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

Alberto Pérez García-Plaza, Víctor Fresno, and Raquel Martínez

NLP & IR Group, UNED, Madrid, Spain
{alpgarcia,vfresno,raquel}@lsi.uned.es

Abstract. Document representation is an essential step in web page clustering. Web pages are usually written in HTML, offering useful information to select the most important features to represent them. In this paper we investigate the use of nonlinear combinations of criteria by means of a fuzzy system to find those important features. We start our research from a term weighting function called Fuzzy Combination of Criteria (*fcc*) that relies on term frequency, document title, emphasis and term positions in the text. Next, we analyze its drawbacks and explore the possibility of adding contextual information extracted from inlinks anchor texts, proposing an alternative way of combining criteria based on our experimental results. Finally, we apply a statistical test of significance to compare the original representation with our proposal.

Keywords: web page, representation, fuzzy logic, clustering.

1 Motivation

Document representation is an essential step in web page clustering. The most common approach consists in trying to capture the importance of the words in the document by means of term weighting functions. Most of these functions work following the Vector Space Model (VSM) [11] and among them, tf-idf is one of the most widely used. This function works with plain text, but does not exploit other additional information that some kind of documents contain.

In order to determine the words that better represent document contents, one of the initial hypothesis of present work is that a good representation should be based on how humans read documents. We usually search for visual clues used by authors to capture our attention as readers.

The HTML tags provide additional information about those visual clues that can be employed to evaluate the importance of document terms in addition to term frequency. Regarding the way of combining different criteria within the VSM, probably the most straightforward way is a linear combination of heuristic criteria like the Analytical Combination of Criteria (*acc*) [4]. These criteria

* The authors would like to thank the financial support for this research to the Spanish research projects MA2VICMR (S2009/TIC-1542) and Holopedia: the automatic encyclopedia of people and organizations (TIN2010-21128-C02).

A. Gelbukh (Ed.): CICLing 2012, Part II, LNCS 7182, pp. 157–168, 2012.

are extracted from both text reading and writing processes, allowing to set different weights for each criterion. Its main drawback comes from the problem of nonlinearity in the combination of criteria, in other words, the fact that the contribution of one criterion can depend on the rest of the criteria: when a term is important in a single criterion, e.g. in title, the corresponding component will have a value which will always be added to the importance of the term in the document, regardless of the importance of the rest of the components.

To solve this issue we need a system that allows to define related conditions to establish term importance, e.g., a term should appear in the title and emphasized in the document to be considered important, in order to avoid rhetoric titles where words are not representative for the document topic. Because of this, we are interested in nonlinear combinations of criteria. In this context, [3] and [10] presented a document representation based on a fuzzy combination of heuristic criteria (*fcc*). The framework they presented is used here as a starting point to explore the possibilities of these systems to help apply expert knowledge and combine criteria in a nonlinear fashion. As a result, we present a representation resulting of our findings, showing significant improvements over *fcc*.

The remainder of this paper is organized as follows. In Section 2 we summarize related works. Section 3 presents *fcc* web page representation . Experiments to study how to improve *fcc* and to add contextual information to our representation are performed in Section 4. Finally, empirical evaluation is performed in Section 5, concluding the paper in Section 6.

2 Background

Most of document representation approaches are based on the VSM, where each document is represented as a vector, and each vector component corresponds to a value which tries to express the importance of the term in the document. These components are also called features, and their value is called feature or term weight. One of the most widely used functions to calculate term weight is tf-idf, that combines term frequency in a document with document frequency of the same term. On one hand, this representation does not take into account the additional information one can find in web pages, just plain text, and, on the other hand, it is worth to notice the fact that, in IR, field where tf-idf was defined, the goal is to find differences instead of similarities.

Some researchers have presented new representations based on variations of tf-idf. In [5] the authors propose to employ keyphrases instead of words, introducing some changes like rewarding instead of penalizing keyphrases that appears in many documents and having into account whether or not they appear in titles or headers by means of a linear combination, but they neither specify the exact weights for each component nor the way of calculating it. In [8] the authors consider that document title, textual content and anchor texts have different importance levels and decide to represent each one with a separate tf-idf feature vector. This requires a particular clustering algorithm so it was not compared to other representations, but with other algorithms. This model does not allow to include new criteria to the representation without changing the whole system

(input format and algorithm). Their results show only average precision, but including recall could lead to different conclusions.

With the same objective, [3] presents a self-content representation for web pages. It is called Fuzzy Combination of Criteria (*fcc*) and has been successfully applied in clustering and classification, where it has been compared with different state of the art alternatives, like *acc* or tf-idf, obtaining good results. The main difference with the above mentioned work is that *fcc* keeps the VSM as it is. To do this, the proposed weighting function uses a fuzzy system to heuristically combine criteria. Concretely, four criteria are used: term frequency, term frequency in title, term frequency in emphasis and term positions in the document. Besides, the fuzzy logic engine provides the possibility of adding new criteria and modify the rules easily, which allows to study the contribution of each criterion. For these reasons, in addition to the discussion about linear and nonlinear combinations detailed in Section 1, we have chosen the framework offered by *fcc* to develop our work. Among the fields explored by previous works on *fcc*, the most promising results were achieved in clustering tasks, reason why in this work we will focus our research in this field, using *fcc* as a baseline.

Another alternative to enrich web page representation is adding some kind of links information. In [15] a study about how to combine textual content and link analysis is performed. They use inlinks and outlinks in order to improve clustering applied to search results. Their empirical results suggest that the combination of both, textual content and links can improve web page clustering. About using anchor texts, [9] gives some interesting ideas. They state that anchor texts contribute meaningful information for IR tasks, but this information is not as good to capture the aboutness of web documents. They agree with [2] that anchor text terms are similar to terms used in search queries. Besides, these terms are not often in web page contents, concretely in [9] they found only 51% of the cases, while in [2] they found 66.4% of terms appearing on both. Anchor texts are a lightweight and efficient alternative compared to other more complex methods of anchor context extraction.

In present work we study web page representation by means of fuzzy combinations of heuristic criteria, analyzing the contribution of each criterion to improve clustering results. We also explore not self-content information like anchor texts to extend the combination.

3 Fuzzy Combination of Criteria

For a human reader, title and emphasized words in a text document have a bigger role than the rest of the document in understanding its main topic. Moreover, the beginning and the end of the body text usually contain overviews, summaries or conclusions with essential vocabulary. The goal of *fcc* [10] is to define the importance level of each word in a document by using a set of heuristic criteria: word frequency counts in titles, emphasized text segments, in the beginning and the end of the document, and in the whole document. As titles and other special texts are encoded with HTML tags, a subset of those tags are used in *fcc* in order to collect "the most important" words in a document.

The fuzzy system is built over the concept of linguistic variable. Each variable describes the membership degree of a word to a particular class. The variables are defined by human experts. The fuzzy system knowledge base is defined by a set of IF-THEN rules that combine the variables. The aim of the rules is to combine one or more input fuzzy sets (antecedents) and to associate them with an output fuzzy set (consequent). Once the consequents of each rule have been calculated, and after an aggregation stage, the final set is obtained.

The *fcc* IF-THEN rules are based on the following ideas: (1) If a word appeared in the title or the word was emphasized, that word should also appear in one of the other criteria in order to be considered important. (2) Words appearing in the beginning or at the end of a document may be more important than the other words, because documents usually contain overviews and summaries in order to attract the interest of the reader. (3) If a word is not emphasized, it is possible that there are no emphasized words in the document at all. (4) If a word does not appear in the title, it is possible that the document does not have a title at all, or the title does not contain important words. (5) If the previous criteria were not able to choose the most important words, the frequency counts may help to find them. The knowledge base for *fcc* is shown in Table 1. Each row has the values of different criteria and the resulting output, called 'Importance'. The inference engine that evaluates the fired rules is based on the *center of mass* (COM) algorithm that weights the output of every fired rule, taking into account the truth degree of its antecedent. The output is a linguistic label (e.g., 'Low', 'Medium', 'Very High') with an associated number related to the importance of a word in the document, and it is calculated by scaling the membership functions by product and combining them by summation. These kind of systems are called

Table 1. Rule base for *fcc*. Inputs are related to normalized term frequencies.

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

additive [7] and their main advantage is the efficiency of the computation. A more detailed explanation of the fuzzy system can be found in [3,10].

4 Proposing a New Combination

In this section we will use the framework offered by *fcc* to further investigate about how information extracted from HTML documents can improve document clustering. Each subsection is based on the previous ones in order to build our representation proposal step by step. Some experimental settings will be the same for all the experiments, so they are outlined hereafter.

Experimental Settings. In preprocessing, a stop word list was used to remove common words. The punctuation was also removed. Suffixes were removed using a standard implementation of the Porter's algorithm for English. Regarding the clustering process, we chose Cluto rbr (k-way repeated bisections globally optimized) as a state of the art algorithm [6]. It is a widely used algorithm with good results in the literature [1,3,13]. Algorithm parameters were set by default.

After weighting terms, we reduced document vectors to 100, 500, 1000, 2000, and 5000 dimensions using two methods: Most Frequent Terms until n level (*mft*) and Latent Semantic Indexing (*lsi*). The *mft* method works as follows: first ranks the terms in each document based on the term weighting function values. Then, terms on the first position in the document rankings are put in order according to how many times they have appeared in the rankings. If two or more terms appear the same number of times in different rankings, we put them in order based on the maximum weight found for each of them. Next we take the terms appearing in the second position in the rankings, and so forth. The process stops when the desired number of terms is reached. Notice that by following this algorithm the resulting list may be larger than the required size, because there are as many rankings as documents in the dataset. Nevertheless, as we put the list in order, we can get the exact number of terms just taking the first n terms. Regarding *lsi*, as suggested by [14], it was applied after a previous reducing step to alleviate its computational complexity. We reduced vector dimension using *mft* from the original size to 5000 features before applying *lsi*.

To evaluate the clustering quality for clustering algorithms, typically the F-Measure (equation 1) is used [12], which is equal to the harmonic mean of recall and precision. The overall F-measure is the weighted average of the F-measure for each category:

$$F(i,j) = \frac{2 \cdot Recall(i,j) \cdot Precision(i,j)}{Recall(i,j) + Precision(i,j)} \quad ; \quad F = \sum_i \frac{n_i}{n} \cdot \max_j \{F(i,j)\} \quad (1)$$

where i is the category, j the cluster and n the number of documents. The F-measure values are in the interval (0,1) and larger values correspond to higher clustering quality.

Datasets. In previous work [3] two different datasets were used: Banksearch and Webkb. Because of this, we decided to use these datasets to obtain comparable

results. Banksearch contains 11,000 documents divided in 11 categories of equal size, divided in two hierarchy levels: 10 main categories at the same level and another one parent of two of them. Our experiments are not oriented to hierarchical clustering, so we use the 10 main categories, corresponding to 10,000 documents. In Webkb we removed 'others' category because introduced noise, resulting in 6 categories for a total of 4,518 documents. Webkb categories are unbalanced with respect to the number of documents in each category (3% of documents in the smallest category, 35% of documents in the biggest one).

4.1 How Does Dimension Reduction Affect Weighting Function?

In order to explore the effect of dimension reduction techniques over the term weighting function we decided to use tf-idf, because it is a standard in clustering, and *fcc*, which will be our baseline.

Table 2. F-measure results for dimension reduction experiments

Rep.\Dim.	100	500	1000	2000	5000	Avg.	S.D.
Banksearch							
tf-idf mft	0,703	0,737	0,768	**0,772**	0,758	0,748	0,028
tf-idf lsi	0,750	0,755	0,756	0,757	0,763	0,756	**0,005**
fcc mft	0,723	0,757	0,768	0,765	**0,768**	0,756	0,019
fcc lsi	**0,775**	**0,763**	**0,785**	0,763	0,758	**0,769**	0,011
Webkb							
tf-idf mft	0,385	0,438	0,466	0,498	**0,513**	0,460	0,051
tf-idf lsi	**0,516**	**0,507**	**0,505**	**0,506**	0,501	**0,507**	**0,006**
fcc mft	0,453	0,472	0,475	0,468	0,475	0,469	0,009
fcc lsi	0,449	0,460	0,473	0,474	0,475	0,466	0,011

Table 2 shows the F-measure results for all the combinations of weighting functions and dimension reduction techniques for both datasets. Each table row contains F-measure values corresponding to the clustering solution obtained by using the representation specified in the first column with the number of features per vector detailed on top of the remaining columns, being Avg. and S.D. the average and the standard deviation for that row.

Between *lsi* and *mft*, results are different depending on the term weighting function. For tf-idf, *lsi* always improves *mft* in Banksearch when vector size is small (100 and 500 features). However, with 1,000 and 2,000 features *mft* obtained higher results, and with 5,000 the difference is $< 1\%$. In Webkb occurs something similar, but the difference also appears with 1,000 features. As *mtf* strongly depends on the term weighting function to select the most important terms, an improvement of *lsi* over *mft* implies that the weighting function is not working as well as it could. Looking at *fcc*, *lsi* does not improve its results in Webkb, except in one case, but the difference is $< 1\%$. In Banksearch *lsi* improves *mft* when reducing to 1,000 or less features, and only in 2 cases the difference between them is $> 1\%$. Comparing both functions, while *fcc* outperforms tf-idf in Banksearch, in Webkb the best results correspond to tf-idf helped by *lsi*.

Our hypothesis is that the improvement obtained by *lsi* over *mft* is a consequence of the term weighting function, because *lsi* is a feature transformation technique that could allow to discover relations among features, removing those

less representative. Therefore, if we are able to choose the most representative features of each document, *mft* should work, at least, similar to *lsi*.

4.2 Analysis of the Combination of Criteria

Section 4.1 left two open issues: to improve the bad performance of *fcc* in Webkb dataset, and to validate our hypothesis. Both of them are clearly related because if we improve the weighting function, our hypothesis says that the new results should be more similar to those obtained using *lsi*. In this section we perform a comprehensive study about how to improve *fcc* for document clustering.

Study of Individual Criteria. The first step is to analyze the contribution of each criteria in order to find any clue about why the combination does not perform in Webkb as well as in Banksearch. To do this, we repeat the clustering process modifying the combination of criteria proposed by *fcc*. We did four variations of this function, one per each criterion, in such a way that the output of the system will correspond only to one criterion at a time. We used *mft* reduction because it does not transform features, allowing us to study the effectiveness of each alternative to give more importance to the most representative terms.

Table 3. F-measure results for criteria analysis experiments

Rep.\Dim.	100	500	1000	2000	5000
Banksearch					
fcc mft	**0,723**	**0,757**	**0,768**	**0,765**	**0,768**
title	0,626	0,646	0,632	0,634	0,639
emphasis	0,586	0,671	0,674	0,685	0,693
frequency	0,689	0,715	0,720	0,724	0,731
position	0,310	0,525	0,538	0,599	0,608
Webkb					
fcc mft	**0,453**	**0,472**	**0,475**	0,468	0,475
title	0,432	0,433	0,404	**0,488**	0,479
emphasis	0,415	0,431	0,433	0,465	**0,489**
frequency	0,441	0,460	0,460	0,468	0,446
position	0,301	0,283	0,317	0,281	0,286

Table 3 shows the results of each individual criterion compared to *fcc*. Focusing on Banksearch results, values corresponding to *fcc* are always higher than individual ones. This means that the combination contributes to improve the results over individual criteria in all cases. Besides, frequency obtains the best values, while position obtains the worst ones. Webkb results are quite different. On one hand, frequency is not always the best among individual criteria and, on the other hand, *fcc* does not always outperform individual criteria, concretely title, emphasis or frequency have higher F-measure values in some cases when reducing vector dimension to 2000 and 5000 features. It seems that frequency strongly affects results, and going further, when title and emphasis could lead to a better clustering, their combination with frequency makes results worse. Therefore, while frequency gets higher results than the other criteria the combination works fine, but when titles or emphasis outperforms frequency, the combination does not work as good as it could. Thus, frequency is very important for a good grouping, as well as title and emphasis, and all of them should be very important

in the combination. However, position is the criterion with the worst results in all cases, so we have to take care using it to establish the importance of a term.

Improving the Fuzzy Combination of Criteria. In *fcc* rules (Table 1), when frequency is 'low' output can be 'very high' (the maximum) depending on the position, if title and emphasis are high. As we saw before, frequency contributes to a good clustering much more than position, so the output should reflect that fact, but in this case frequency is totally ignored. This occurs again when title is 'low' and frequency 'medium'. Both criteria are important for a good grouping, but the output is 'very high' based in term position, the same as the previous case. In these cases we are clearly underestimating the discrimination power of frequency and title. The same happens when frequency is 'medium', being title and emphasis 'low': position decides again that importance can be the minimum or not, but frequency should count more than position, as we saw before. Summarizing, *fcc* overestimates the contribution of position, underestimating at the same time the discriminative power of title, emphasis and frequency.

On the other hand, the high number of rules in *fcc* makes the possible combinations more difficult to understand. As the fuzzy system is able to combine the conclusions of the rules, another possibility for the knowledge base is the use of a set of single-input rules for each criterion and let the system calculate the output (*addfcc*, Table 4). This approach represents the knowledge in a simple way, reducing the number of cases that is needed to specify.

Table 4. Rule base for *addfcc*. Inputs are related to normalized term frequencies

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

Nevertheless, if we are looking for very specific definitions for each criterion, we may miss part of the knowledge expressed in the *fcc* system, especially when dealing with dependencies among criteria and not all of them contribute equally to the combination, as occurs in our case. In order to avoid this problem, an intermediate approach is proposed. We refer to it as Extended Fuzzy Combination of Criteria (*efcc*, Table 5). The main idea is to have two sets of rules: one for frequencies and another for the rest of the criteria, in such a way that we have always at least one rule of each set fired by the system, which will combine the outputs. Thus, we simplify the problem of underestimating frequency, because both subsets are always evaluated and combined. We have also reduced the discriminative power of position criterion, that is considered the least important.

For tf-idf and *fcc* we only show the best results for each dataset from Section 4.1 in order to simplify the comparison. Results on Table 6 show how *efcc* clearly improves clustering results in Webkb, while in Banksearch *addfcc* outperforms

Table 5. Rule base for *efcc*. Inputs are related to normalized term frequencies.

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

Table 6. F-measure results for *addfcc* and *efcc* experiments

Rep.\Dim.	100	500	1000	2000	5000	Avg.	S.D.
Banksearch							
tf-idf lsi	0,750	0,755	0,756	0,757	0,763	0,756	**0,005**
fcc lsi	0,775	0,763	**0,785**	0,763	0,758	0,769	0,011
efcc mft	0,768	0,778	0,758	0,740	0,759	0,760	0,014
efcc lsi	**0,780**	0,756	0,744	0,755	0,757	0,758	0,013
addfcc mft	0,775	**0,788**	0,777	**0,784**	**0,779**	**0,781**	**0,005**
Webkb							
tf-idf lsi	**0,516**	0,507	0,505	0,506	**0,501**	0,507	0,006
fcc mft	0,453	0,472	0,475	0,468	0,475	0,469	0,009
efcc mft	**0,516**	**0,546**	**0,545**	**0,566**	0,484	**0,532**	0,032
efcc lsi	0,483	0,483	0,483	0,483	0,484	0,483	**0,000**
addfcc mft	0,459	0,493	0,494	0,491	0,471	0,482	0,016

the rest. Thus, *efcc* solves the first open issue stated at the beginning of Section 4.2: improving the bad performance of *fcc* in Webkb, with good results in Banksearch too. Besides, *addfcc* leads to worse results than *efcc* in Webkb , but obtains the best results in Banksearch in almost all cases.

These experiments for *efcc* also corroborates our hypothesis about the improvement obtained by *lsi* over *mft*, stated in Section 4.1: we have improved our weighting function and, as a result, *mft* has achieved clustering results as good as, or even better than *lsi*, with a much lower computational cost.

4.3 Anchor Texts

For this experiment we needed to employ a recently crawled collection, in such a way that it was easy to find other web pages with hyperlinks to the collection documents. We decided to use the dataset Social ODP 2k9 (SODP) [16] consisting of 12,616 documents retrieved from social bookmarking sites and classified by extracting the category for each URL from the first classification level of Open Directory Project. Thus, the entire collection is divided in 17 unbalanced categories, having from 39 to 3,289 documents each. In addition to the documents themselves, we collected the anchor texts corresponding to a maximum of 300 unique inlinks per each document in the collection (2,704 web pages have less than 50 inlinks, 4,717 have less than 100, so the rest, approximately 60%, have more than 100 inlinks).

We decided to add anchor texts to *efcc* in two different ways: (a) in addition to each document textual content, and (b) in addition to each document title,

i.e., giving them the same importance than title terms. Besides, we did three experiments for each case: (1) just adding anchor texts, (2) adding anchor texts and removing text corresponding to outlinks, and (3) removing a set of stop words based on a study over collection anchor text terms, containing words like click, link, homepage, etc. As we introduce here a new collection, we decided to add *fcc* as baseline to validate our results. We also include *addfcc* in these experiments due to its good performance in Banksearch (Section 4.2).

Table 7. F-measure results for anchor text experiments

Rep.\Dim.	100	500	1000	2000	5000	Avg.	S.D.
SODP							
fcc mft	0,195	0,237	0,254	0,256	0,266	0,242	0,028
addfcc mft	0,208	0,267	0,276	0,279	0,282	0,262	0,031
efcc mft	0,233	0,273	**0,287**	0,283	**0,296**	0,275	0,025
efcc a-1 mft	0,225	0,262	0,279	0,286	0,290	0,268	0,027
efcc a-2 mft	0,245	0,246	0,285	0,289	0,269	0,267	0,024
efcc a-3 mft	0,248	0,260	0,285	**0,294**	0,293	0,276	0,022
efcc b-1 mft	**0,254**	**0,287**	0,275	0,282	0,285	**0,277**	0,015
efcc b-2 mft	**0,254**	0,249	0,276	0,279	0,291	0,270	0,016
efcc b-3 mft	0,249	0,261	0,263	0,278	0,285	0,267	**0,012**

Table 7 shows that *efcc* based approaches outperforms *fcc* and *addfcc* in all cases. This corroborates our findings of Section 4.2 about the drawbacks of *fcc* and confirms our believe about the need of a system where not all criteria contribute the same to the combination, in contrast to *addfcc*. Regarding the contribution of anchor texts, there is no clear alternative to improve *efcc* Anchor texts help improve clustering results with small vector sizes, particularly when anchor texts terms are considered as page titles. However, when we increase vector size, they seem to introduce noise, because clustering results get worse. About using anchor texts as titles, the best option is just adding anchor texts as title terms (named b-1). Although it is interesting to have found an improvement for smaller vector sizes, this improvement is always about 1%, and clearly does not compensate for all the process needed to obtain anchor texts.

These results might be due to poor link density or bad anchor text quality, or just to the nature of clustering problems, where the aim is to capture the aboutness of documents and not just concrete keywords. This conclusions coincide with other works like [2,9] (see Section 2), where authors conclude that anchor text terms are similar to terms used in search queries and these terms are not often in web page contents. Because of this, we believe that anchor texts are more suitable for IR tasks than for clustering problems.

5 Empirical Evaluation

In this section we perform a robust evaluation of *efcc* to be sure about whether or not exists a real improvement over *fcc*. As we are using a deterministic algorithm, we want to avoid the possible bias introduced by feeding the algorithm with a single set of vectors for each dataset. The solution presented here consists in dividing each dataset in 100 different sub-datasets 50% smaller than the original,

where the categories are in proportion to the original ones. We performed 100 experiments per each vector size and each sub-dataset, resulting a total of 3,000 different clustering experiments. Due to computational reasons, we chose *mft* reduction for all the experiments. This decision was also made to compare both term weighting functions in the exactly same conditions. Finally, we calculated the statistical significance between F-measure results of both representations. To this end, we employed a paired two-tailed t-test over the results obtained by both representations for each concrete vector size in the 100 sub-datasets.

Table 8. F-measure results for t-test experiments

Rep.\Dim.	**100**	**500**	**1000**	**2000**	**5000**
Banksearch					
efcc mft	**0,764**	**0,774**	**0,770**	0,760	0,753
fcc mft	0,718	0,760	0,765	**0,768**	**0,768**
Difference	0,047	0,014	0,006	-0,008	-0,015
p-value	0,000	0,000	0,002	0,000	0,000
Webkb					
efcc mft	**0,487**	**0,514**	**0,528**	**0,534**	0,483
fcc mft	0,446	0,462	0,470	0,485	**0,490**
Difference	0,041	0,051	0,059	0,049	-0,007
p-value	0,000	0,000	0,000	0,000	0,016
SODP					
efcc mft	**0,230**	**0,271**	**0,279**	**0,282**	**0,289**
fcc mft	0,200	0,233	0,246	0,251	0,266
Difference	0,030	0,037	0,033	0,031	0,023
p-value	0,000	0,000	0,000	0,000	0,000

In Table 8, for each vector size and representation we show the average F-measure values corresponding to the 100 clustering experiments (one per each sub-dataset), the difference between the corresponding averages, and the p-value resulting of applying the statistical t-test between both representations. Attending to p-values, in all cases except one, we can say that values are from different populations with likelihood $> 99\%$. Besides, looking at the averages, in most of the cases *efcc* outperforms *fcc*. Regarding differences between representations, just in three cases *fcc* performs better than *efcc*, being the difference lower than 1% in two cases and lower than 2% in the other. In the rest of the experiments *efcc* gets an improvement over *fcc*, higher than 3% in SODP, and greater than 4% in Webkb and even with the smallest vector size in Banksearch.

6 Conclusions

Our experiments showed that *efcc* worked better than *fcc* by means of a better combination of criteria, where term frequency is considered as discriminant as title and emphasis, and position is taken into account as the least important criterion. This approach makes also possible to reduce the number of rules needed to specify the knowledge base taking advantage of the additive properties of the fuzzy system, and thus makes the system easier to understand. Moreover, we have shown that with a good weighting function we can use lightweight dimension reduction techniques, as the proposed *mft*, instead of using *lsi*, which implies an important reduction in computational cost. In order to continue exploring new criteria for the combination, we have evaluated the use of anchor

texts to enrich document representation. Although results were not bad, the cost of preprocessing anchor texts and their dependence on link density limit the applicability of this alternative. For this reasons we believe that it could be an interesting option when a collection fulfills these requirements and time complexity is not a problem, but in most of the cases this will not happen and we will have to carry out document representation only with document contents. Finally we performed statistical significance tests to ensure that the application of our findings has a real effect compared to previous work.

Future work could be oriented to find a way of automatically adjusting the representation to specific datasets and analyzing whether or not improves clustering results. Moreover, present work could be applied to different fields, not only the representations by themselves, but the underlying ideas.

References

1. Dredze, M., Jansen, A., Coppersmith, G., Church, K.: Nlp on spoken documents without asr. In: EMNLP, pp. 460–470 (2010)
2. Eiron, N., McCurley, K.S.: Analysis of anchor text for web search. In: Proceedings of the 26th SIGIR, pp. 459–460 (2003)
3. Fresno, V.: Representacion autocontenida de documentos HTML: una propuesta basada en combinaciones heuristicas de criterios. PhD thesis (2006)
4. Fresno, V., Ribeiro, A.: An analytical approach to concept extraction in html environments. J. Intell. Inf. Syst. 22(3), 215–235 (2004)
5. Hammouda, K., Kamel, M.: Distributed collaborative web document clustering using cluster keyphrase summaries. Information Fusion 9(4), 465–480 (2008)
6. Karypis, G.: CLUTO - a clustering toolkit. Technical Report #02-017 (November 2003)
7. Kosko, B.: Global stability of generalized additive fuzzy systems. IEEE Transactions on Systems, Man, and Cybernetics - C 28, 441–452 (1998)
8. Liu, Y., Liu, Z.: An improved hierarchical k-means algorithm for web document clustering. In: ICCSIT, September 2-29, pp. 606–610 (2008)
9. Noll, M.G., Meinel, C.: The metadata triumvirate: Social annotations, anchor texts and search queries. In: Proceedings of the WI-IAT, vol. 1, pp. 640–647 (2008)
10. Ribeiro, A., Fresno, V., Garcia-Alegre, M.C., Guinea, D.: A fuzzy system for the web page representation (2003)
11. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM (1975)
12. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining (2000)
13. Tan, Q., Mitra, P.: Clustering-based incremental web crawling. ACM Trans. Inf. Syst. 28, 17:1–17:27 (2010)
14. Tang, B., Shepherd, M., Milios, E., Heywood, M.I.: Comparing and combining dimension reduction techniques for efficient text clustering. In: Proceedings of the Workshop on Feature Selection for Data Mining, SDM (2005)
15. Wang, Y., Kitsuregawa, M.: Evaluating contents-link coupled web page clustering for web search results. In: CIKM, pp. 499–506 (2002)
16. Zubiaga, A., Martínez, R., Fresno, V.: Getting the most out of social annotations for web page classification. In: ACM DocEng, pp. 74–83 (2009)