2nd *Annual Symposium on Engineering Education: Professional Engineering Scenarios*, ref. no. 02/056, vol. 1, pp. 22/1-22/6, London, UK, January 2002.

[6] M. Dimitrova, C. Sadler, S. Hatzipanagos and A. Murphy, "Addressing learner diversity by promoting flexibility in e-Learning environments," In: *Proceedings of the* 14th *International Workshop on Database and Expert Systems Applications*, DEXA'03, pp. 287-291, Prague, Czech Republic, September 2003.

[7] L. Calvi and P. de Bra, "Improving the usability of hypertext courseware through adaptive linking," In: *Proceedings of the* 8th ACM *Conference on Hypertext and Hypermedia*, HT'1997, pp. 224-225, April 1997.

[8] D.P. da Silva, R. van Durm, E. Duval and H. Olivié, "A simple model for adaptive courseware navigation," In: *Proceedings of* INFWET '97, Canada, November 1997.

[9] P. Brusilovsky, J. Eklund and E. Schwarz, "Web-based education for all: a tool for developing adaptive courseware," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp. 291-300, Apr. 1998.

[10] IMS Global Learning consortium, Inc.: http://www.imsglobal.org/

[11] ARIADNE Foundation for the European Knowledge Pool: http://www.ariadne-eu.org/

[12] "IEEE Standard for Learning Object Metadata," IEEE Standard for Learning Technology 1484.12.1-2002, ISO/IEC 11404, 2002.

[13] T. Bray, J. Paoli, C.M. Sperberg-McQueen and E. Maler (editors), "eXtensible Markup Language (XML) 1.0", W3C Recommendation, October 2000.

[14] J. Milstead, S. Feldman, "Metadata: cataloging by any other name," *Online Magazine*, vol. 23, no. 1, January 1999.

[15] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 5-43, May 2001.

[16] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann and I. Horrocks, "The semantic web: the roles of XML and RDF," IEEE *Internet Computing*, vol. 4, no. 5, pp. 63-73, September/October 2000.

[17] S.M. Cherry (2002), "Weaving a web of ideas," IEEE *Spectrum*, vol. 39, no. 9, pp. 65 - 69, September 2002.

[18] J.C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.

[19] B. Everitt. Cluster Analysis. Third Edition. Edward Arnold, London, 1993.

[20] P. Willett, "Recent trends in hierarchical document clustering: a critical review," *Information Processing & Management*, vol. 24, no. 5, pp. 577-597, 1988.

[21] M.R. Anderberg. Cluster analysis for applications. Academic Press, New York, 1973.

[22] A.K. Jain, M.N. Murty and P.J. Flynn, "Data clustering: a review," ACM *Computing Surveys*, pp. 264-323, vol. 31, no. 3, September 1999.

[23] K. Sparck Jones. Automatic Keyword Classification. Butterworth, London, 1971.

[24] G. Salton, "On the use of term associations in automatic information retrieval," In: *Proceedings of the* 11th *International Conference on Computational Linguistics*, COLING'86, pp. 380-386, Bonn, Germany, August 1986.

[25] H.J. Peat and P. Willett, "The limitations of term co-occurrence data for query expansion in document retrieval systems," *Journal of the American Society for Information Science*, pp. 378-383, vol. 42, no. 5, June 1991.

[26] N. Jardine and C.J. van Rijsbergen, "The use of hierarchical clustering in information retrieval," *Information Storage and Retrieval*, vol. 7, pp. 217-240, 1971.

[27] C.J. van Rijsbergen. Information Retrieval. Second Edition, Butterworth, London, 1979.

[28] E.M. Voorhees, "The cluster hypothesis revisited," In: *Proceedings of the* 8th *Annual International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'85, pp. 188-196, Montreal, Canada, 1985.

[29] W.M. Shaw Jr., R. Burgin and P. Howell, "Performance standards and evaluations in IR test collections: cluster-based retrieval models," *Information Processing & Management*, vol. 33, no. 1, pp. 1-14, January 1997.

[30] M.A. Hearst and J.O. Pedersen, "Reexamining the cluster hypothesis: Scatter/Gather on retrieval results," In: *Proceedings of the* 19th *Annual International* ACM SIGIR

Conference on Research and Development in Information Retrieval, SIGIR'96, pp. 76-84, Zurich, Switzerland, August 1996.

[31] O. Zamir and O. Etzioni, "Web document clustering: a feasibility demonstration," In: *Proceedings of the* 21th *Annual International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'98, pp. 46-54, Melbourne, Australia, August 1998.

[32] A. Leuski, "Evaluating document clustering for interactive information retrieval," In: *Proceedings of the Tenth International* ACM *Conference on Information and Knowledge Management*, CIKM 2001, pp. 30-40, Atlanta, USA, November 2001.

[33] A. Schenker, M. Last and A. Kandel, "A term-based algorithm for hierarchical clustering of Web documents," In: *Proceedings of the Joint* 9th IFSA *World Congress and* 20th NAFIPS *International Conference*, vol.5, pp. 3076-3081, Vancouver, Canada, July 2001.

[34] Y. Wang and M. Kitsuregawa, "Link based clustering of Web search results," *Lecture Notes in Computer Science*, vol. 2118, pp. 225-236, July 2001.

[35] D.B. Crouch, C.J. Crouch and G. Andreas, "The use of cluster hierarchies in hypertext information retrieval," In: *Proceedings of* ACM *Hypertext* '89, pp. 225-237, Pittsburgh, USA, November 1989.

[36] D.R. Cutting, D.R. Karger, J.O. Pederson and J.W. Tukey, "Scatter/gather: a cluster-based approach to browsing large document collections," In: *Proceedings of the* 15th *Annual International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'92, pp. 318-329, Copenhagen, Denmark, June 1992.

[37] M.A. Hearst, D.R. Karger and J.O. Pedersen, "Scatter/gather as a tool for the navigation of retrieval results," In: *Papers from the* AAAI *Fall Symposium* AI *Applications in Knowledge Navigation and Retrieval*, Technical Report FS-95-03, pp. 65-71, Cambridge, USA, November 1995.

[38] J.R. Wen, J.Y. Nie and H.J. Zhang, "Query clustering using user logs," ACM *Transactions on Information Systems*, vol. 20, no. 1, pp. 59-81, January 2002.

[39] A. Visa, "Technology of text mining," In: *Proceedings of the Second International Workshop on Machine Learning and Data Mining in Pattern Recognition*, MLDM 2001, pp. 1-11, Leipzig, Germany, July 2001.

[40] E. Rasmussen. Clustering algorithms. In: W.B Frakes and R. Baeza-Yates (editors). Information Retrieval: Data Structures & Algorithms. Prentice Hall, New Jersey, 1992.

[41] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison Wesley, ACM Press, New York, 1999.

[42] M. Dillon and A.S. Gray, "FASIT: A fully automatic syntactically based indexing system," *Journal of the American Society for Information Science*, vol. 34, no. 2, pp. 99-108, March 1983.

[43] D.D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task," In: *Proceedings of the* 15th *International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'92, pp. 37-50, Copenhagen, Denmark, June 1992.

[44] C. Zhai, X. Tong, N. Milic-Frayling and D.A. Evans, "Evaluation of syntactic phrase indexing-CLARIT NLP track report," In: *Proceedings of the Fifth Text REtrieval Conference*, TREC-5, pp. 347-357, Gaithersburg, USA, November 1996.

[45] G. Salton, C.S. Yang and C.T. Yu, "A theory of term importance in automatic text analysis," *Journal of the American Society for Information Science*, vol. 26, no. 1, pp. 33-44, January-February 1975.

[46] J. Fagan, "Automatic phrase indexing for document retrieval: an examination of syntactic and non-syntatic methods," In: *Proceedings of the* 10th *International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'87, pp. 91-101, New Orleans, USA, June 1987.

[47] P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer and P.S. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79-85, June 1990.

[48] K.M. Hammouda and M.S. Kamel, "Phrase-based document similarity based on an index graph model," In: *Proceedings of* 2002 IEEE *International Conference on Data Mining*, ICDM 2002, pp. 203-210, Maebashi City, Japan, December 2002.

[49] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[50] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391-407, September 1990.

[51] G.K Zipf. Human Behavior and the Principle of Least Effort. Addison Wesley, Massachussetts, 1949.

[52] H.P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159-165, April 1958.

[53] C.E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October 1948.

[54] W.M. Shaw Jr., "Subject indexing and citation indexing. Part I: clustering structure in the cystic fibrosis document collection," *Information Processing & Management*, vol. 26, no. 6, pp. 705-718, 1990.

[55] W.M. Shaw Jr., "Subject and citation indexing. Part II: the optimal, cluster-based retrieval performance of composite representations," *Journal of the American Society for Information Science*, vol. 42, no. 9, pp. 676-684, October 1991.

[56] R. Burgin, "The retrieval effectiveness of five clustering algorithms as a function of indexing exhaustivity", *Journal of the American Society for Information Science*, vol. 46, no. 8, pp. 562-572, September 1995.

[57] Y. Yang and J.O. Pedersen, "A comparative study on feature selection in text categorization," In: *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML'97, pp. 412-420, 1997.

[58] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513-523, February 1988.

[59] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation*, vol. 28, no. 1, pp. 11-21, 1972.

[60] G. Salton, J. Allan and C. Buckley, "Automatic structuring and retrieval of large text files," *Communications of the* ACM, vol. 37, no. 2, pp. 97-108, February 1994.

[61] A. Molinari and G. Pasi, "A fuzzy representation of HTML documents for information retrieval systems," In: *Proceedings of the* 5th IEEE *International Conference on Fuzzy Systems*, FUZZ-IEEE'96, vol. 1, pp. 107-112, New Orleans, USA, September 1996.

[62] P. Willet, "Similarity coefficients and weighting functions for automatic document classification: an empirical comparison," *International Classification*, vol. 10, no. 3, pp. 138-142, 1983.

[63] H. Small, "Visualizing science by citation mapping," *Journal of the American Society for Information Science*, vol. 50, no. 9, pp. 799-813, July 1999.

[64] A. Popescul, G.W. Flake, S. Lawrence, L.H. Ungar and C.L. Giles, "Clustering and identifying temporal trends in document databases," In: *Proceedings* IEEE *Advances in Digital Libraries* 2000, pp. 173-182, Washington, USA, May 2000.

[65] X. He, H. Zha, C.H.Q. Ding and H.D. Simon, "Web document clustering using hyperlink structures," *Computational Statistics & Data Analysis*, vol. 41, no. 1, pp. 19-45, November 2002.

[66] D.S. Modha and W.S. Spangler, "Clustering hypertext with applications to Web searching," In: *Proceedings of the Eleventh* ACM *Conference on Hypertext and Hypermedia*, HT'2000, pp. 143-152, San Antonio, USA, May-June 2000.

[67] S. Dominich. Mathematical Foundations of Information Retrieval. Kluwer Academic Publishers, Dordrecht, Boston, London, 2001.

[68] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327-352, 1977.

[69] K. Krishna and R. Krishnapuram, "A clustering algorithm for asymmetrically related data with applications to text mining", In: *Proceedings of the Tenth International* ACM *Conference on Information and Knowledge Management*, CIKM'01, pp. 571-573, Atlanta, USA, November 2001.

[70] H. Yoshida, T. Shida and T. Kindo, "Asymmetric similarity with modified overlap coefficient among documents," In: *Proceedings of the* 2001 IEEE *Pacific Rim Conference on Communications, Computers and Signal Processing*, vol. 1, pp. 99-102, Victoria, Canada, August 2001.

[71] J.H. Ward Jr., "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236-244, March 1963.

[72] C.T. Meadow. Text Information Retrieval Systems. Academic Press, San Diego, 1992.

[73] G. Salton. Automatic Information Organization and Retrieval. McGraw-Hill, New York, 1968.

[74] M. Rorvig, "Images of similarity: a visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets," *Journal of the American Society for Information Science*, vol. 50, no. 8, pp. 639-651, June 1999.

[75] D. Boley, "Principal direction divisive partitioning," *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 325-344, 1998.

[76] M. Nilsson, "Hierarchical clustering using non-greedy principal direction divisive partitioning," *Information Retrieval*, vol. 5, no. 4, pp. 311-321, October 2002.

[77] H. Tanaka, T. Kumano, N. Uratani and T. Ehara, "An efficient document clustering algorithm and its application to a document browser," *Information Processing & Management*, vol. 35, no. 4, pp. 541-557, 1999.

[78] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," In: *Proceedings of the Eleventh International* ACM *Conference on Information and Knowledge Management*, CIKM 2002, pp. 515-524, Mclean-VA, USA, November 2002.

[79] J. MacQueen, "Some methods for classification and analysis of multivariate observations," In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, Berkeley, 1967.

[80] G. Salton. The SMART Retrieval System. Prentice Hall, Englewood Cliffs, 1971.

[81] C.C. Aggarwal, A. Hinneburg and D.A. Keim, "On the surprising behaviour of distance metrics in high dimensional space," In: *Proceedings of the* 8th *International Conference on Database Theory*, pp. 420-434, London, UK, January 2001.

[82] I.S. Dhillon and D.S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1-2, pp. 143–175, January-February 2001.

[83] A. Strehl, J. Ghosh and R.J. Mooney, "Impact of similarity measures on web-page clustering," In: *Proceedings of the* 17th *National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search*, AAAI 2000, pp. 58–64, Austin Texas, USA, July 2000.

[84] E. Forgy, "Cluster analysis of multivariate data: efficiency vs. interpretability of classifications," *Biometrics*, vol. 21, no. 3, pp. 768-780, 1965.

[85] L. Kaufman and P.J. Rousseeuw. Finding Groups in Data: an introduction to cluster analysis. Wiley, New York, 1990.

[86] J.M Peña, J.A. Lozano and P. Larranaga, "An empirical comparison of four initialization methods for the K-means algorithm," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027-1040, October 1999.

[87] L. Kaufman and P.J. Rousseeuw, "Clustering by means of medoids," In: *Proceedings of the First International Conference on Statistical Data Analysis Based on the L/sub 1/-Norm and Related Methods*, pp. 405-416, Neuchatel, Switzerland, August-September 1987.

[88] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," In: *Proceedings of the* 20th *International Conference on Very Large Data Bases*, pp. 144-155 Santiago, Chile, September, 1994.

[89] M. Steinbach, G. Karypis and V. Kumar, "A comparison of document clustering techniques", In: *Proceedings of the Sixth* ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining*, KDD-2000 *Text Mining Workshop*, Boston, USA, August, 2000.

[90] R. Dubes and A.K. Jain, "Validity studies in clustering methodologies," *Pattern Recognition*, vol. 11, no. 4, pp. 235-254, 1979.

[91] A. El-Hamdouchi and P. Willet, "Techniques for measurement of clustering tendency in document retrieval systems," *Journal of Information Science*, vol. 13, no. 6, pp. 361-365, 1987.

[92] D.D. Lewis, "Evaluating text categorization," In: *Proceedings of the* 1991 *Speech and Natural Language Workshop*, pp. 312-318, February 1991.

[93] D.D. Lewis and W.A. Gale, "A sequential algorithm for training text classifiers," In: Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'94, pp. 3-12, Dublin, Ireland, July 1994.

[94] V. Dasigi, R.C. Mann and V.A. Protopopescu, "Information fusion for text classification – an experimental comparison," *Pattern Recognition*, vol. 34, no. 12, pp. 2413-2425, December 2001.

[95] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher and J. Moore, "Partitioning-based clustering for web document categorization," *Decision Support Systems*, vol. 27, no. 3, pp. 329-341, December 1999.

[96] D.H. Kraft, J. Chen and A. Mikulcic, "Combining fuzzy clustering and fuzzy inference in information retrieval," In: *Proceedings of the* 9th IEEE *International Conference on Fuzzy Systems*, FUZZ-IEEE 2000, vol. 1, pp. 375-380, San Antonio, USA, May 2000.

[97] S. Miyamoto, "Fuzzy multisets and fuzzy clustering of documents," In: *Proceedings of the* 10th IEEE *International Conference on Fuzzy Systems*, vol. 2, pp. 1539-1542, Melbourne, Australia, December 2001.

[98] R. Krishnapuram, A. Joshi and L. Yi, "A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering," In: *Proceedings of the* 8th IEEE *International Conference on Fuzzy Systems*, FUZZ-IEEE 1999, vol. 3, pp. 1281-1286, Seoul, South Korea, August 1999.

[99] A. Joshi and Z. Jiang, "Retriever: improving web search engine results using clustering," In: A. Gangopadhyay (editor). Managing Business with Electronic Commerce: Issues and Trends. Idea Press, 2001.

[100] International Organization for Standardization (ISO), ISO 8879, "Information processing - text and office systems - Standard Generalized Markup Language (SGML)," Geneva, 1986.

[101] O. Lassila and R.R. Swick (editors), "Resource Description Framework (RDF) - Model and Syntax Specification," W3C Recommendation, February 1999.

[102] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130-137, July 1980.

[103] A. Griffiths, H.C. Luckhurst and P. Willet, "Using interdocument similarity information in document retrieval systems," *Journal of the American Society for Information Science*, vol. 37, no.1, pp. 3-11, January 1986.

[104] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[105] K. Hirota and W. Pedrycz, "Fuzzy computing for data mining," *Proceedings of the* IEEE, vol. 87, no. 9, pp. 1575-1600, September 1999.

[106] S. Russell and W. Lodwick, "Fuzzy clustering in data mining for telco database marketing campaigns," In: *Proceedings of the* 18th NAFIPS *International Conference*, pp. 720-726, New York, USA, June 1999.

[107] J. Wu, H. Yan and A.N. Chalmers, "Color image segmentation using fuzzy clustering and supervised learning," *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 397-403, October 1994.

[108] Y.A. Tolias and S.M. Panas, "Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions," IEEE *Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans)*, vol. 28, no. 3, pp. 359-369, May 1998.

[109] F. Höppner, F. Klawonn, R. Kruse and T. Runkler. Fuzzy Cluster Analysis: methods for classification, data analysis and image processing. John Wiley & Sons, Chichester, England, 1999.

[110] J.C. Bezdek, J. Keller, R. Krisnapuram and N. R. Pal. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers, 1999.

[111] S. Miyamoto, "An overview and new methods in fuzzy clustering," In: *Proceedings of the* 2nd *International Conference on Knowledge-Based Intelligent Electronic Systems*, KES'98, vol. 1, pp. 33-40, Adelaide, Australia, April 1998.

[112] D. Dumitrescu, B. Lazzerini and L.C. Jain. Fuzzy Sets and their Application to Clustering and Training. CRC Press, 2000.

[113] A.B. Geva, "Hierarchical unsupervised fuzzy clustering," IEEE *Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 723-733, December 1999.

[114] A. El-Hamdouchi and P. Willett, "Hierarchic document clustering using Ward's method," In: *Proceedings of the* 9th *International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'86, pp. 149-156, Pisa, Italy, 1986.

[115] D. Merkl, "Exploration of document collections with self-organizing maps: a novel approach to similarity representation," In: *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD'97, pp. 101-111, Trondheim, Norway, June 1997.

[116] R. Chau and C. Yeh "Multilingual text categorisation for global knowledge discovery using fuzzy techniques," In: *Proceedings of the* 2002 IEEE *International Conference on*

Artificial Intelligence Systems, ICAIS'02, pp. 82-86, Divnomorskoe, Russia, September 2002.

[117] M.E.S. Mendes and L. Sacks, "Dynamic knowledge representation for e-Learning applications," In: *Proceedings of the* 2001 BISC *International Workshop on Fuzzy Logic and the Internet*, FLINT 2001, Memorandum No. UCB/ERL M01/28, pp. 176-181, U. C. Berkeley, USA, August 2001.

[118] F. Klawonn and A. Keller, "Fuzzy clustering based on modified distance measures," In: *Proceedings of the Third International Symposium on Intelligent Data Analysis*, IDA'99, LNCS 1642, pp. 291-301, August 1999.

[119] R. Krishnapuram and J.M. Keller, "A possibilistic approach to clustering," IEEE *Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, May 1993.

[120] R. Krishnapuram and J.M. Keller, "The possibilistic c-means algorithm: insights and recommendations," IEEE *Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 384-393, August 1996.

[121] R.N. Davé and R. Krishnapuram, "Robust clustering methods: a unified view," IEEE *Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 270-293, May 1997.

[122] M. Barni, V. Cappellini and A. Mecocci, "Comments on: a possibilistic approach to clustering," IEEE *Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 393-396, August 1996.

[123] I. Gath and A. Geva, "Unsupervised optimal fuzzy clustering," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 773-781, July 1989.

[124] G. Beni and X. Liu, "A least biased fuzzy clustering method," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 954-960, September 1994.

[125] X.L. Xie and G.A. Beni, "A validity measure for fuzzy clustering," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841-847, August 1991.

[126] Y. Fukuyama and M. Sugeno, "A new method of choosing the number of clusters for the fuzzy c-means method," In: *Proceedings of the* 5th *Fuzzy Systems Symposium*, pp. 247-250, 1989.

[127] M.R. Rezaee, B.B.F. Lelieveldt and J.H.C. Reiber, "A new cluster validity index for the fuzzy c-mean," Pattern Recognition Letters, vol. 19, no. 3-4, pp. 237-246, March 1998.

[128] N.R. Pal and J.C. Bezdek, "On cluster validity for the fuzzy c-means model," IEEE *Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370-379, August 1995.

[129] J.A. Swets, "Measuring the Accuracy of Diagnostic Systems," Science, vol. 240, no. 4858, pp. 1285-1293, June 1988.

[130] Information Society Technologies (IST) program of the European Commission: http://www.cordis.lu/ist/

[131] Dublin Core Metadata Initiative (DCMI), "Dublin Core Metadata Element Set, Version 1.1: Reference Description," DCMI Recommendation, July 2003. Available at: http://dublincore.org/documents/dces/

[132] National Institute for Standards and Technology (NIST): http://www.nist.gov/

[133] M.R. Quillian, "Word concepts: a theory and simulation of some basic capabilities," *Behavioural Science*, vol. 12, pp. 410-430, 1967.

[134] M. Minsky, "A Framework for Representing Knowledge," In P. H. Winston (Editor). The Psychology of Computer Vision, pp. 211-277, McGraw-Hill, New York, 1975.

[135] T. R. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.

[136] International Organization for Standardization (ISO), ISO/IEC 13250:2000, "Information technology - SGML Applications - Topic Maps," Geneva, 2000.

[137] M.S. Lacher and S. Decker, "RDF, topic maps, and the semantic web," *Markup Languages: Theory and Practice*, vol. 3, no. 3, pp.313-331, Summer 2002.

[138] G. D. Moore, "RDF and topic maps: an exercise in convergence," *Interchange*, vol. 7, no. 2, pp. 11-20, June 2001.

[139] S. Pepper and G. Moore, "XML topic maps (XTM) 1.0", TopicMaps.Org Specification, August 2001.

[140] W. Pedrycz and F. Gomide. An introduction to fuzzy sets: analysis and design. MIT Press, Cambridge, MA, 1998.

[141] B. Le Grand and M. Soto, "Visualisation of the Semantic Web: topic maps visualisation," In: *Proceedings of the Sixth International Conference on Information Visualisation*, IV'02, pp. 344-349, London, UK, July 2002.

[142] B. Bos, H.W. Lie, C. Lilley and I. Jacobs (editors), "Cascading Style Sheets - level 2 (CCS2)", W3C Recommendation, May 1998.

[143] S. Adler, A. Berglund, J. Caruso, S. Deach, T. Graham, P. Grosso, E. Gutentag, A. Milowski, S. Parnell, J. Richman and S. Zilles (editors), "The eXtensible Stylesheet Language (XSL)", W3C Recommendation, October 2001.

[144] J. Clark (editor), "XSL Transformations (XSLT)", W3C Recommendation, November 1999.

[145] J.P. Bao, J.Y. Shen, X.D. Liu and H.Y. Liu, "Quick asymmetric text similarity measures", In: *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, vol.1, pp. 374-379, Xi'an, China, November 2003.

[146] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, no. 7, pp. 1109-1120, July 1997.

[147] S. Sen and R.N. Davé, "Agglomerative model for fuzzy relational clustering," In: *Proceedings of the* 19th *International Conference of the North American Fuzzy Information Processing Society*, NAFIPS 2000, pp. 267-271, Atlanta, USA, July 2000.

[148] A. Devillez, P. Billaudel and G.V. Lecolier, "A fuzzy hybrid hierarchical clustering method with a new criterion able to find the optimal partition," *Fuzzy Sets and Systems*, vol. 128, pp. 323-338, June 2002.

[149] J.F. Kolen and T. Hutcheson, "Reducing the time complexity of the fuzzy c-means algorithm," IEEE *Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 263-267, April 2002.

[150] R. Brandaw, K. Mitze and L.F. Rau, "Automatic condensation of electronic publications by sentence selection," *Information Processing & Management*, vol. 31, no. 5, pp. 675-685, September 1995.

[151] D.P. Bertsekas. Nonlinear Programming. Athena Scientific, Belmont, Massachusetts 1995.